DILOG lecture notes

1 A non-symbolic introduction to logic

Before venturing into the subject of logic as a world of symbols, formal languages, and rules, let us look at it from a relatively high level. Besides making the fundamental concepts less abstract, it will hopefully indicate their usefulness as tools for reasoning, even outside logic proper.

In general, logic concerns the consistency of sets of beliefs and the validity of arguments. A set of beliefs is consistent if there is at least one possible situation in which it is reasonable to hold them all together. If there is no such situation, then the set of beliefs is said to be inconsistent. An argument is valid if the premises guarantee the conclusion; otherwise, it is invalid.

Whether beliefs are consistent can be difficult, if not impossible, to determine unless we express them clearly. To only think about and imagine things is not particularly helpful when trying to convince or ask others about the consistency of one's beliefs. In this course, we assume they have to be written down. Thereby we deal with beliefs only as far as they can be expressed in writing (or verbally). The same goes for premises and conclusions when dealing with arguments.

At least in this context, we express beliefs, premises, and conclusions as declarative sentences. A declarative sentence expresses something that is either true or false in a given situation. Thus, it needs to reference something real or imagined in the world. We can determine a sentence as declarative if we can prepend it with "Is it true that..." and turn it into a meaningful question. For example, "Boris is a sled dog" is a declarative sentence, and it makes perfect sense to ask, "Is it true that Boris is a sled dog?" as long as there is a clear reference to some dog named Boris. The exclamation "Get over it!" is not a declarative sentence and would not survive our test, trying to use it in forming a question: "Is it true that get over it!?"

The same declarative sentence can be true in one situation yet false in another. A trivial example is: "It is raining outside." It could be true today but false tomorrow. It could be true in Seattle but false in Tuscon. Of course, one could argue that "It is raining outside" is just a generalized representation of sentences such as "It is raining outside the entrance of Notre Dame in Paris at 14:32 on April 12, 2023." Clearly, there are many such sentences, and all could be generalized to "It is raining outside" as long as the context otherwise is clear. Hence, whether a unique declarative sentence can refer to more than one situation is questionable. However, if we choose to look at sentences such as "It is raining outside" as types of sentences, then we should have no problem assigning either true or false to what seems to be one and the same sentence.

We will take it one step further and speak about *propositions*–the meanings of sentences. The declarative sentence ”It is raining outside” is syntactically the same whenever written or uttered but does not necessarily express the same proposition. Since propositions presuppose meaning, we are better positioned to directly assign truth or falsity to those rather than to declarative sentences.

Since we can express beliefs, premises, and conclusions as declarative sentences, and propositions give us the meaning of those sentences, we define consistency and validity in terms of propositions instead. By doing so, we can use these notions in tandem and make a meaningful connection between consistency and validity.

Definition 1.1. A set of propositions is *consistent* if and only if it is possible for all of them to be true at the same time.

Definition 1.2. A set of propositions is *inconsistent* if and only if it is impossible for them to be true at the same time.

Definition 1.3. The premises of an argument is a set of propositions.

Definition 1.4. The conclusion of an argument is a proposition.

Definition 1.5. An argument is *valid* if and only if it is impossible for the conclusion to be false when all the premises are true.

When attempting to determine the consistency of a set of propositions, it is about the consistency among the propositions themselves, not whether they are consistent with reality. Herein we will reason based on possible situations that can either be simple or complex. A simple situation is one that requires no further deliberation to determine the truth value of a proposition relative to that situation. For example, the proposition ”The moon orbits the earth” is true precisely in a situation where the moon orbits the earth. Hence, that situation is simple. But, the proposition ”The moon orbits the earth, and the earth orbits the sun” requires a complex situation under the assumption that the proposition is true, namely a combination of the situation where the moon orbits the earth and the situation where the earth orbits the sun. The proposition ”Kim rides a bicycle or runs” is true in a complex situation involving either of two cases, which in turn make up simple situations, namely one in which Kim rides a bike and another in which Kim runs.

We can check for the consistency of a set of propositions by first figuring out the situations that would have to hold for each of them to be true separately. Then we combine them into a more complex situation with exactly one possible situation from each set. If we can find at least one such complex situation with compatible component situations, then we have a consistent set of propositions.

Let us look at an example. Are the following three propositions consistent?

1. Kim rides a bike or runs.
2. Karen takes the tram or drives.
3. Kim and Karen go together.

The first proposition is true in a situation where [Kim rides a bike] and in a situation where [Kim runs]. The second proposition is true in a situation where [Karen takes the tram] and in a situation where [Karen drives]. The third proposition is true only in a situation where [Kim and Karen go together]. From these situations, we can form four possible combinations:

1. {[Kim rides a bike], [Karen takes the tram], [Kim and Karen go together]}
2. {[Kim rides a bike], [Karen drives], [Kim and Karen go together]}
3. {[Kim runs], [Karen takes the tram], [Kim and Karen go together]}
4. {[Kim runs], [Karen drives], [Kim and Karen go together]}

None of the above resulting situations could possibly occur since Kim's, and Karen's means of transportation make it impossible for them to go together.[1] Hence, the set of propositions is inconsistent.

If we would substitute "None of Kim and Karen goes by foot" for "Kim and Karen go together," then the set of propositions would have been consistent since none of them goes by foot as long as Kim rides a bike.

Let us now look at the validity of arguments and the use of counterexamples to determine validity. The definition of valid arguments states that there is no situation in which all the premises are true, and the conclusion is false. Another way of expressing that is that there can be no situation in which all the premises are true at the same time as it is true that the conclusion is false.

Definition 1.6. A *counterexample* is a situation in which all the premises, together with the falsity of the conclusion, are true at the same time.

Definition 1.7. The *counterexample set* of an argument is the falsity of its conclusion, together with its premises.

According to Definition 1.2, if there is no situation for a set of propositions to be true at the same time, we call that set inconsistent. Hence, if an argument is valid, its counterexample set (i.e., the premises and the falsity of the conclusion) is inconsistent, and thus there is no counterexample.

Theorem 1.1. An argument is *valid* if and only if the counterexample set is inconsistent.

To show that an argument is valid (or not), we can use Theorem 1.1 by taking the falsity of the conclusion and testing the resulting counterexample set for consistency the same way we tested the set of propositions above about Kim and Karen.

Suppose we have the following argument: "Kim either bikes or takes the bus. If Kim bikes, she saves money. Kim does not take the bus. Hence, Kim saves money." Note that "If Kim bikes, she saves money" can be expressed as "Kim saves money or does not bike." The falsity of the conclusion is "Kim does not save money," and the counterexample set is

> Kim either bikes or takes the bus. Kim saves money or does not bike. Kim does not take the bus. Kim does not save money.

Again, we have four possible combinations of situations:

[1] Perhaps there is a way for them to go together if Karen drives a van with a treadmill or stationary bike in the back or if Kim runs alongside the vehicle. The meaning and interpretation of words clearly have consequences for the consistency of sets of propositions.

1. {[Kim bikes], [Kim saves money], [Kim does not take the bus], [Kim does not save money]}

2. {[Kim bikes], [Kim does not bike], [Kim does not take the bus], [Kim does not save money]}

3. {[Kim takes the bus], [Kim saves money], [Kim does not take the bus], [Kim does not save money]}

4. {[Kim takes the bus], [Kim does not bike], [Kim does not take the bus], [Kim does not save money]}

In each of the combinations, we find contradicting premises. The counterexample set is therefore inconsistent, and we can conclude that the argument is valid.

From Theorem 1.1, it also follows that any argument with an inconsistent set of premises is valid. Suppose there is no situation in which the premises can be true at the same time. In that case, there will be no situation in which the premises, plus any other proposition, such as the falsity of the conclusion, can be true simultaneously.

Understanding the interplay between consistency and validity will make it easier to grasp the material that will follow later in the course. Try to develop and test your own examples by putting the definitions and theorems to work. In the next chapter, we will introduce the language of propositional logic and its semantics.

2 Propositional logic

Using propositional logic, we can represent and combine propositions in natural language with a small set of operators known as logical connectives and the rules accompanying these. It makes us less sensitive to meaning while carrying out certain logical operations.

We may want to represent beliefs, assumptions, premises and conclusions of arguments, facts, etc. Once we have a logical representation, we can quite straightforwardly test sets of propositions for consistency or investigate which simple or complex propositions follow from whatever set of propositions, such as a knowledge base.

There is no limit to what we can represent in propositional logic as long as it is declarative sentences. However, the logical connectives that determine how we can represent reasoning certainly limit the number of ways in which propositional logic can support reasoning. There are also limits to representing relations between propositions. For example, we cannot draw any meaningful inference about the number 4 given the following: “Any number less than five is a good number. Therefore, 4 is a good number.” We cannot represent the common reference to the natural numbers between the premise and the conclusion in propositional logic, as there are no variables that can stand for things or individuals.

2.1 Symbols and syntax

Sentences in propositional logic are referred to as formulas. We can differentiate between prime formulas and composite formulas. The former consists of a single proposition, and the latter of at least one proposition and at least one logical connective.

To represent propositional formulas, we use three different sets of symbols. One contains the logical symbols, i.e., the propositional connectives, with which we can combine formulas in certain ways; these connectives will be presented below. The second set contains symbols representing prime formulas, i.e., formulas that consist of simple propositions and do not contain any logical connectives. We will use upper case letters for these, starting with P. The third set of symbols represents composite formulas that consist of prime formulas and logical connectives. For these, we use upper case letters starting with A. The prime and composite formulas are together the set of formulas.

We can assign true or false to a prime formula quite arbitrarily. If we are using propositional logic for representing a declarative sentence such as “It is dark,” then we could assign true or false to it according to the current light conditions, or we could assume it is true (or false) and reason based on that assumption. The truth value assigned to a proposition need not have anything to do with the conditions of the real world. In fact, many times, we will assign true or false to propositions without even knowing what they may represent–the reason being that we often will reason about logical schemas rather than models of the real world.

2.1.1 Constructing formulas

Formulas in propositional logic are constructed with the help of letters representing formulas (e.g., A and B), together with any of the connectives (i.e., negation '$\neg$', conjunction '$\wedge$', disjunction '$\vee$', conditional '$\rightarrow$', and biconditional '$\leftrightarrow$'). A formula can consist of a single formula letter (e.g., 'A'), a negated formula (e.g., '$\neg A$'), the conjunction of two formulas (e.g., '$A \wedge B$'), the disjunction of two formulas (e.g., '$A \vee B$'), the conditional of two formulas (e.g., '$A \rightarrow B$'), or the biconditional of two formulas (e.g., '$A \leftrightarrow B$').

Connectives bind more or less strongly to the formulas next to them. Sometimes parentheses are needed to disambiguate between different ways of reading a formula. Negation binds the strongest, so it is never necessary to write $(\neg A) \wedge B$ to indicate that the negation is a negation of A and nothing else. It is sufficient to write $\neg A \wedge B$ in that case. There is no difference between conjunction and disjunction in terms of binding strength. Therefore, use parentheses and write either $A \vee (B \wedge C)$ or $(A \vee B) \wedge C$ depending on what you mean. The conditional and biconditional have the lowest binding strength. Consequently, $A \wedge B \rightarrow C$ is read as $(A \wedge B) \rightarrow C$, and parentheses are necessary to make it read as $A \wedge (B \rightarrow C)$.

2.2 The logical connectives and their semantics

2.2.1 Negation '$\neg$'

A formula $\neg A$ is true if and only if A is false.

2.2.2 Conjunction '$\wedge$'

A formula $A \wedge B$ is true if and only if A is true and B is true.

2.2.3 Disjunction '$\vee$'

The disjunction in propositional logic represents the inclusive OR. A formula $A \vee B$ is true if and only if at least one of A and B is true. Note the difference to exclusive OR (i.e., "either... or...") which is true if and only if exactly one of the disjuncts is true.

2.2.4 Conditional '$\rightarrow$'

A formula $A \rightarrow B$ is false if and only if A is true and B is false; otherwise, it is true. The first formula of a conditional is called the antecedent, and the second formula is the consequent. Hence, in this case, A is the antecedent, and B is the consequent.

2.2.5 Biconditional '$\leftrightarrow$'

A formula $A \leftrightarrow B$ is true if and only if A and B have the same truth value.

2.3 Interpretations

Definition 2.1. An interpretation of a formula A in propositional logic is an assignment of truth values to the prime formulas of A.

For example, if A is the formula $P \vee Q$, assigning true to P and false to Q is one (out of four possible) interpretation of A. A straightforward representation of an interpretation using the symbols $\top$ for true and $\bot$ for false yields $\top \vee \bot$ for the same interpretation. An interpretation of $B \rightarrow C$ such that B is false and C is true can thus be written $\bot \rightarrow \top$.

2.4 Evaluating formulas

To evaluate the truth value of some formula, given an interpretation of that formula, we can simply make use of the definitions of the logical connectives and replace the innermost compound formulas with their respective truth value in a sequence. An innermost formula is a formula whose components consist of prime formulas and propositional connectives. In the formula $(P \vee \neg Q) \wedge (P \vee Q)$, the formulas $P \vee \neg Q$ and $P \vee Q$ are innermost formulas.

Rules for reducing an interpretation of the prime formulas of an innermost formula to a single truth value:

1. A negation reduces to $\top$ if $\neg\bot$, and to $\bot$ if $\neg\top$.
2. A conjunction reduces to $\top$ if $\top \wedge \top$, otherwise it reduces to $\bot$.
3. A disjunction reduces to $\bot$ if $\bot \vee \bot$, otherwise it reduces to $\top$.
4. A conditional reduces to $\bot$ if $\top \rightarrow \bot$, otherwise it reduces to $\top$.
5. A biconditional reduces to $\top$ if $\top \leftrightarrow \top$ or $\bot \leftrightarrow \bot$, otherwise it reduces to $\bot$.

Let the symbol $\Leftrightarrow$ stand for truth-value equivalent, then for an interpretation of the above formula such that P is true and Q is true, we can evaluate the truth value of the complete formula in a few steps, like so:

$$(\top \vee \neg\top) \wedge (\top \vee \top) \Leftrightarrow (\top \vee \bot) \wedge \top \Leftrightarrow \top \wedge \top \Leftrightarrow \top. \tag{1}$$

2.5 Reduced sets of logical connectives

It turns out we can represent all formulas representable with the connectives already presented (i.e., '$\neg$', '$\wedge$', '$\vee$', '$\rightarrow$', and '$\leftrightarrow$') using only '$\neg$' and '$\wedge$', or '$\neg$' and '$\vee$'. Consider the following list of truth-value equivalents.

1. $A \wedge B \Leftrightarrow \neg(\neg A \vee \neg B)$

2. $A \vee B \Leftrightarrow \neg(\neg A \wedge \neg B)$

3. $A \rightarrow B \Leftrightarrow \neg A \vee B \Leftrightarrow \neg(A \wedge \neg B)$

4. $A \leftrightarrow B \Leftrightarrow \neg(\neg A \vee \neg B) \vee \neg(A \vee B) \Leftrightarrow \neg(A \wedge \neg B) \wedge \neg(\neg A \wedge B)$

Using these equivalents, we can take any formula constructed using the original connectives and turn it into a truth-value equivalent formula using only negation and conjunction or negation and disjunction.

Yet another possibility is using the conditional together with false (i.e., '$\rightarrow$' and '$\bot$'), based on the following equivalences:

1. $\neg A \Leftrightarrow A \rightarrow \bot$

2. $A \wedge B \Leftrightarrow (A \rightarrow (B \rightarrow \bot)) \rightarrow \bot$

3. $A \vee B \Leftrightarrow (A \rightarrow B) \rightarrow B$

4. $A \leftrightarrow B \Leftrightarrow ((A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow \bot)) \rightarrow \bot$

2.6 Simplification

Some formulas can be readily simplified by substituting them with equivalent formulas that are shorter (i.e., have fewer letters representing formulas and/or fewer connectives). The following equivalences represent ways of simplifying formulas.

1. $\neg\neg P \Leftrightarrow P$

2. $P \wedge P \Leftrightarrow P$

3. $P \vee P \Leftrightarrow P$

4. $Q \wedge P \Leftrightarrow P \wedge Q \Leftrightarrow \neg(\neg P \vee \neg Q)$

5. $Q \vee P \Leftrightarrow P \vee Q \Leftrightarrow \neg(\neg P \wedge \neg Q)$

6. $P \rightarrow Q \Leftrightarrow \neg P \vee Q \Leftrightarrow \neg(P \wedge \neg Q)$

7. $P \wedge (Q \wedge R) \Leftrightarrow (P \wedge Q) \wedge R$

8. $P \vee (Q \vee R) \Leftrightarrow (P \vee Q) \vee R$

9. $\neg(P \land Q) \Leftrightarrow \neg P \lor \neg Q$

10. $\neg(P \lor Q) \Leftrightarrow \neg P \land \neg Q$

11. $P \land \neg P \Leftrightarrow \bot$

12. $P \land \top \Leftrightarrow P$

13. $P \land \bot \Leftrightarrow \bot$

14. $(P \land Q) \lor (P \land \neg Q) \Leftrightarrow P$

15. $(P \lor Q) \land (P \lor \neg Q) \Leftrightarrow P$

Furthermore, for any formulas A and B:

1. B is a valid consequence of A if and only if $A \Leftrightarrow A \land B$

2. B is a valid consequence of A if and only if $B \Leftrightarrow A \lor B$

2.7 Normal forms

Normal forms involving only negation, conjunction, and disjunction can make it easier to evaluate the truth value of a formula given a certain interpretation, and spot inconsistencies, relative to its original counterpart. The normal form of a formula is a formula that is truth-value equivalent to the original formula but follows a particular type of structure and is formed only with the help of negation, conjunction, and disjunction. Furthermore, only prime formulas are negated, so any formula in normal form consists of conjunctions and disjunctions of possibly negated prime formulas, also referred to as literals.

Definition 2.2. A *literal* is a prime formula P, or its negation $\neg P$.

Converting a formula into normal form starts with two overarching steps: (1) move any negations inwards so that only prime formulas are negated, and (2) convert any conditional and biconditional into a truth-value equivalent formula with only conjunctions and/or disjunctions. After that, conjunctions and disjunctions will possibly be altered, depending on the sought normal form.

To move the negations of a formula inwards, use the following truth-value equivalents:

1. $\neg\neg A \Leftrightarrow A$

2. $\neg(A \land B) \Leftrightarrow \neg A \lor \neg B$

3. $\neg(A \lor B) \Leftrightarrow \neg A \land \neg B$

4. $\neg(A \rightarrow B) \Leftrightarrow A \land \neg B$

5. $\neg(A \leftrightarrow B) \Leftrightarrow \neg A \leftrightarrow B \Leftrightarrow A \leftrightarrow \neg B$

Any remaining conditional or biconditional can then be eliminated by observing the following:

1. $A \rightarrow B \Leftrightarrow \neg A \lor B$

2. $A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \land (B \rightarrow A)$

We may need to distribute a conjunction over a disjunction, or vice versa, and would then make use of the *distributive laws*:

1. $A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$
2. $A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$

Lastly, we should mention the term *clause*, which is used to describe groupings of conjunctions or disjunctions depending on the normal form used. The meaning of the term will, in each instance, be clear from the context.

2.7.1 Disjunctive normal form

This type of normal form consists of disjunctions of clauses, where a clause is either a literal or a conjunction of literals. In the formula $\neg P \vee (Q \wedge \neg R)$, which is in disjunctive normal form, $\neg P$ is a clause, and so is $Q \wedge \neg R$.

To transform a formula into disjunctive normal form

1. Move all negations inward until only prime formulas are negated.
2. Turn all conditionals and biconditionals into the equivalent disjunctions and/or disjunctions.
3. Distribute any conjunctions over disjunctions so that any conjunction is a conjunction only of literals.
4. Simplify the clauses as needed to that each clause contains at most one instance of each formula letter, i.e., replace any clause containing a contradiction with $\perp$, and remove duplicates so that a clause $P \wedge P \wedge \ldots$ becomes $P \wedge \ldots$.
5. If at least one clause which is not $\perp$ remains, then remove all the $\perp$s. Otherwise, remove all $\perp$s but one.

An additional, but not necessary, step is to simplify the resulting formula using the simplification rules outlined in the corresponding section above. If only the clause $\perp$ is left, then the whole formula is inconsistent.

Developed disjunctive normal form If every clause in a formula on disjunctive normal form contains each of the prime formulas, negated or not, then each clause will describe an interpretation of the prime formulas that makes the whole formula true. Hence, a formula such as $(A \wedge \neg B \wedge C) \vee (A \wedge \neg B \wedge \neg C)$ will be true for the following two interpretations:

1. A is true, B is false, and C is true.
2. A is true, B is false, and C is false.

A formula on disjunctive normal form can be developed until each clause contains all of the letters for the prime formulas by successive applications of the equivalence $P \Leftrightarrow (P \wedge Q) \vee (P \wedge \neg Q)$.

If the developed disjunctive normal form of two different formulas contain the exact same clauses, then they are equivalent. To test if a formula B is a valid consequence of another formula A, transform both into developed disjunctive normal form. If all the clauses of A are among the clauses of B, then whenever A is true, B will be true, and thus B is a valid consequence of A.

2.7.2 Conjunctive normal form

Formulas on conjunctive normal form consist of conjunctions of clauses, where a clause is a literal or a disjunction of literals. The formula $(P \vee Q \vee R) \wedge (S \vee T)$ is on conjunctive normal form, and its clauses are $P \vee Q \vee R$ and $S \vee T$.

Note that some formulas are both on disjunctive and conjunctive normal form. For example, the formula $\neg P \vee Q \vee R$ is on both disjunctive and conjunctive normal form. If considered to be on disjunctive normal form, then it has three clauses, name $\neg P$, Q, and R. On the other hand, it only has the single clause $\neg P \vee Q \vee R$ if considered to be on conjunctive normal form.

To transform a formula into conjunctive normal form, follow these steps:

1. Move all negations inward to that only prime formulas are negated.

2. Transform any conditional or biconditional into their equivalent disjunctions and conjunctions.

3. Distribute any disjunction over conjunctions so that any disjunction is a disjunction of literals.

4. Simplify the clauses such that each clause contains at most one instance of each formula letter, i.e., a clause with $P \vee P \vee \ldots$ becomes $P \vee \ldots$, and a clause with $P \vee \neg P \vee \ldots$ become $\top$. If at least one clause distinct from $\top$ remains, then remove all the clauses with $\top$. Otherwise, remove all but one of the $\top$s.

The resulting formula can be simplified using the rules in the section on simplification. If only the clause $\top$ remains, then the whole formula is true under any interpretation.

Developed conjunctive normal form A formula on developed conjunctive normal form has every prime formula, whether negated or not, represented in each of its clauses. Thus, the formula $A \wedge (A \vee \neg B)$ is not on developed conjunctive normal form, but $(A \vee B) \wedge (A \vee \neg B)$ is.

To develop a formula on conjunctive normal form, apply the equivalence $P \Leftrightarrow (P \vee Q) \wedge (P \vee \neg Q)$ successively until each prime formula is present in all of the clauses, negated or unnegated.

Testing whether a formula B is a valid consequence of another formula A (i.e. if B is true under all interpretations under which A is true) can be performed as follows: Transform both formulas into developed conjunctive normal form. If all the clauses of B also are clauses of A, then B is a valid consequence of A. As a result, if A and B have the exact same clauses, then they are equivalent.

2.8 Duality

Let A and A' be two formulas constructed from the prime formulas $P_1, \ldots, P_n$, then if each prime formula P_i has the opposite truth value in A' in comparison to what it has in A, and A' evaluates to the opposite truth value of A, then A' is the dual of A.

From the above, we can conclude that the conjunction and disjunction are duals. Suppose the formula A is $P \wedge Q$, and A' is $P \vee Q$, namely A but with the conjunction switched to a disjunction. Opposite interpretations of A and A' yields opposite truth values for A and A', for all interpretations:

1. $\top \wedge \top \Leftrightarrow \top$ and $\bot \vee \bot \Leftrightarrow \bot$
2. $\top \wedge \bot \Leftrightarrow \bot$ and $\bot \vee \top \Leftrightarrow \top$
3. $\bot \wedge \top \Leftrightarrow \bot$ and $\top \vee \bot \Leftrightarrow \top$
4. $\bot \wedge \bot \Leftrightarrow \bot$ and $\top \vee \top \Leftrightarrow \top$

Hence, A and A' are duals, and it turns out that negation is the dual of itself (and by the same token any prime formula is the dual of itself), as shown by $\neg\top \Leftrightarrow \bot$ and $\neg\bot \Leftrightarrow \top$.

As a consequence, we can construct the dual of a formula consisting only of negations, conjunction, and disjunctions by interchanging conjunctions and disjunctions. For example, the dual of $A \wedge (\neg B \vee A)$ is $A \vee (\neg B \wedge A)$. More generally, if A is a formula constructed from negations, conjunctions, and disjunctions, and A' is derived from A in such a way that every conjunction is replaced by a disjunction, and vice versa, and a negation is tacked onto every prime formula (i.e., P turns into $\neg P$, and $\neg P$ turns into $\neg\neg P$ which is equivalent to P), then $\neg A \Leftrightarrow A'$.

Furthermore, if B' is the dual of B, and C' is the dual of C, then (1) if $\neg B$ is true under all interpretations, then so is B', (2) if B is true under all interpretations, then so is $\neg B'$, (3) if B is equivalent to C, then B' is equivalent to C', and (4) if $B \to C$ is true for all interpretations, then so is $C' \to B'$.

2.9 Logical consequence

Let us start with some common definitions:

Definition 2.3. A *tautology* is a formula that evaluates to true under any interpretation.

Definition 2.4. A *contradiction* is a formula that evaluates to false under any interpretation.

Definition 2.5. A formula that is neither a tautology nor a contradiction is a *contingent* formula, i.e., its truth value is contingent on the interpretation.

Common examples of tautologies are $A \vee \neg A$ and $A \to A$–they will evaluate to true regardless of the truth value assigned to A. Another example is $(A \wedge \neg A) \to B$, which evaluates to true under all interpretations because the antecedent $A \wedge \neg A$ is a contradiction. A conditional with a false antecedent evaluates to true independent of the truth value of its consequent.

Definition 2.6. A formula B is a logical consequence of A if and only if the conditional $A \to B$ is a tautology.

Note that logical consequence is something we say about formulas at the meta-level. Hence, logical consequence is not something that has a truth value. To symbolize logical consequence, we use the semantic turnstile '$\models$', and instead of logical consequence we speak about entailment. To the left of the turnstile we write the set of formulas that *entails* the formula to the right of the turnstile. The expression

$$A_1, \ldots, A_n \models B$$

reads 'A_1 thru A_n entails B.'

Sometimes we will use a more compact way of writing sets of formulas using a caligraphic style of letters such as $\mathcal{A}$. In such a case we write

$$\mathcal{A} \models \mathcal{B}$$

to say that the formulas in $\mathcal{A}$ entails B. To reiterate, it means that any interpretation that makes all the formulas in $\mathcal{A}$ true also makes B true.

The turnstile is also used to symbolize tautologies and contradictions, but when written with the turnstile we will speak about valid formulas rather than tautologies. Note that this use of valid is different from valid in speaking about valid arguments. The expression

$$\models B$$

is pronounced 'B is valid,' and

$$\mathcal{A} \models$$

indicates that no interpretation makes every formula in $\mathcal{A}$ true, and is pronounced '$\mathcal{A}$ is inconsistent.'

Since entailment describes the validity of the corresponding conditional, we can test whether $A \models B$ by checking whether $A \to B$ is valid. As you may remember from the previous chapter, an argument is valid if and only if its counterexample set is inconsistent. Hence, one way to test whether $A \to B$ is valid is to test whether some interpretation makes $A \wedge \neg B$ true. To do that we consider the interpretations that make $\neg B$ true and evaluate A under each of them.

The following can be useful when trying to determine entailment:

1. $A \models B$ if and only if $A \to B$
2. $A_1, \ldots, A_n \models B$ if and only if $A_1, \ldots, A_{n-1} \models A_n \to B$
3. $A_1, \ldots, A_n \models B$ if and only if $\models A_1 \to (\ldots(A_n \to B)\ldots)$
4. $A_1, \ldots, A_n \models B$ if and only if $\models A_1 \wedge \cdots \wedge A_n \to B$

In $\models B$, the set of premises is empty; in $\mathcal{A} \models$, the consequent is empty. Again, entailment is the validity of the corresponding conditional. Hence, we can think of $\models B$ as $\to B$ and $\mathcal{A} \models$ as $A_1 \wedge \cdots \wedge A_n \to$, i.e., a conditional with an empty antecedent and an empty consequent respectively.

A conditional with a true antecedent has to have a true consequent to be true, and a conditional with a false consequent has to have a false antecedent to be true. Hence, if we replace the empty antecedent with $\top$, then B has to be true for the resulting conditional $\top \to B$ to be true. Similarly, if we replace the empty consequent with $\bot$, then the conjunction of the premises $A_1, \ldots, A_n \in \mathcal{A}$ has to be false for $A_1 \wedge \cdots \wedge A_n \to \bot$ to be true.

If $\models B$ corresponds to $\top \to B$, one would think it should be $\top \models B$ instead—and similarly $\mathcal{A} \models \bot$ for $A_1 \wedge \cdots \wedge A_n \to \bot$. However, the empty set of premises is seen as an empty conjunction, and the empty conclusion is seen as an empty disjunction. Furthermore, the empty conjunction and the empty disjunction are assigned truth values as follows.

1. A conjunction is true if and only if none of its conjuncts is false. If the conjunction is empty, then none of the conjuncts is false because there are none. Hence, the empty conjunction is vacuously true.

2. A disjunction is true if and only if it has at least one true disjunct. If the disjunction is empty, no disjunct can be true, and therefore the empty disjunction is false.

In conclusion, $\models B$ means that B is valid because an empty conjunction (of the premises) as antecedent in the corresponding conditional is truth-value equivalent to $\top \to B$. The opposite, $\mathcal{A} \models$, that $\mathcal{A}$ is an invalid set of premises because the corresponding conditional with an empty disjunction as consequent is truth-value equivalent to $A_1 \wedge \cdots \wedge A_n \to \bot$, where $A_1, \ldots, A_n \in \mathcal{A}$.

3 Deduction in propositional logic

As we have already seen, it is possible to check the validity, consistency, or inconsistency of a propositional logic formula simply by enumerating all the possible interpretations and evaluating the formula's truth value for each interpretation. Such an evaluation is based on the semantics of the connectives, i.e., the conjunction $A \wedge B$ is true if and only if both A and B are true, etc. The number of interpretations grows exponentially with the number of prime formulas–it is 2^n where n is the number of propositions. While it is a reasonably small number for two ($2^2 = 4$) to three ($2^3 = 8$), or even four ($2^4 = 16$), prime formulas, it quickly becomes forbiddingly large.

There are ways of restricting the number of interpretations that must be checked, and some of those ways could be implemented algorithmically as long as we deal with propositional logic. However, once we need to check the validity of formulas in first-order logic, it will sometimes be impossible to enumerate all possibilities, and so-called deductive systems will prove handy–this will become evident in subsequent chapters. Furthermore, the rather mechanical nature of some deductive systems makes them suitable for implementing programs for automated reasoning.

To better understand the fundamentals of deductive systems, we will look at three such systems for propositional logic–we will use these for first-order logic as well, but with some minor additions. They are semantic tableaux, resolution, and natural deduction. Before that, we will look closer at the turnstile symbol and its variations to differentiate between semantic and syntactic reasoning.

When reasoning based on the semantics of propositional logic, we have used the double turnstile $\models$ to indicate that a formula is a logical consequence of some set of formulas. For example, if a formula B is a logical consequence of a set of formulas $\mathcal{A}$, we would write that as $\mathcal{A} \models B$. There are different ways of expressing the turnstile in English, but we will go with 'entails,' and for $\mathcal{A} \models B$ we would say that $\mathcal{A}$ entails B. If $\mathcal{A}$ is the set of premises of an argument with B as its conclusion, it would be the same as saying that B is a logical consequence of $\mathcal{A}$. However, sometimes there will be no set of formulas to the left of the turnstile, as in $\models A$, which expresses that A is a theorem. Seldom, if ever, will we encounter arguments with no premises, and so interpreting the turnstile to describe logical consequence only sometimes captures its actual meaning. However, the word entails also seems like an odd way of expressing $\models$ when there is nothing to the right of it, as in $A \models$, meaning that A is inconsistent.

The solution to the above issue of finding exactly one way of expressing the turnstile as part of an expression about formulas is the following. If

$$\mathcal{A} \models B,$$

then we say, ' $\mathcal{A}$ entails B.' If

$$\models B$$

we say that 'B is valid,' and if

$$A \models$$

we say that 'A is inconsistent.'

Given a deductive system, instead of using the double turnstile (which indicates the use of semantics), the (syntactic) turnstile $\vdash$ would be used, and we would speak about derivability rather than entailment. Thus, $\mathcal{A} \vdash B$ would mean that 'B is derivable from $\mathcal{A}$.' If $\vdash A$, then 'A is a proof' (in a given deductive system), and if $\mathcal{A} \vdash$, then $\mathcal{A}$ is inconsistent. If the deductive system is not clear from context, we may index the turnstile, such as $\vdash_S$ where S stands for 'semantic tableaux,' to make the deductive system explicit.

3.1 Semantic tableaux

The basic idea behind the semantic tableaux method (also known as truth trees) is to create a branch that represents the combination of prime formulas needed to make the set of top-level conjuncts of a formula true. The top-level conjuncts would, for example, be the premises and conclusion of an argument. In the formula $A \land B$, A and B are the top-level conjuncts. Given $(A \to B \land C) \land (A \lor (\neg B \lor C)) \land B$, the top-level conjuncts are $A \to B \land C$ and $A \lor (\neg B \lor C)$ and B. That does not presuppose anything in particular about the structure of the formula. A formula $A \to B$ has only one top-level conjunct, namely the formula itself.

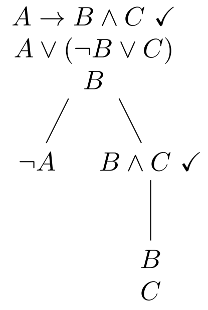
Referring to top-level conjuncts is not a necessity but can be helpful in reasoning about the truth about formulas in relation to semantic tableaux. Let us look closer at one of the previously mentioned formulas and ask: Is there an interpretation that makes $(A \to B \land C) \land (A \lor (\neg B \lor C)) \land B$ true? Well, first of all, for that formula to be true, all the top-level conjuncts would have to be true, so let us list them:

$$\begin{array}{c} A \to B \land C \\ A \lor (\neg B \lor C) \\ B \end{array}$$

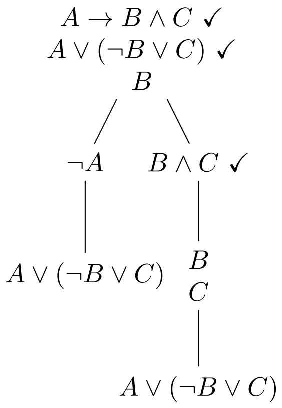
Clearly, B has to be true; there is nothing more to say. But, the first and second premises are not so straight forward. Based on the equivalence $P \to Q \Leftrightarrow \neg P \lor Q$, the first premise is true if A is false (thus $\neg A$ true), or $B \land C$ is true. We attempt to model truth, and therefore we refer to $\neg A$ as true rather than to A as false. Let us start constructing a tree to model that:

$$\begin{array}{ccc} & A \to B \land C \checkmark & \\ & A \lor (\neg B \lor C) & \\ & B & \\ / & & \backslash \\ \neg A & & B \land C \end{array}$$

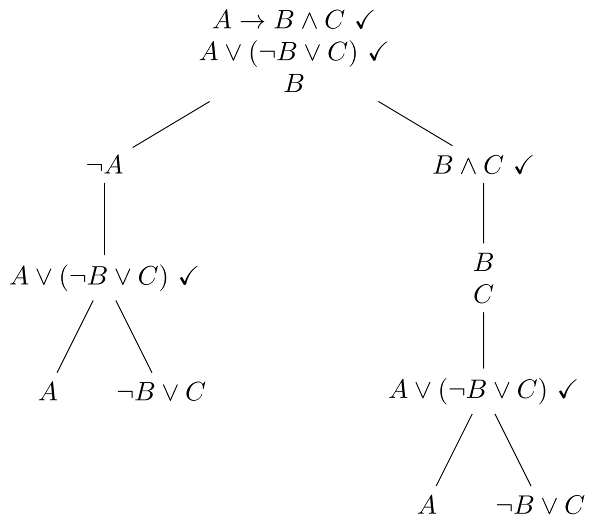
The checkmark after the first formula is there as a reminder that we have dealt with it. The formula $B \wedge C$ is true only when both B and C are true, and we can extend that branch of the tree to model that in the same way we listed the top-level conjuncts:



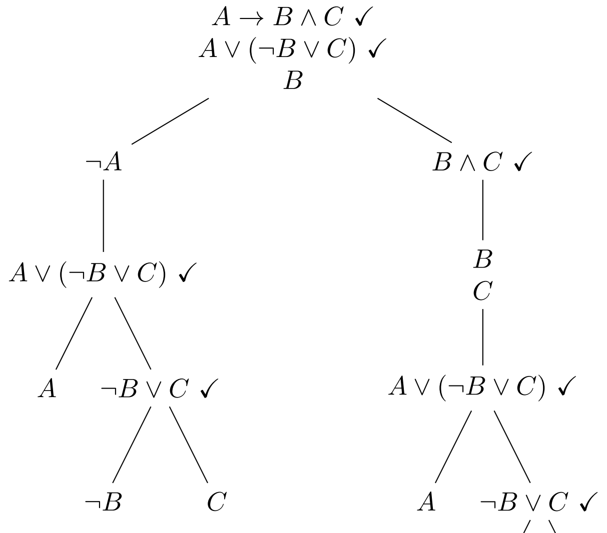
Let us now turn to the second formula at the top. Each branch of the tree represents a possible interpretation that would make the original formula true–a branch may turn out to contain contradicting nodes, such as A and $\neg A$. Hence we speak about a "possible" interpretation. Consequently, that second formula would have to be true for each of the branches. We will not take this somewhat unnecessary step in the future, but let us model what was just said for clarity:



For each of the leaves of the tree, either A or $\neg B \vee C$ has to be true. We split each of the branches to model each of these cases:



At this point, there are only the leaves containing $\neg B \vee C$ left. Since it is a disjunction, they have to branch out into the two cases that would make the disjunction true:



Now, consider each of the branches, from left to right. Which ones represent a possible interpretation that would make the original formula true?

The first branch contains a contradiction–it suggests an interpretation where both $\neg A$ and A are true at the same time, which is impossible, at least in propositional logic. This phenomenon, when a branch contains a prime formula as well as its negation, is referred to as a branch that *closes*. Later on, we will represent that with the $\perp$ symbol. The second branch also contains a contradiction, but this time involving B and $\neg B$. The third branch does, however, suggest a working interpretation with B true, A false, and C true. The fourth branch also represents a working interpretation with all of A, B, and C true. The fifth branch contains a contradiction B and $\neg B$, but the sixth branch suggests a third working interpretation where A is either true or false (since it is not present on that branch at all), and B and C are both true.

Let us verify the three interpretations suggested by the semantic tableau. Remember the original formula:

$$(A \to B \wedge C) \wedge (A \vee (\neg B \vee C)) \wedge B$$

For the first, A false, B true, and C true, we have

$$\begin{aligned}(\bot \to \top \wedge \top) \wedge (\bot \vee (\neg\top \vee \top)) \wedge \top &\Leftrightarrow (\bot \to \top) \wedge (\bot \vee \top) \wedge \top\\ &\Leftrightarrow \top \wedge \top \wedge \top\\ &\Leftrightarrow \top.\end{aligned}$$

The second, all of A, B, and C true, yields

$$\begin{aligned}(\top \to \top \wedge \top) \wedge (\top \vee (\neg\top \vee \top)) \wedge \top &\Leftrightarrow (\top \to \top) \wedge (\top \vee \top) \wedge \top\\ &\Leftrightarrow \top \wedge \top \wedge \top\\ &\Leftrightarrow \top.\end{aligned}$$

The third, B and C true, gives us

$$\begin{aligned}(A \to \top \wedge \top) \wedge (A \vee (\neg\top \vee \top)) \wedge \top &\Leftrightarrow (A \to \top) \wedge (A \vee \top) \wedge \top\\ &\Leftrightarrow \top \wedge \top \wedge \top\\ &\Leftrightarrow \top.\end{aligned}$$

Each of the three interpretations yielded a true formula, as expected. In fact, since A is the only prime formula that changes truth value, it would have been enough to evaluate the last interpretation. It covers the previous two. Any interpretation involving B set to false would result in a false outcome because B on its own is a top-level conjunct.

While we are at it, let us turn the formula into disjunctive normal form:

$$\begin{aligned}(A \to B \wedge C) \wedge (A \vee (\neg B \vee C)) \wedge B &\Leftrightarrow (\neg A \vee (B \wedge C)) \wedge (A \vee (\neg B \vee C)) \wedge B\\ &\Leftrightarrow (\neg A \vee (B \wedge C)) \wedge ((A \wedge B) \vee ((\neg B \wedge B) \vee (C \wedge B)))\\ &\Leftrightarrow (\neg A \vee (B \wedge C)) \wedge ((A \wedge B) \vee (\bot \vee (C \wedge B)))\\ &\Leftrightarrow (\neg A \vee (B \wedge C)) \wedge ((A \wedge B) \vee (C \wedge B))\\ \Leftrightarrow (((A \wedge B) \wedge \neg A) \vee ((C \wedge B) \wedge \neg A)) &\vee (((A \wedge B) \wedge (B \wedge C)) \vee ((C \wedge B) \wedge (B \wedge C)))\\ &\Leftrightarrow \bot \vee (C \wedge B \wedge \neg A) \vee (A \wedge B \wedge C) \vee (B \wedge C)\\ &\Leftrightarrow (\neg A \wedge B \wedge C) \vee (A \wedge B \wedge C) \vee (B \wedge C)\\ &\Leftrightarrow B \wedge C.\end{aligned}$$

As we can see, the last formula is the same as the last interpretation we extracted from the semantic tableau. In fact, each branch of a semantic tableau that does not contain a contradiction will have a corresponding clause in the disjunctive normal form of the same formula.

If all the branches of a semantic tableau close (i.e., contain a contradiction), no interpretation would satisfy the formula. Similarly, a formula on disjunctive normal form with only $\bot$ as the sole clause will never be true, regardless of interpretation.

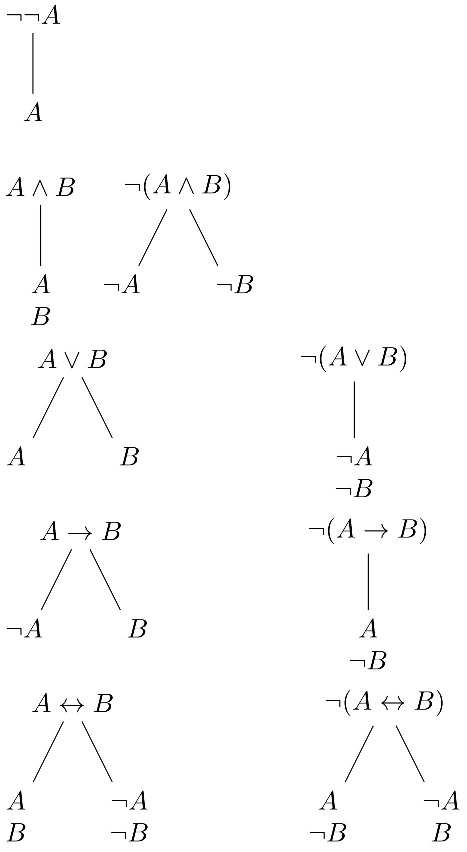
Going back to Chapter 1, we acknowledged that one way to test the validity of an argument is to create a counterexample set by taking the premises together with the negation of the conclusion. If the counterexample set is inconsistent, then the original argument is valid.

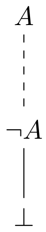
Using the method of semantic tableaux, we would start with the counterexample set and generate the tree. If all the branches close, there is no interpretation that makes the counterexample set true; thus, it is inconsistent, and we can conclude the original argument to be valid.

Using the turnstile symbol, to test if $\mathcal{A} \models B$ we create a counterexample set with $\mathcal{A}$ and $\neg B$, followed by checking if $\mathcal{A}, \neg B \vdash_S$ using a semantic tableau. The $\vdash$ symbol is indexed with S to indicate that the derivation is performed using a semantic tableau.

3.1.1 Rules

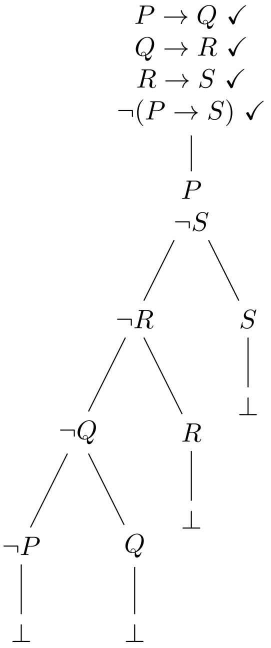
Here are the semantic tableaux rules for each of the logical connectives. They correspond to the original semantics of the connectives.





3.1.2 Example

The conclusion $P \rightarrow S$ follows from the premises $P \rightarrow Q$, $Q \rightarrow R$, and $R \rightarrow S$. Let us check that the argument indeed is valid by using the semantic tableaux method to show that the counterexample set is inconsistent.



3.2 Resolution

Assume a formula $P \wedge Q \wedge \neg Q$. Since Q and $\neg Q$ cannot be true at the same time, we have a contradiction–no interpretation will satisfy that formula, i.e., make it evaluate to true. To make it explicit, we could rewrite the formula as $P \wedge \perp$. If we can derive $\perp$ from any two conjuncts in a formula with a structure similar to the one above, then we have proved that formula to be inconsistent.

The basic idea behind the method of resolution is to successively simplify a formula on conjunctive normal form until there are at least two contradicting conjuncts, i.e., *clauses*. Any contradicting clauses indicate that the formula is inconsistent[2]. If the formula represents a counterexample set, and that counterexample set is found to be inconsistent, then the original argument is valid.

[2] Depending on the original formula, it may, or may not, be necessary to use all the clauses in order to resolve $\perp$—use only the clauses needed to arrive at a contradiction.

To simplify a formula on conjunctive normal form we apply the *resolution rule* in order to simplify the formula. That is essentially the only rule needed, and it looks like this:

$$(A \lor P), (B \lor \neg P) \vdash A \lor B$$

The clause that follows from applying the rule is called the *resolvent*. In this case, $A \lor B$ is the resolvent.

To show that the rule indeed is correct, we consider the corresponding counterexample set:

$$(A \lor P) \land (B \lor \neg P) \land \neg(A \lor B) \Leftrightarrow (A \lor P) \land (B \lor \neg P) \land \neg A \land \neg B$$

Clearly, for the formula to be true, both A and B would have to be false. But then we end up with

$$(\bot \lor P) \land (\bot \lor \neg P) \land \top \land \top \Leftrightarrow P \land \neg P$$

which is a contradiction. Hence, the resolution rule is valid.

In addition to the resolution rule, we shall also make use of $P, \neg P \vdash \bot$, i.e., any proposition along with its negation entails a contradiction. If we can derive a contradiction from the counterexample set, thereby showing that it is inconsistent, we conclude the original formula to be valid.

3.2.1 Example

Let us again show that $P \to S$ follows from $P \to Q$, $Q \to R$, and $R \to S$. Start by writing the premises and the negation of the conclusion as a conjunction:

$$P \to Q \land Q \to R \land R \to S \land \neg(P \to S). \tag{2}$$

Turn the formula into conjunctive normal form by using the equivalences

$$A \to B \Leftrightarrow \neg A \lor B$$

and

$$\neg(A \to B) \Leftrightarrow A \land \neg B.$$

That yields

$$(\neg P \lor Q) \land (\neg Q \lor R) \land (\neg R \lor S) \land P \land \neg S. \tag{3}$$

Turning the original formula into conjunctive normal form was straightforward. Now, list the given clauses in a numbered list, and apply the resolution rule to one pair of clauses at a time.

(1)	$\neg P \lor Q$	Given
(2)	$\neg Q \lor R$	Given
(3)	$\neg R \lor S$	Given
(4)	P	Given
(5)	$\neg S$	Given
(6)	Q	Resolvent 1, 4
(7)	R	Resolvent 2, 6
(8)	S	Resolvent 3, 7
(8)	$\bot$	Resolvent 5, 8

One is allowed to resolve any two clauses, whether they are both given, resolvents from previous applications of the rule, or one of each. The number of steps needed to obtain $\bot$ as the resolvent can depend on the order the clauses are resolved, and only if the original argument is valid will it be possible to obtain $\bot$. From the consistent formula $P \wedge Q \wedge R$, it will not be possible to resolve anything.

3.3 Natural deduction

Contrary to semantic tableaux and resolution, which are rather mechanical in nature, natural deduction is closer to how a human would reason based on a set of assumptions, and requires a bit of strategic thinking.

When reading the rules of natural deduction below, any caligraphic uppercase letter, e.g., $\mathcal{A}$, should be read as a set of arbitrary premises.

Here are the rules:

(Axiom) $A \vdash A$

($\wedge$I) If $\mathcal{A} \vdash A$ and $\mathcal{B} \vdash B$ then $\mathcal{A}, \mathcal{B} \vdash A \wedge B$

($\wedge$E) If $\mathcal{A} \vdash A \wedge B$ then $\mathcal{A} \vdash A$ and $\mathcal{A} \vdash B$

($\vee$I) If $\mathcal{A} \vdash A$ or $\mathcal{A} \vdash B$ then $\mathcal{A} \vdash A \vee B$

($\vee$E)* If $\mathcal{A} \vdash A \vee B$ and $\mathcal{B}, A \vdash C$ and $\mathcal{C}, B \vdash C$ then $\mathcal{A}, \mathcal{B}, \mathcal{C} \vdash C$

($\rightarrow$I)* If $\mathcal{A}, A \vdash B$ then $\mathcal{A} \vdash A \rightarrow B$

($\rightarrow$E) If $\mathcal{A} \vdash A$ and $\mathcal{B} \vdash A \rightarrow B$ then $\mathcal{A}, \mathcal{B} \vdash B$

($\leftrightarrow$I) If $\mathcal{A} \vdash A \rightarrow B$ and $\mathcal{B} \vdash B \rightarrow A$ then $\mathcal{A}, \mathcal{B} \vdash A \leftrightarrow B$

($\leftrightarrow$E) If $\mathcal{A} \vdash A \leftrightarrow B$ then $\mathcal{A} \vdash A \rightarrow B$ or $\mathcal{A} \vdash B \rightarrow A$

($\neg\neg$E) If $\mathcal{A} \vdash \neg\neg A$ then $\mathcal{A} \vdash A$

($\neg$E) If $\mathcal{A} \vdash \neg A$ and $\mathcal{B} \vdash A$ then $\mathcal{A}, \mathcal{B} \vdash \bot$

(EFQ) If $\mathcal{A}, \mathcal{B} \vdash \bot$ then $\mathcal{A}, \mathcal{B} \vdash C$

(TND)* If $\mathcal{A}, A \vdash B$ and $\mathcal{B}, \neg A \vdash B$ then $\mathcal{A}, \mathcal{B} \vdash B$

(RAA)* If $\mathcal{A}, A \vdash \bot$ then $\mathcal{A} \vdash \neg A$

(RAA')* If $\mathcal{A}, \neg A \vdash \bot$ then $\mathcal{A} \vdash A$

*These rules are so-called *discharge rules* because they discharge the assumptions that were made—some authors refer to discharging assumptions as "closing" assumptions. In $\vee$E, the assumptions A and B are discharged. In $\rightarrow$I, the assumption A is discharged, but not forgotten as it becomes the antecedent of a conditional instead of being a premise. In TND the assumptions A and $\neg A$ are removed because the truth value of B is not affected by the truth value of A. In the RAA rule, if the assumption A yields a contradiction, then we can derive its negation and scrap the assumption A since it was only used for testing—the RAA' rule works similarly.

3.3.1 Examples

Example 1 True to tradition, let us show that $P \to S$ follows from $P \to Q$, $Q \to R$, and $R \to S$. We start by listing the premises as assumptions and then see if we can find a way of arriving at the conclusion using the rules presented above.

In the listing below there are three columns. The leftmost column contains, for each line, the assumptions the line is derived from; any assumption is considered to be derived from itself. The middle column is the line numbers. The rightmost column contains the assumptions and what have been derived, suffixed by a label spelling out whether the line is an assumption or the deduction rule used to derive it.

{1}	(1)	$P \to Q$	Premise
{2}	(2)	$Q \to R$	Premise
{3}	(3)	$R \to S$	Premise
{4}	(4)	P	Assumption
{1, 4}	(5)	Q	$\to$E 1, 4
{1, 2, 4}	(6)	R	$\to$E 2, 5
{1, 2, 3, 4}	(7)	S	$\to$E 3, 6
{1, 2, 3}	(8)	$P \to S$	$\to$I 4-7

Example 2 Let us now turn to an example that requires a considerably more involved derivation. We will prove the following:

$$\neg A, B \vee C, B \to A, (\neg B \wedge C) \to D \models D.$$

The trick will be to derive $\neg B \wedge C$ as D then follows quite trivially. Here is the derivation. We will look at it line by line further down.

$\{1\}$	(1)	$\neg A$	Premise
$\{2\}$	(2)	$B \vee C$	Premise
$\{3\}$	(3)	$B \to A$	Premise
$\{4\}$	(4)	$(\neg B \wedge C) \to D$	Premise
$\{5\}$	(5)	B	Assumption (to be discharged by RAA)
$\{3, 5\}$	(6)	A	$\to$E 3, 5
$\{1, 3, 5\}$	(7)	$\bot$	$\neg E$ 1, 6
$\{1, 3\}$	(8)	$\neg B$	RAA 5-7
$\{9\}$	(9)	B	Assumption (to be discharged by $\vee$E)
$\{1, 3, 9\}$	(10)	$\bot$	$\neg$E 8, 9
$\{1, 3, 9\}$	(11)	C	EFQ 10
$\{12\}$	(12)	C	Assumption (to be discharged by $\vee$E)
$\{1, 2, 3\}$	(13)	C	$\vee$E 2, 9-11, 12
$\{1, 2, 3\}$	(14)	$\neg B \wedge C$	$\wedge$I 8, 13
$\{1, 2, 3, 4\}$	(15)	D	$\to$E 4, 14

Lines 1-4 contain the premises. A premise depends only on itself. Therefore, only the corresponding line number is written in the leftmost column on these lines.

Line 5 is an assumption made with the intent of deriving a contradiction–a bit of forethought is sometimes required when trying to derive formulas using natural deduction. As with premises, an assumption depends only on itself, as shown in the leftmost column.

Line 6 stems from an application of the →E rule. We have $(B \to A)_3 \vdash (B \to A)_3$ and $B_5 \vdash B_5$ which yields $(B \to A)_3, B_5 \vdash A_{3,5}$. For clarity, the assumptions upon which the premises rest are included as subscripts, but that is only for the purpose of this explanation. In the rightmost column, we see that the result follows immediately from lines 3 and 5—there is a difference between the set of assumptions and the lines to which a rule is applied.

Line 7 is an application of the ¬E rule, setting the conditions for the following application of the reductio ad absurdum (RAA) rule. It is based on $\neg A_1 \vdash \neg A_1$ and $(B \to A)_3, B_5 \vdash A_{3,5}$ which yield $\neg A_1, (B \to A)_3, B_5 \vdash \bot_{1,3,5}$.

Line 8 is an application of RAA. It is based on $\neg A_1, (B \to A)_3, B_5 \vdash \bot_{1,3,5}$ which result in $\neg A_1, (B \to A)_3 \vdash \neg B_{1,3}$. The line follows from line 5 through 7. At this point, the assumption on line 5 was *discharged* by applying RAA. We made an assumption, arrived at a contradiction, and concluded that the assumption was false. Looking at the definition of the RAA rule, we see that the resulting sequent $\mathcal{A} \vdash \neg A$ does not contain the premise A, the premise that was assumed, only its negation.

Line 9 is another assumption of B so that we can arrive at a contradiction and be able to apply EFQ. We do this in preparation for a subsequent application of ∨E on line 13.

Line 10 is the result of applying ¬E to lines 8 and 9. We have $B_9 \vdash B_9$ and $\neg B_{1,3} \vdash \neg B_{1,3}$ which give $B_9, \neg B_{1,3} \vdash C_{1,3,9}$.

Line 11 is the result of applying ex falso quodlibet (EFQ) to line 10—anything follows from a contradiction and since we want to go from $B \vee C$ to C when we later apply ∨E, we opt for C here.

On line 12, the assumptions of C is also for the coming application of ∨E—we want to derive C from $B \vee C$. Note that we already here have $C_{11} \vdash C_{11}$ by the axiom.

Line 13 is an application of the ∨E rule, which may look a bit complicated. Essentially it says that if we have $B \vee C$, and C can be derived from both B and C (separately), then we can eliminate $B \vee C$ and replace it with C. In this particular case, we have $(B \vee C)_2 \vdash (B \vee C)_2$, and $\neg B_{1,3}, B_5 \vdash C_{1,3,9}$, and $C_{11} \vdash C_{11}$ which yields $(B \vee C)_2, \neg B_{1,3} \vdash C_{1,2,3}$ from a combination of the lines 2, 9 through 10, 12.

Line 14 is a straightforward application of the ∧I rule. From $\neg B_{1,3} \vdash \neg B_{1,3}$ and $C_{1,2,3} \vdash C_{1,2,3}$ we obtain $\neg B_{1,3}, C_{1,2,3} \vdash (\neg B \wedge C)_{1,2,3}$ from lines 8 and 13.

Line 15 results in the sought conclusion D by applying the →E rule where $((\neg B \wedge C) \to D)_4 \vdash ((\neg B \wedge C) \to D)_4$ and $(\neg B \wedge C)_{1,2,3} \vdash (\neg B \wedge C)_{1,2,3}$ yields $(\neg B \wedge C)_{1,2,3}, (\neg B \wedge C) \to D)_4 \vdash D_{1,2,3,4}$ from lines 4 and 12.

The conclusion in the last line rests upon all four original premises. Hence, no premise was unnecessary, and the assumptions made during the derivation were both discharged.

Example 3 This example may look easy at first, but involves a considerable longer derivation than the previous examples. We will show that $\neg D$ follows from $D \to W$, $A \vee \neg W$, and $\neg(D \wedge A)$.

{1}	(1)	$D \to W$	Premise
{2}	(2)	$A \vee \neg W$	Premise
{3}	(3)	$\neg(D \wedge A)$	Premise
{4}	(4)	$\neg(\neg D \vee \neg A)$	Assumption (to be discharged by RAA')
{5}	(5)	$\neg D$	Assumption (to be discharged by RAA')
{5}	(6)	$\neg D \vee \neg A$	∨I 5
{4,5}	(7)	$\bot$	¬E 4, 6
{4}	(8)	D	RAA' 5-7
{9}	(9)	$\neg A$	Assumption (to be discharged by RAA')
{9}	(10)	$\neg D \vee \neg A$	∨I 9
{4,9}	(11)	$\bot$	¬E 4, 10
{4}	(12)	A	RAA' 9-11
{4}	(13)	$D \wedge A$	∧I 8, 12
{3,4}	(14)	$\bot$	¬E 3, 13
{3}	(15)	$\neg D \vee \neg A$	RAA' 4-14
{16}	(16)	D	Assumption (to be discharged by →I)
{1,16}	(17)	W	→E 1, 16
{18}	(18)	$\neg W$	Assumption (to be discharged by ∨E)
{1,16,18}	(19)	A	EFQ 17, 18
{20}	(20)	A	Assumption (to be discharged by ∨E)
{1,2,16}	(21)	A	∨E 2, 18-19, 20
{1,2}	(22)	$D \to A$	→I 16, 21
{23}	(23)	$\neg A$	Assumption (to be discharged by ∨E)
{24}	(24)	D	Assumption (to be discharged by RAA)
{1,2,24}	(25)	A	→E 22, 24
{1,2,23,24}	(26)	$\bot$	¬E 23, 25
{1,2,23}	(27)	$\neg D$	RAA 24-26
{28}	(28)	$\neg D$	Assumption (to be discharged by ∨E)
{1,2,3}	(29)	$\neg D$	∨E 15, 23-27, 28

Example 4 Show that D follows from $(A \to B) \to C$, $\neg B \leftrightarrow C$, $A \to B$, and $(C \wedge \neg A) \to D$.

$\{1\}$	(1)	$(A \to B) \to C$	Premise
$\{2\}$	(2)	$\neg B \leftrightarrow C$	Premise
$\{3\}$	(3)	$A \to B$	Premise
$\{4\}$	(4)	$(C \wedge \neg A) \to D$	Premise
$\{1, 3\}$	(5)	C	→E 1, 3
$\{2\}$	(6)	$C \to \neg B$	↔E 2
$\{1, 2, 3\}$	(7)	$\neg B$	→E 5, 6
$\{8\}$	(8)	A	Assumption (to be discharged by RAA)
$\{3, 8\}$	(9)	B	→E 3, 8
$\{1, 2, 3, 8\}$	(10)	$\bot$	¬E 7, 9
$\{1, 2, 3\}$	(11)	$\neg A$	RAA 8-10
$\{1, 2, 3\}$	(12)	$C \wedge \neg A$	∧I 5, 11
$\{1, 2, 3, 4\}$	(13)	D	→E 4, 12

3.3.2 Some useful derivations

$\neg(A \wedge B) \vdash \neg A \vee \neg B$

$\{1\}$	(1)	$\neg(A \wedge B)$	Premise
$\{2\}$	(2)	$\neg(\neg A \vee \neg B)$	Assumption (to be discharged by RAA')
$\{3\}$	(3)	$\neg A$	Assumption (to be discharged by RAA')
$\{3\}$	(4)	$\neg A \vee \neg B$	$\vee$I 3
$\{2,3\}$	(5)	$\bot$	$\neg$E 2, 4
$\{2\}$	(6)	A	RAA' 3-5
$\{7\}$	(7)	$\neg B$	Assumption (to be discharged by RAA')
$\{7\}$	(8)	$\neg A \vee \neg B$	$\vee$I 7
$\{2,7\}$	(9)	$\bot$	$\neg$E 2, 8
$\{2\}$	(10)	B	RAA' 7-9
$\{2\}$	(11)	$A \wedge B$	$\wedge$I 6, 10
$\{1,2\}$	(12)	$\bot$	$\neg$E 1, 11
$\{1\}$	(13)	$\neg A \vee \neg B$	RAA' 2-12

$\neg A \vee \neg B \vdash \neg(A \wedge B)$

$\{1\}$	(1)	$\neg A \vee \neg B$	Premise
$\{2\}$	(2)	$A \wedge B$	Assumption (to be discharged by RAA)
$\{3\}$	(3)	$\neg A$	Assumption (to be discharged by $\vee$E)
$\{2\}$	(4)	A	$\wedge$E 2
$\{2, 3\}$	(5)	$\bot$	$\neg$E 3, 4
$\{6\}$	(6)	$\neg B$	Assumption (to be discharged by $\vee$E)
$\{2\}$	(7)	B	$\wedge$E 2
$\{2, 6\}$	(8)	$\bot$	$\neg$E 6, 7
$\{1, 2\}$	(9)	$\bot$	$\vee$E 1, 3-5, 6-8
$\{1\}$	(10)	$\neg(A \wedge B)$	RAA 2-9

$\neg(A \vee B) \vdash \neg A \wedge \neg B$

$\{1\}$	(1)	$\neg(A \vee B)$	Premise
$\{2\}$	(2)	A	Assumption (to be discharged by RAA)
$\{2\}$	(3)	$A \vee B$	$\vee$I 2
$\{1, 2\}$	(4)	$\bot$	$\neg$E 1, 3
$\{1\}$	(5)	$\neg A$	RAA 2-4
$\{6\}$	(6)	B	Assumption (to be discharged by RAA)
$\{6\}$	(7)	$A \vee B$	$\vee$I 6
$\{1, 6\}$	(8)	$\bot$	$\neg$E 1, 7
$\{1\}$	(9)	$\neg B$	RAA 6-8
$\{1\}$	(10)	$\neg A \wedge \neg B$	$\wedge$I 5, 9

$\neg A \wedge \neg B \vdash \neg(A \vee B)$

$\{1\}$	(1)	$\neg A \wedge \neg B$	Premise
$\{2\}$	(2)	$A \vee B$	Assumption (to be discharged by RAA)
$\{3\}$	(3)	A	Assumption (to be discharged by $\vee$E)
$\{1\}$	(4)	$\neg A$	$\wedge$E 2
$\{1,3\}$	(5)	$\bot$	$\neg$E 3, 4
$\{6\}$	(6)	B	Assumption (to be discharged by $\vee$E)
$\{1\}$	(7)	$\neg B$	$\wedge$E 2
$\{1,6\}$	(8)	$\bot$	$\neg E$ 6, 7
$\{1,2\}$	(9)	$\bot$	$\vee$E 2, 3-5, 6-8
$\{1\}$	(10)	$\neg(A \vee B)$	RAA 2-9

$\neg(A \to B) \vdash A \wedge \neg B$

$\{1\}$	(1)	$\neg(A \to B)$	Premise
$\{2\}$	(2)	$\neg(A \wedge \neg B)$	Assumption (to be discharged by RAA')
$\{3\}$	(3)	A	Assumption (to be discharged by $\to$I)
$\{4\}$	(4)	$\neg B$	Assumption (to be discharged by RAA')
$\{3,4\}$	(5)	$A \wedge \neg B$	$\wedge$I 3, 4
$\{2,3,4\}$	(6)	$\bot$	$\neg$E 2, 5
$\{2,3\}$	(7)	B	RAA' 4-6
$\{2\}$	(8)	$A \to B$	$\to$I 3-7
$\{1,2\}$	(9)	$\bot$	$\neg$E 1, 8
$\{1\}$	(10)	$A \wedge \neg B$	RAA' 2-9

$A \to B, \neg B \vdash \neg A$

$\{1\}$	(1)	$A \to B$	Premise
$\{2\}$	(2)	$\neg B$	Premise
$\{3\}$	(3)	A	Assumption (to be discharged by RAA)
$\{1,3\}$	(4)	B	$\to$E 1,3
$\{1,2,3\}$	(5)	$\bot$	$\neg$E 2, 4
$\{1,2\}$	(6)	$\neg A$	RAA 3-5

$A \to B \vdash \neg B \to \neg A$

$\{1\}$	(1)	$A \to B$	Premise
$\{2\}$	(2)	$\neg B$	Assumption (to be discharged by $\to$I)
$\{3\}$	(3)	A	Assumption (to be discharged by RAA)
$\{1,3\}$	(4)	B	$\to$E 1, 4
$\{1,2,3\}$	(5)	$\bot$	$\neg$E 3, 5
$\{1,2\}$	(6)	$\neg A$	RAA 4-6
$\{1\}$	(7)	$\neg B \to \neg A$	$\to$I 3-7

4 Sets, relations, and functions

Most of us use the notions of sets and relations daily, albeit we tend not to use these exact terms. If we go to the store and buy apples, we also purchase pip fruit, and in even more general terms, fruit. These are *sets* and *subsets* of fruit. We may later choose to offer an apple to a friend, a family member, or someone totally unrelated, except that it possibly would be a human; apples are also a fundamental ingredient in apple pie. The latter involves both sets and *relations*.

While we cannot grab the actual set of apples, we can pick any of the individual apples that are *members* of the set. A set itself is always abstract, while its members are sometimes tangible objects, like apples, friends, family members, or anyone unrelated. The set of all people in a concert hall are all live objects but do not have to be visible to us, yet we can speak about them as a set.

A part of a set is called a subset. The children of a family make up a subset of the family. So do all the grown-ups, even if there are no children—a subset does not have to be a so-called *strict subset*. The pie recipes with apples on the ingredient list comprise a subset of all pie recipes. The people in front of us in a concert hall are a subset of all people in the concert hall, and the people behind us are another subset of the same set. In this course, we skip discussing vagueness that otherwise arises in many practical instances, such as when trying to determine the border between what is in front of us and what is behind us.

Being a friend of someone is a *relation*, as is being on the same street as, or being a family to someone. Being on the ingredient list of some recipes is also a relation. The relation 'is a friend of' is a relation among the members of the set of humans, while the relation 'is on the ingredient list of' is a relation between the set of ingredients and the set of recipes. Relations can thus involve one or several sets.

Some relations are special in the way that they point only to one specific individual. The 'is the biological mother of' is such a relation, but the relation 'is the biological parent of' is not. In the first case, we can express the relation as a function from the set of children to the sets of parents. For any child, there will only be exactly one biological mother (whether we know who that person is or whether that person is alive is beside the point). That is analogous to always obtaining the same output of a mathematical function such as $f(x) = x^2$; we do not expect to end up with different results for the same x—different xs can yield the same result though, as would 2 and -2. If we consider the relation 'is the biological parent of,' however, we would end up with two individuals; therefore, that particular relation does not underlie a function.

In addition to referring to sets, subsets, relations, and functions, we need to be able to speak about different combinations of sets, such as the set of all individuals who are working or studying, the set of individuals who are doing both, or the set of individuals who are working but do not study. In addition, we sometimes would like to speak about the number of individuals in a set.

Sets, relations, and functions are concepts highly applicable in most fields, regardless of the degree of formalism. In this particular case, however, they will be used in the context of predicate logic, and therefore we need to look at the more abstract definitions of what we so far have been referring to from a rather practical point of view.

4.1 Sets

A set is defined by its members. If it has no members, then it is referred to as the *empty set*, denoted by the symbol $\varnothing$. We say that A is a set if and only if there is some individual that is a member of A, or A is the empty set. If a is a member of the set A, we can use the symbol $\in$ denoting 'is a member of,' and write it as $a \in A$. Using that symbol, which is fundamental to the theory of sets, and some logical notation from earlier chapters, we say that

$$A \text{ is a set} \leftrightarrow (\text{there is some } a \text{ such that } a \in A) \vee A = \varnothing.$$

We will use the symbol $\in$ a lot, so remember its meaning 'is a member of.' To say the opposite, that something is not the member of a set, we use $\notin$ and define it as

$$a \notin A \leftrightarrow \neg(a \in A).$$

If the precise members of a set are known to us, and they are not too many, we can denote a set by listing its members between curly brackets, such as $\{\text{apple pie}, \text{blue berry pie}, \text{shepherd's pie}, \text{magpie}\}$. The empty set would thus be written as $\{\,\}$.

Often, however, a member of a set is not known until seen, so to speak. For example, we can hardly know all birds in the world at one and the same time, but when we see a bird, we would, most of the time anyway, know that it belongs to the set of birds. In such cases, a set can be written as

$$\{x|P(x)\}$$

where $P(x)$ means that x fulfills some property P, such as being a bird. As soon as we see some individual a, we check whether $P(a)$ is true and if it is, we know that a belongs to the set. In this manner, the empty set can be defined as

$$\{x|x \neq x\},$$

namely the set of things that are not identical to themselves—since any individual is identical to itself, $x \neq$ is false for all individuals.

A set with only one member, e.g., $\{a\}$. is called a *singleton set* or a *unit set.*

We can use multiple properties to specify a member further. Let $S(x)$ be the property that x is spreckled, $B(x)$ that x is black, $C(x)$ that x is a crow, and $H(x)$ that x is a hen. Then the set of all spreckled hens and black crows can be written as

$$\{x|\,[S(x) \wedge H(x)] \vee [B(x) \wedge (C(x)]\}\,.[3]$$

Two sets are equal if they contain exactly the same members. Thus

$$A = B \leftrightarrow \text{for any individual } x, x \in A \leftrightarrow x \in B.$$

A subset is a set whose members are all members of another set—as such, a subset is not just a subset but a 'subset of' another set, denoted by $\subseteq$. If A is a subset of B then

$$\text{for any individual } x, A \subseteq B \leftrightarrow x \in A \rightarrow x \in B.$$

[3] Whether we use parentheses or square brackets are immaterial.

If B contains all the members of A plus some other members, then A is a *strict subset* of B, hence

$$\text{for any individual } x, A \subset B \leftrightarrow (x \in A \rightarrow x \in B) \wedge A \neq B.$$

Each individual can only show up once in each set. Thus $\{2, 2, 5\}$ is not a set since 2 occurs more than once, but $\{2, 3\}$ is. In addition, the order of the members of a set does not matter, $\{2, 3\} = \{3, 2\}$.

The empty set is special as it is a subset of all sets, even of itself. It is not, however, a member of all sets—yet, as you will see, the empty set is a member of some sets.

It may seem strange at first that the empty set, indeed, is a subset of all sets, but it is a consequence of our definition of a subset. If A is a subset of B, then any member of A is also a member of B, so if A is not a subset of B, then there has to be some member of A that is not a member of B. Consequently, if the empty set is not a subset of some other set C, then the empty set has to have at least one member that is not a member of C. But, since the empty set has no member at all, it cannot be the case that the empty set is not a subset of C; hence the empty set is the subset of C, as well as of every other set.

A partition of a set A is a collection of sets, $A_1, A_2, \ldots$ (essentially a set of sets), such that each member of A is a member of exactly one of $A_1, A_2, \ldots$. Hence, all of $A_1, A_2, \ldots$ will be disjoint, i.e., they will not share a single element. A very practical analogy is dividing a cake into pieces, where the resulting pieces would form a collection called a partition.

The powerset if a set A is a collection of all the possible subsets of A, and here the empty set is included as a member. For $B = \{1, 2\}$, the powerset of B, denoted $\mathcal{P}(B)$ is $\{\{\,\}, \{1\}, \{2\}, \{1, 2\}\}$. The cardinality of $\mathcal{P}(A)$ is 2^n, where n is the number of elements of A.

The cardinality of a set is the number of members it has. The cardinality of $A = \{\text{apple pie}, \text{blue berry pie}, \text{shepherd's pie}, \text{magpie}\}$, denoted $|A|$, is 4.

4.1.1 Operations on sets

Given a collection of sets, we can form new sets by applying certain operations on sets. Some of these operations are somewhat analogous to logical negation, conjunction, and disjunction.

Definition 4.1. The complement $\overline{A}$ of a set A, has as members all the individuals not in A, hence

$$\text{for any individual } x, x \in \overline{A} \leftrightarrow x \notin A, \text{or}$$

$$\overline{A} = \{x | x \notin A\}.$$

Definition 4.2. The union $A \cup B$ of two sets A and B consists of all elements that are either in A, in B, or in both. Using symbols, we have

$$\text{for any individual } x, x \in A \cup B \leftrightarrow x \in A \vee x \in B, \text{or}$$

$$A \cup B = \{x | x \in A \vee x \in B\}.$$

Definition 4.3. The intersection $A \cap B$ consists of the individuals that are members of both A and B,

$$\text{for any individual } x, x \in A \cap B \leftrightarrow x \in A \wedge x \in B, \text{or}$$

$$A \cap B = \{x | x \in A \wedge x \in B\}.$$

Definition 4.4. The difference between the sets A and B, written as $A \setminus B$ is defined by

$$\text{for any individual } x, x \in A \setminus B \leftrightarrow x \in A \wedge x \notin B, \text{or}$$

$$A \setminus B = \{x | x \in A \wedge x \notin B\}.$$

Definition 4.5. The cartesian product of two sets A and B is a set of all ordered pairs formed by letting the first member of each pair come from A and the second member of the pair from B. Using the symbol ×, we write

$$\text{for any individual pair } (x, y), (x, y) \in A \times B \leftrightarrow x \in A \wedge y \in B, \text{or}$$

$$A \times B = \{(x, y) | x \in A \wedge y \in B\}.$$

4.2 Relations

In addition to referring to the properties of individuals, it is useful to have the ability to define relations between individuals. Relations can, in turn, be used to define additional and more complex relations. Examples of relations include: 'greater than,' 'parent of,' 'ingredient in,' 'gave something to,' and so on. The first three examples are relations between two individuals, while the third is a relation involving three individuals, as in "Bob gave the keys to Anna"—in this context, the keys comprise an individual in addition to the obvious Anna and Bob.

Relations are, contrary to the everyday usage of the word, a set of ordered pairs—as we have seen, a relation may involve a greater number of individuals, but for this presentation, so-called *binary* relations will be the focus. The following is a relation

$$\{(\text{apples, apple pie}), (\text{potatoes, shepherd's pie})\},$$

namely the relation 'is an ingredient in,' and

$$\{(\text{magpie}, \text{potatoes}), (\text{magpie}, \text{shepherd's pie})\}$$

is the relation 'eats.' We could give these relations labels and define them as xIy if and only if 'x is an ingredient in y,' and xEy if and only if 'x eats y' respectively.

Given the set

$$\{(0, 1), (3, 2), (2, 0)\}$$

of ordered pairs, the relation 'greater than' over that set of ordered pairs is

$$\{(3, 2), (2, 0)\}.$$

We usually refer to a relation as a relation from one set, called the *domain* of the relation, to another set, called the *counterdomain* of the relation. The union of the domain and counterdomain are together the *field* of the relation.

Since relations are sets of ordered pairs, we can speak about arbitrary relations in terms of sets of ordered pairs that may have certain properties. A relation R is *symmetric* in a set A if and only if for any pair $(x, y) \in R$ there is a pair $(y, x) \in R$, with $R \subseteq A \times A$. The same relation is *asymmetric* if and only if for any $(x, y) \in R$, the pair $(y, x) \notin R$. The relation R is *transitive* if and only if, given $(x, y) \in R$ and $(y, z) \in R$, there is a pair $(x, z) \in R$. We say that a relation R is reflexive in A if and only if for any individual $x \in A$, there is a pair $(x, x) \in R$.

4.3 Functions

A function is nothing more than a relation that is restricted in the sense that each individual of the domain is related to exactly one individual of the counterdomain—when referring to functions, the counterdomain is often known as the *range* of the function. Unless something has gone awry, the relation from the set of passengers to the set of seats in an airplane is a function. It maps each individual to exactly one seat. Even if the plane is overbooked it is still a function, since no passenger will be assigned to more than one set. The relation from the set of shoppers in a store to the set of products is not a function since each shopper can be related, through the relation 'buys,' to more than one product.

Note that the reverse does not have to be true. A function can map different individuals in the domain to the same individual of the counterdomain. Take, for example, the function $f(x) = x^2$. The result of $f(2) = 4$ and $f(-2) = 4$ are the same.

Let f be a function from the set A to the set B. That can be written rather compactly as $f : A \to B$. It does not say anything about what the function does other than taking an individual in A as input and resulting in an individual in B; the sets A and B may be the same.

The set

$$\{(\text{A4}, \text{Audi}), (\text{Accord}, \text{Honda}), (\text{Charger}, \text{Dodge})\}$$

is a function with the domain

$$\{\text{A4}, \text{Accord}, \text{Charger}\}$$

and counterdomain

$$\{\text{Dodge}, \text{Honda}, \text{Audi}\}.$$

Considering the mapping between model names and makes at a more general level, we can assume it is a function as no two makes produce cars with the same model name—if, for some reason, a car with the same model name would be produced by two or more different brands, then the mapping would no longer be a function.

5 Predicate logic

The one thing predicate logic has in common with propositional logic is the ability to communicate truth or falsity. The differences between the two are otherwise rather substantial. Both when it comes to the level of expressiveness and the underlying constructs. In propositional logic, we express complete statements as declarative sentences that are either true or false. Each interpretation of the sentence letters, as true or false, yields a truth value of a formula. Logical consequence depends on the structure of propositions, and any inference will consist of some combination of propositions already among the premises. Predicate logic, on the other hand, is more flexible—it allows us to state things about specific individuals in the *domain of discourse*. The domain of discourse is the set of individuals (i.e., objects or elements) we are interested in reasoning about, and it may differ from time to time. It is always chosen. It is nothing that exists on its own in a vacuum. Furthermore, predicate logic equips us with *quantifiers*, which support a more intricate reasoning process. They allow us to generalize over the domain of discourse by referring to *all* or *some* of the individuals.

Let us look at an example where the domain of discourse consists of the fruits in some grocery store—the domain of discourse, as long as it is not empty, can be chosen arbitrarily to suit the purpose. Suppose that some bananas are unripe while others are ripe. In propositional and predicate logic, we could express the proposition that "Some bananas are unripe." However, a major difference between the languages is that predicate logic allows for the inference that "Not all bananas are ripe," thanks to quantifiers and predicates. Given the proposition "Some bananas are unripe" (and only that) in propositional logic, we can only restate "Some bananas are unripe." No other option is available to us except for inserting additional premises to support further reasoning. In predicate logic, on the other hand, the sentence "Some bananas are unripe" can be represented in much greater detail, which allows for further logical reasoning based on that sentence alone. Later we will look at how.

5.1 Predicates

A *predicate* is a property or relation. Continuing with the fruit domain, the level of ripeness is a property of bananas (and other fruits, of course), and 'is riper than' is a relation between bananas. Properties can also substitute for verbs and adjectives. Hence, 'is yellow' is a predicate, just as 'is ripening.'

Predicates have a so-called *arity* which denotes the number of individuals it takes as arguments. The 'is yellow' predicate is of arity one because it says something about one individual, while the predicate 'is riper than' is of arity two since it involves a comparison of two individuals. A predicate of arity 0 is the same as a proposition in propositional logic.

The predicate 'is yellow' is true of some individuals but not all. It will be true of some bananas but false of others. The proposition formed by replacing x in 'x is yellow' with some individual in the domain of discourse will be true or false depending on the specific individual replacing x. Hence, we can see a predicate as a propositional function that yields true or false depending on the arguments.

It may seem odd to refer to the objects of the domain of discourse as individuals, and it is not done by necessity. Note, however, that we should be able to reference each object in the domain of discourse separately, i.e., each object is unique, albeit not necessarily known. Therefore, the objects are referred to as individuals rather than objects or elements.

Once we have settled on a domain of discourse and a set of predicates, we need to keep track of them. The predicates are commonly represented by capital letters, and the individuals by the first small letters of the alphabet. The arity of a predicate contributes to its uniqueness. Consequently, the letter P could represent several predicates as long as they have different arities. The arity of a predicate is usually denoted by a superscript such as P^1, and thus P^2 would be distinct from P^3, P^4, and so on. A predicate of arity n is also referred to as an n-place predicate.

When a predicate is coupled with the necessary number of individuals from the domain of discourse, we essentially have a true or false proposition. If P is the predicate 'is yellow' and some individual denoted by a is yellow, then P is true of a, or that $P(a)$ is true—to save space, one can skip the parentheses and simply write Pa.

In summary, with a set of predicates and labels denoting the various individuals in the domain of discourse, we can state things that are either true or false about any particular individual. The domain of discourse can be stated explicitly or be determined from the context. It is not necessary to assign labels to each individual in the domain from the start—only the individuals that, for some reason, warrant a specific label are given one. Sometimes we will just assign a label to some arbitrary individual under the assumption that that individual possesses certain properties or stands in certain relations to other individuals.

5.2 Quantifiers

The quantifiers support generalizing about the individuals in the domain of discourse. Returning to the fruit example, rather than enumerating each individual and stating that it is true of that individual that it is a fruit, we can, with the help of the quantifier 'all,' state that "It is true for each individual (in the domain of discourse), that it is a fruit," or "It is true for all x, that x is a fruit. Taking F as the predicate 'is a fruit,' we can write "For all x, $F(x)$." There is a special symbol for 'all,' namely $\forall$. Hence, in predicate logic, we would write the above as

$$\forall x F(x).$$

.

The other quantifier available to us is that of 'some,' which we can use to express that 'some fruits are ripe,' meaning that *at least one* fruit is ripe. Analogous to the above, we start with "It is true for some individual, that it is a fruit and that it is ripe." Letting F have the same meaning as above and R to mean 'is ripe,' we have "There exists some x, such that $F(x)$ and $R(x)$." The special symbol for 'some' or 'there exists' is $\exists$. Using that symbol along with the logical connective for 'and' yields

$$\exists x \left(F(x) \wedge R(x) \right).$$

The general names of the quantifiers are the *universal quantifier* and *existential quantifier*, for $\forall$ and $\exists$, respectively. The universal quantifier accounts for every individual in the universe, i.e., every individual in the domain of discourse, while the existential quantifier presumes the existence of at least one individual that satisfies the formula in question. What that means will be clarified below.

With the help of these quantifiers and some formula A, we can express that:

- A is true for all individuals.
- A is true for some individuals.
- A is true for no individuals.
- A is false for some individuals.

5.3 Syntax

The language of predicate logic consists of

- Predicate-letters (uppercase letters)
- Name-letters that label unique (but possibly arbitrary) individuals as needed (usually $a, b, c, \dots$)
- Function-letters (usually $f, g, h, \dots$)
- Variables as placeholders for names (usually $x, y, z, \dots$) quantifiers
- Logical connectives ($\neg, \wedge, \vee, \rightarrow, \leftrightarrow$)
- Quantifiers ($\forall$ and $\exists$)

Parentheses are not part of the language but are used to disambiguate between possible combinations, such as between $A \wedge (B \rightarrow C)$ and $(A \wedge B) \rightarrow C$. The type of each letter in a formula should be clear either from the context or explicit statements.

Any predicate with arity n, along with n names or variables, is a formula.[4] If A is a formula, then its negation $\neg A$ is a formula. Two formulas, A and B, connected by any of the logical connectives constitute a formula. If A is a formula and x is a variable, then $\forall x A$ and $\exists x A$ are formulas.

5.4 Free and Bound Variables

A quantifier *binds* the variable that immediately follows it. In $\forall x A$, the universal quantifier binds the variable x, and in $\exists x \forall y A$, the existential quantifier binds x, and the universal quantifier binds y. Here, A can be any formula.

Except for negation, the quantifiers have precedence over all the other connectives. Hence, $\forall x P(x) \wedge Q(x)$ means $(\forall x P(x)) \wedge Q(x)$. In that formula, the variable x in $P(x)$ is bound to the quantifier because it is within the *scope* of the quantifier, while the variable x in $Q(x)$ is not. To extend the scope of the quantifier, we have to use parentheses like so:

$$\forall x \left(P(x) \wedge Q(x)\right).$$

If we were to have P as a two-place predicate instead with, e.g., an existential quantifier binding only the second argument to P, we would write it as

$$\forall x \left(\exists y P(x, y) \wedge Q(x)\right).$$

[4] The name *first-order* predicate logic is due to the fact that the predicates only range over individual terms.

Here, the scope of the universal quantifier still extends over the formula $\exists y P(x, y) \wedge Q(x)$ (enclosing parentheses excluded), while the scope of the existential quantifier only includes $P(x, y)$.

As a rule, if a quantifier together with the variable that it binds is followed immediately by

- a predicate, then the scope of the quantifier covers only that predicate and its terms
- a negation, then the scope of the quantifier covers the negation and the formula immediately following the negation
- a left parenthesis, then the scope of the quantifier extends until, and including, the matching right parenthesis
- another quantifier, then the scope of the quantifier is the same as the second quantifier, in addition to the second quantifier itself.

If a variable is within the scope of a quantifier and bound to that quantifier, then that variable is said to be *bound*. A variable that is not bound is *free*. In the following, the one instance of x is bound while all instances of y are free:

$$\exists x P(x, y) \to Q(y).$$

A formula with only bound variables is *closed*, otherwise it is *open*.

5.5 Translation to predicate logic

When translating a sentence (or passage) in natural language to predicate logic, one needs to map the various parts of the sentence to corresponding entities in predicate logic. The entities can be of the following three types: (1) terms, (2) predicates, and (3) quantifiers. Let us have another look at them in the context of translating sentences into English.

Terms are either *constants*, *variables* och *functions*. *Constants* names or describes given objects and are usually denoted by small letters from the beginning of the alphabet, e.g.,

$a =$ 'Buddy Rich,' or

$b =$ 'The white rabbit in Alice's Adventures in Wonderland.'

Variables, on the other hand, name or describe some object once replaced with a definite description or name. They are usually denoted by small letters from the second half of the alphabet. Given the sentence "All my plants are green," we could use a variable such as x and write it as "For any x, if x is a plant, then x is green." The sentence will be true whenever we replace x with a description or the name of an actual plant.

Predicates can have zero or more *arguments* depending on their meaning, e.g., P has zero arguments, $Q(x)$ has one argument, $S(x, y)$ has two arguments, etc. They are commonly written using uppercase letters to distinguish them from constants and variables. Using parentheses to enclose the arguments is optional but can sometimes enhance readability. Hence, the previous examples could be written Qx, and Sxy without any changes in meaning. *Predicates* with zero arguments are used to denote statements, just like propositional letters in sentential logic. *Predicates* with one argument, also called one-place predicates, are useful for indicating a property or action of an object, as in "Roselda is red" and "Tom picks flowers." Setting

$R =$ 'is red,'

$P =$ 'picks flowers,'

$r =$ 'Roselda,' and

$t =$ 'Tom,'

the two sentences could be written as Rr and Pt respectively. Two place predicates typically express some form of relation, as in "Tom picks flowers for Roselda." Here we could set

$Qxy =$ 'x picks flowers for y,'

and thus express the sentences as Qtr. The order of the arguments is important; x was replaced with t (denoting Tom), and y was replaced by r (denoting Roselda). Changing the order of t and r, as in Qrt would result in the statement "Roselda picks flowers for Tom." Relations such as "Tom drives faster than Bob but slower than Roselda" could, in a similar manner, be expressed using a three-place predicate $Sxyz$, defined as

$Sxyz =$ 'y drives faster than x but slower than z.'[5] With $b =$ 'Bob,' the sentence "Tom drives faster than Bob but slower than Roselda" would be written as $Sbtr$. Note that if the relation $x < y$ would concern speed (i.e., x drives slower than y) then the previous predicate $Sxyz$ expresses the relation $x < y < z$. Hence, the order chosen for the arguments seems rather natural.

The sentence "x is cold" (here, 'cold' refer to the everyday notion of temperature) is true depending on the replacement for x. If x is replaced with something we just grabbed from a fully functional fridge, then we would consider the sentence to be true. On the other hand, if we substituted x with freshly baked (and warm) bread, then the sentence would be false. Hence the truth or falsity of the sentence depends only on the object whose name or description we substitute for x, given that we agree on what it means for something to be cold. In predicate logic, "x is cold" would be called an open formula rather than a sentence because it contains a *free variable.* The simplest way of turning it into a closed formula or a sentence in predicate logic would be to replace the free variable x with a constant, such as $a =$ 'the Arctic.' Other ways of doing it are described below.

Quantifiers are used to make the truth value of a formula in predicate logic depend on the properties and the relations denoted by the various predicates and the domain of objects whose descriptions or names can be substituted for constants or variables.

The *universal quantifier*, $\forall$, provides a single substitute for expressions like "Everything...," "For every x...," "All x...," etc. Hence, the truth or falsity of a sentence such as "For all x it holds that x is black" do not depend on the object replacing the variable x in a single instance, but rather on the meaning of 'is black,' the *domain* of objects under consideration, and the fact that the statement is a universal generalization. For example, if we explicitly referred to *a set of black marbles* (i.e., the domain of objects would be a set of black marbles), then the sentence would be true, but if we referred to all things in the universe, then it would obviously be false since there are objects in the universe that has colors other than black.

[5] Note that the variables need to be included in the definition since the reader otherwise would be left with no clue as to whether $Sxyz$ meant "y drives faster than x but slower than z" or "x drives faster than y but slower than z," etc.

A sentence that certainly would be true in case the domain included *all the things in the universe is* "There exists an x such that x is black." However, the same sentence would be false if the domain was taken to be *all the daisies on a field* (none of which were black). To symbolize subexpressions such as "There exists...," "Some...," "At least one...," etc. we use the *existential quantifier*, $\exists$.

Using the quantifier symbols we would write the previous examples as "$\forall x$, x is black" and "$\exists x$, x is black." Writing a variable immediately after a quantifier is known as *binding* the variable to the quantifier in question. If all variables of a formula are bound to quantifiers, then that formula is *closed* and counts as a *sentence* in predicate logic.

Each quantifier has a scope, meaning how much of a formula it covers in terms of binding a given variable. For example, in $\forall xPx \wedge \exists yQxy$ the scope of the universal quantifier only includes Px. Because the second instance of x is not bound to any quantifier, it is a free variable. Note that in the above formula, the x in Px is not the same x as the x in Qxy. Consequently, the formula could be written as $\forall xPx \wedge \exists yQzy$, replacing the second x with the variable z. However, by adding parenthesis to the initial formula, as in $\forall x[Px \wedge \exists yQxy]$, the scope now includes both xs. That would make the formula *closed* since it no longer contains any free variables.

A dictionary is a list of definitions of the predicates and constants used in a formula. We have already made use of dictionaries earlier in the text. Again, the order of the arguments is imperative. Apparently, the following definitions are not equal:

(1) $G(x, y)$: 'x is greater than y,' and
(2) $G(y, x)$: 'x is greater than y.'

However, the actual symbols do not matter as long as the positions are consistent, in this case, meaning that the first variable is greater than the second variable. Hence,

(3), $G(y, x)$: 'y is greater than x,'
is equal to (1).

Example 5.1. We should now have enough terminology for translating sentences from natural language into predicate logic, starting with a rather simple sentence for which the existential quantifier will suffice:

> Some red foxes run fast and are constantly on the lookout.

It should be said that the simplest way of translating the sentence would be to use a zero-place predicate; for example,
A: "Some red foxes run fast and are constantly on the lookout,"
but then we would miss all of the possible benefits we might get from using predicate logic.[6] Furthermore, the specific granularity, i.e., the amount of detail, of a translation would depend on the context in which it will be used. In trying to reap the benefits of translating the sentence into predicate logic rather than into sentential logic, we could start by addressing the terms and the predicates. For starters, we could choose as a term the group of foxes which has a size of at least one fox. Since we aren't referring to any particular group, it would be incorrect to use a constant for the group. By defining the predicate

[6] Note that the literature isn't consistent when it comes to zero-place predicates, some allow only for predicates with at least one argument. However, one could argue that any predicate to which we apply constants as arguments would equal a unique zero-place predicate.

$R_f x$: 'x is a red fox,'
and by using the existential quantifier, we can begin by stating that there indeed are some foxes that are red, like this: $\exists x R_f x$. The subscript $_f$ is there to distinguish R_f from R, which we will use later. Let us then add the predicates
Ax: 'x runs fast,' and
Bx: 'x is constantly on the lookout.'
Given these predicates, the sentence can now be written as

$$\exists x(R_f x \land Ax \land Bx).$$

The enclosing parentheses extend the scope of the quantifier to cover the whole formula, and since there are no free variables, we have a sentence in predicate logic.

In order to increase the granularity a bit, we could exchange R_f with the predicates
Rx: 'x is red,' and
Fx: 'x is a fox,'
with which we could write the sentence as

$$\exists x(Fx \land Rx \land Ax \land Bx).$$

If we translate this back into natural language, we will get "For some x, x is a fox, x is red, x runs fast, and x is constantly on the lookout," asserting that there is at least one such fox, which is perfectly in line with the original sentence.

We could also pay attention to time since the foxes in question are on the lookout constantly, i.e., at every instant of time. Hence, we have to include yet another term, namely an instant of time (well, in fact, several instances of time). To do this, we exchange the predicate B for
Tx: 'x is an instant of time,' and
Cxy: 'x is on the lookout at time y.'
Apparently, we will also have to use a second variable in this case. Now, since "constantly" refers to all instances of time, we must use the *universal quantifier* for binding the variable that represents instances of time. The order of the quantifiers is as important as the order of the arguments as will be shown soon.

Using these newly created predicates, we can further increase the granularity and write the sentence as

$$\exists x \forall y \left(Fx \land Rx \land Ax \land (Ty \to Cxy)\right).$$

Let us substitute all occurrences of x with the constant a, assuming that *a denotes a particular individual* (in this case a fox) that satisfies[7] the predicates F, R, A and C (for the original English sentence to be true, there has to be at least one such an individual). When substituting all occurrences of x with a, we also have to remove the outermost quantifier since quantifiers only bind variables, not constants. This results in

$$\forall y \left(Fa \land Ra \land Aa \land (Ty \to Cay)\right).$$

For any object that we substitute for y, either that object is not an instance of time (i.e., does not satisfy T), or it is an instance of time at which the fox is on the lookout.

[7] If we pass an object as an argument to a predicate that makes the statement true, then that object is said to *satisfy* the predicate.

Should we have changed the order of the quantifiers, as in

$$\forall y \exists x \left(Fx \wedge Rx \wedge Ax \wedge (Ty \rightarrow Cxy) \right),$$

then we would first have replaced the variable y with an arbitrary instance of time, say t_i, yielding

$$\exists x \left(Fx \wedge Rx \wedge Ax \wedge (Tt_i \rightarrow Cxt) \right).$$

That would tell us that for any instance of time, there is some fox that is on the lookout, which is quite different from a fox being on the lookout at all times.

It might be worth mentioning that with no limits of the domain, the sentence $\exists x \forall y \left(Fx \wedge Rx \wedge Ax \wedge (Ty \rightarrow Cxy) \right)$ asserts that some fox has a lifespan as long as the universe. This could easily be remedied by letting the instances of time included in the domain be limited to times relevant for the particular fox or explicitly stating it by adding yet another predicate that is satisfied only by instances of time that indeed are relevant to the sentence.

In addition to the previous formula, one could consider other ways of writing the same statement in predicate logic, for example, by moving the universal quantifier inwards as in $\exists x(Fx \wedge Rx \wedge Ax \wedge \forall y(Ty \rightarrow Cxy))$.

Another approach to translating a sentence is to reword the sentence to make it read more like predicate logic at the start. The original sentence "Some red foxes run fast and are constantly on the lookout" could thus be written as "For some x, x is a red fox, and x is constantly on the lookout." To take it further, we obtain "For some x, x is a fox and x is red, and x is constantly on the lookout." If we want to consider the time as well, we can go, "For some x, x is a fox, and x is red, and for all y, if y is an instance of time, then x will be on the lookout at time y." Once this stage is reached, it is easier to replace various parts of a sentence with corresponding predicates.

Example 5.2. Let us look at a translation of a longer sentence taken from a past exam.

> Even though the examiner hopes all students will satisfy the requirements for grade E or better, somebody will receive a lower grade and be disappointed.

For starters, the sentence can be written as "For some x, x is an examiner, and for all y, if y is a student, then x hopes that y will satisfy the requirements for grade E or better, and for some z, z is a student, and z will receive a lower grade than E and be disappointed." So far, we have included variables for the terms *examiner* and the *students*. Let

Ex: 'x is an examiner,'
Sx: 'x is a student,'
Hxy: 'x hopes that y will satisfy the requirements for grade E or better,'
Lx: 'x will receive a lower grade than E,' and
Dx: 'x will be disappointed.'

With these predicates, the sentence can now be written as

$$\exists x[Ex \wedge \forall y(Sy \rightarrow Hxy)] \wedge \exists x[Sx \wedge Lx \wedge Dx].$$

Here, square brackets are used in addition to regular parentheses to make the sentence easier to read.

As usual, there is room for some general remarks about the translation. So far, we have not considered how the students are related to the examiner. If we don't specify the domain, then it would seem like the examiner hopes that all students in the universe would satisfy the requirements for grade E or better. Maybe there is such an examiner, but the students referred to in the sentence are likely to be the students of that particular examiner. By turning S into a two-place predicate, as

Sxy: 'x is a student of y,'

we can eliminate this ambiguity. But then we would also have to extend the scope of the existential quantifier since we are referring to the same examiner throughout, as in

$$\exists x[Ex \land \forall y(Syx \to Hxy) \land \exists y(Syx \land Ly \land Dy)].$$

Note that we also had to change variables in the latter part since the x referring to the examiner had to be used there as well.

We can also treat the requirements and individual terms by turning H into a three-place predicate

$Hxyz$: 'x hopes that y will satisfy z'

and by adding the predicate

Rx: 'x is a requirement for grade E or better.'

This way we would get

$$\exists x[Ex \land \forall y(Syx \to \forall z(Rz \to Hxyz)) \land \exists y(Syx \land Ly \land Dy)].$$

In the same manner, we can refer to the grades, which are E or better, separately. Extending R to be

Rxy: 'x is a requirement for y'

and using

Gx: 'x is a grade, E or better'

we obtain

$$\exists x[Ex \land \forall y(Syx \to \exists s\forall z(Gs \land Rzs \to Hxyz)) \land \exists y(Syx \land Ly \land Dy)].$$

Now is the time to incorporate a constant into our sentence, namely a constant for grade E. Let us define it as

e: 'grade E.'

Furthermore, by including the predicate

Bxy: 'x is at least as good as y'

and by changing Gx to simply stand for 'x is a grade' we can write $Gx \land Bxe$ to say that x is a grade that is at least as good as E. This can be used in the latter part of the sentence to represent the grade lower than E as well, given that we also provide another version of L, namely Lxy: 'x will receive y.' Now if we update the complete sentence with this, we get

$$\exists x[Ex\land\forall y(Syx \to \exists s\forall z(Gs\land Bse\land Rzs \to Hxyz))\land\exists y\exists z(Syx\land Gz\land\neg Bze\land Lyz\land Dy)].$$

Here is the complete dictionary used in the last version of the sentence:

Ex: 'x is an examiner,'

Sxy: 'x is a student of y,'

Gx: 'x is a grade,'

Bxy: 'x is at least as good as y,'

Rxy: 'x is a requirement for y,'
$Hxyz$: 'x hopes that y will satisfy z,'
Lxy: 'x will receive y,'
Dx: 'x will be disappointed.'

There is not much more to do to increase the granularity of the translation further. Of course, 'at least as good as' could be split into two parts, 'better than' and 'equal to,' but other than that, I see no obvious opportunities.

Note that our translation was based on the fact that it is the requirements for some grade, E or better, that the examiner hopes the students will fulfill, not the grade itself. You should also pay special attention to the order and the scope of the quantifiers. We are referring to the examiner by x throughout the whole sentence, but z is used to represent the requirements in the first part, while in the second part, it denotes the grade. Should we switch the order of the quantifiers $\exists s \forall z$, then we would have ended up saying that:

> "For each grade at least as good as E there is a requirement which the examiner hopes the student will fulfill and that that would be the case for all grades at least as good as E."

However, the original sentence implies that *the examiner hopes that each student will fulfill all the requirements for some grade, as long as that grade is at least as good as E.*

Depending on the complexity of the original sentence, there are usually multiple ways of translating a sentence written in natural language into predicate logic. This is partly due to the level of granularity one might need for a particular application but also because of the different ways in which quantifiers and predicates can be combined. In general, for any sentence in natural language, you can choose multiple ways of representing it in predicate logic, and one could be more suitable than another, depending on the context in which the translation will be used.

5.6 Satisfiability, Truth, and Validity

The truth value of a prime formula $P(a)$ depends on the actual predicate chosen for P and the individual chosen for a, the latter being affected by the chosen domain of discourse. Take the planets in our solar system as the domain of discourse and let P stand for 'is inhabited.' Then $P(a)$ is true whenever the earth is substituted for a, yielding 'the earth is inhabited,' but false when any other planet is substituted for a, as in 'Saturn is inhabited.'

A formula $\exists x P(x)$ is true as long as P is true of at least one individual, and $\forall x P(x)$ is true as long as P is true of each and every individual in the domain of discourse. Consequently, $\exists x P(x)$ would be true when the domain of discourse and the meaning of P remain as above. However, $\forall x P(x)$ would not be true. The truth of a formula in predicate logic depends on the interpretation of the predicate letters, name letters, and function letters, as well as the domain of discourse.

Definition 5.1. An *interpretation* consists of four parts

1. A specific *domain of discourse.*
   This is a set of individuals about which one would like to reason.

2. An *assignment of the predicate letters to relations* over the domain of discourse with corresponding arities.
   The meaning of each predicate is expressed through a relation over the domain of discourse to which the corresponding predicate letter is assigned. For example, suppose the domain of discourse consists of all humans. In that case, the meaning of the predicate 'is the parent of' can be expressed by the relation consisting of all ordered pairs such that the first member of the pair is a parent, and the second member is a child of that parent. Such relations convey the meaning of the predicates.[8] A predicate such as 'is yellow' is assigned to the subset of the domain of discourse containing all things yellow.

3. An *assignment of the function letters to functions* from the set of tuples with the corresponding arity to the domain of discourse. A function that takes two arguments is a function from the set of ordered pairs to the elements of the domain of discourse. A function taking one argument is a function from the domain of discourse to the domain of discourse.

4. An assignment of each name letter to a specific individual in the domain of discourse.

The notion of truth in predicate logic depends on satisfiability which, in turn, depends on the chosen interpretation. After all, whether a formula $P(x)$ can be satisfied depends on the meaning of the predicate letter P and the domain of discourse. For example, if P stands for 'is a Normandy cheese' but the domain of discourse is the set of natural numbers, then the formula $P(x)$ can hardly be satisfied—we can not replace x with any of the individuals a_i in the domain of discourse such that $P(a_i)$ would be true of. However, if the domain of discourse consists of all the world's cheeses, then P would be true of at least some individuals and thus satisfiable in that domain of discourse.

What we just referred to was two different interpretations, each containing a domain of discourse and a one-place predicate letter—there was no function or constant letter involved. The first interpretation had the natural numbers as the domain of discourse, with the interpretation of P represented by the empty set since no natural number satisfies the predicate of being a Normandy cheese. The second interpretation had all the world's cheeses as the domain of discourse, and the predicate letter P assigned to the subset of the domain of discourse containing the Normandy cheeses.

Definition 5.2. An *open formula* is a formula that contains free variables.

Definition 5.3. A *closed formula* is a formula where any variable is bound to a quantifier.

Since the truth of an open formula depends on the individuals substituted for its variables, we refer to open formulas as *satisfiable* (or not) instead of true or false. If an open formula is true for at least one substitution of its variables, then it is satisfiable; otherwise, it is not satisfiable. A closed formula, however, that contains no free variables is simply true or false.

[8] With the predicate 'is a parent of,' we have a building block sufficient to define a grandparent, since a grandparent is the parent a of a child who in turn is the parent of the child who is the grandchild of a.

Definition 5.4. A formula A is *satisfied* in an interpretation if and only if there is at least one way of substituting the individuals in the domain of discourse for the variables in A that makes the formula true.

A prime formula $P(x_1, \dots, x_n)$ is satisfied by an n-tuple of elements $(a_1, \dots, a_n)$ if and only if $(a_1, \dots, a_n)$ is in the relation on the domain of discourse corresponding to P. The semantics of the logical connectives are the same as in propositional logic. A formula $\neg P$ is satisfied if and only if P is not satisfied. A formula $P \wedge Q$ is satisfied if P and Q are satisfied by the same substitution of variables. The conjunction $P \vee Q$ is satisfied if at least one of P and Q is satisfied by the same substitution of variables. A conditional $P \rightarrow Q$ is satisfied if P is not satisfied or if Q is satisfied by the same substitution of variables. A biconditional $A \leftrightarrow B$ is satisfied when B is satisfied whenever A is satisfied and vice versa, by the same substitution of variables. A formula $\forall x_1 \dots \forall x_n P(x_1, \dots, x_n)$ is satisfied if and only if any choice of n-tuple of individuals in the domain of discourse is in the relation assigned to P. A formula $\exists x_1 \dots \exists x_n P(x_1, \dots, x_n)$ is satisfied if and only if at least one n-tuple of individuals in the domain of discourse is in the relation assigned to P.

Definition 5.5. A formula A is *true for an interpretation* if and only if any substitution of individuals—in the corresponding domain of discourse—for the variables in A satisfies A.

Definition 5.6. A formula A is false in an interpretation if and only if no substitution of individuals—in the corresponding domain of discourse—for the variables in A satisfies A.

Definition 5.7. An interpretation is a *model* for a set of formulas $\mathcal{A}$ if and only if every formula in $\mathcal{A}$ is true for the interpretation.

Definition 5.8. A formula A is logically valid if and only if it is true, regardless of the interpretation.

The formula $\forall x \, (P(x) \vee \neg P(x))$ is one such valid formula. Regardless of the meaning of the predicate P and the chosen domain of discourse, any individual either satisfies or does not satisfy P.

Definition 5.9. A formula is *satisfiable* if and only if there is at least one interpretation in which the formula is true.

Any predicate logic formula without free variables is a *sentence*. Only sentences can be true or false. Formulas with free variables can possibly be satisfied, as we have seen, by substituting particular individuals in a domain of discourse for these variables, but they are not true or false in themselves. Contrary to propositional logic, there is no simple method for evaluating the truth value of a sentence in predicate logic. To determine whether a sentence is true or false, we need to consider all the possible variations of assigning individuals to the variables—that is, of course, impossible when the domain of discourse is infinite. But, even in cases where the domain of discourse is finite, the number of variations can be very large.

Example 5.3. Assume an interpretation with the domain of discourse $D = \{1, 2, 5, 8, 9, 11, 14\}$, and the predicates 'is an even number'

$$E = \{2, 8, 14\}$$

and 'have an absolute difference divisible by six'

$$F = \{(1,1),(2,2),(2,8),(2,14),(5,5),(5,11),(8,2),(8,8),\\ (8,14),(9,9),(11,5),(11,11),(14,2),(14,8),(14,14)\}.$$

The formula $\forall x \forall y\,(E(x) \wedge E(y) \to F(x,y))$ is true in that interpretation since any assignment of individuals to the variables x and y satisfy the formula. For example, setting $x = 5$ and $y = 11$ makes the antecedent $E(5) \wedge E(11)$ false, and thereby the conditional true. As another example, set $x = 2$ and $y = 8$, then the antecedent $E(2) \wedge E(8)$ is true but $F(2,8)$ is also true. Hence the whole conditional is true. Checking the rest of the possible assignments yield similar results.

Example 5.4. Assume an interpretation with the domain of discourse consisting of all islands on the planet, the predicate 'has a name starting with the letter *M*,' denoted by P, and the constant m denoting *Madagascar*. The formula $\forall x P(x)$ is not true in the interpretation since P is true only of some islands. However, the formula $\exists x P(x)$ is true in the interpretation—since P has at least one member—and so is $P(m)$. The formula $P(x)$ is satisfiable.

Example 5.5. The formula $\forall x \forall y\,(P(x,y) \to \forall z(P(z,x) \to P(z, f(x,y))))$ is not valid, but it is true under certain interpretations.

1. Let the domain of discourse be the natural numbers, let $P(x,y)$ be true of any pair (a,b) where $a > b$, and let the function $f(x,y)$ be the difference between x and y—that is the interpretation. To see that the formula is true for that interpretation, we consider any two natural numbers that satisfy $P(x,y)$, say 2 and 1. Assigning these numbers to the variables x and y yields $P(2,1) \to \forall z(P(z,2) \to P(z, f(2,1)))$. Since $f(2,1) = 1$ we have $P(2,1) \to \forall z(P(z,2) \to P(z,1))$. At this point we can substitute $\top$ for $P(2,1)$ as P is true of the pair $(2,1)$ to get $\top \to \forall z(P(z,2) \to P(z,1))$. Because the antecedent is true, the consequent $\forall z(P(z,2) \to P(z,1))$ must be true as well for the whole formula to be true. The domain of discourse is infinite, and so are the possible assignments of individuals to the variable z—we cannot check them all. However, any natural number greater than 2 is also greater than 1. Hence, there is no assignment to z that satisfies $P(z,2)$ but not $P(z,1)$. That would be true for any natural numbers (a,b) we could have chosen instead of $(2,1)$, as long as they satisfy P. Therefore the original formula is true in the given interpretation.

2. Consider the following interpretation: Let the domain of discourse be $\{a,b,c,d,e\}$, let $P = \{(a,b),(c,d),(c,e)\}$, and let $f(a,b) = d$, $f(c,d) = e$, and $f(c,e) = d$. There are three ways of assigning individuals to the variables of the antecedent. Let us consider each case:

   (a) Substitute a for x and b for y to get

   $$\forall x \forall y(P(a,b) \to \forall z(P(z,a) \to P(z,d))).$$

   Since the antecedent $P(a,b)$ is true it remains to check whether the consequent $\forall z(P(z,a) \to P(z,d)))$ is true in this interpretation as well. No pair with a in the second position satisfies P. Hence, no assignment of z will make $P(z,a)$ true. Therefore, $\forall z(P(z,a) \to P(z,d)))$ is true under this interpretation, and so is the original formula when we substitute a for x, and b for y.

(b) Substitute c for x and d for y to get

$$P(c,d) \to \forall z(P(z,c) \to P(z,e)).$$

Since (c,d) satisfies P, which makes the antecedent true, we need to check whether $\forall z(P(z,c) \to P(z,e))$ is true in the current interpretation. It turns out that no assignment of an individual to z satisfies $P(z,c)$—the antecedent will be false for any assignment to z. Consequently, $\forall z(P(z,c) \to P(z,e))$ is true in the given interpretation, and hence the original formula is true when c is substituted for x and d for y.

(c) Substitute c for x and e for y to get

$$P(c,e) \to \forall z(P(z,c) \to P(z,d)).$$

Since $P(c,e)$ is true in this interpretation we need to check whether $\forall z(P(z,c) \to P(z,d))$ is similarly true. The antecedent $P(z,c)$ is false in this interpretation regardless of the individual chosen for z. Therefore, $\forall z(P(z,c) \to P(z,d))$ is true for any assignment of individuals to z, and so is the original formula when c is substituted for x and e is substituted for y.

Having checked that all assignments of x and y that make the antecedent $P(x,y)$ true also make the consequent $\forall z(P(z,x) \to P(z,f(x,y)))$ true, we can conclude that the formula

$$\forall x \forall y(P(x,y) \to \forall z(P(z,x) \to P(z,f(x,y))))$$

is true in the given interpretation.

3. To show that the formula $\forall x \forall y\,(P(x,y) \to \forall z(P(z,x) \to P(z,f(x,y))))$ is not valid, assume an interpretation with the domain of discourse $\{-1,0,1\}$, and with $P = \{(-1,-1),(-1,0),(-1,1),(0,0),(0,1),(1,1)\}$ and $f(x,y) = x - 2y$. Substituting 1 for x and y each yields

$$P(1,1) \to \forall z(P(z,1) \to P(z,-1)).$$

The antecedent $P(1,1)$ is true. Let us see if there is some assignment of z that makes $P(z,1) \to P(z,-1)$ false. There is at least one such an assignment, for example $z = 1$, which results in $P(1,1) \to P(1,-1)$. Here the antecedent is true since P is true of $(1,1)$, while the consequent is false since P is not true of $(1,-1)$. That falsifies the original formula in this interpretation—therefore that formula is not valid (i.e., it is not true in every interpretation).

5.7 Equivalence

In predicate logic, a sentence (i.e., a formula without free variables) conveys something about the domain of discourse rather than specific individuals. Suppose a formula is true for a particular interpretation. In that case, it says something about the domain of discourse of that interpretation that is either true or false—saying something about the whole domain of discourse is clearly different from stating something about a specific individual. Therefore, logical equivalence in predicate logic involves formulas with all variables bound to quantifiers.

Definition 5.10. Two predicate logic formulas are truth value equivalent if and only if they are true in the exact same interpretations.

Later on, we will want to alter formulas into structurally different but logically equivalent formulas. The same rules as for propositional logic apply, but additional rules are needed to deal with quantifiers.

The universal quantifier distributes over $\wedge$ while the existential quantifier distributes over $\vee$. For any formulas A and B, we thus have the following equivalences

$$\forall x(A \wedge B) \Leftrightarrow \forall x A \wedge \forall x B,$$

$$\exists x(A \vee B) \Leftrightarrow \exists x A \vee \exists x B.$$

To see that the $\forall$-quantifier distributes over $\wedge$, suppose we have a domain of discourse of individuals that all share two properties P and Q, e.g., price and some level of quality. If every individual has both a price and a level of quality, then every individual has a price, and every individual has a quality level, and vice versa. To see that the $\exists$-quantifier distributes over $\vee$, assume a domain of discourse involving all fruits. Some are sweet, and some are sour (and some may be both). In that case, we are correct to say that there exists some individual that is sweet or sour, and also there exists some individual that is sweet, or there exists some individual that is sour (or both, of course).

Another fundamental equivalence results from expressing the universal quantifier in terms of the existential quantifier. To say that everything is sweet is the same as saying that there exists nothing that is not sweet. Similarly, we can express the existential quantifier in terms of the universal quantifier. There exists at least one sweet individual if and only if not all individuals are not sweet. Hence

$$\forall x P(x) \Leftrightarrow \neg\exists x \neg P(x), \text{ and}$$

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

In cases where we want to move a quantifier in front of a conjunction or disjunction—for reasons that will become obvious—the following rules apply.

$$A \wedge \forall x B \Leftrightarrow \forall x(A \wedge B), \quad A \vee \forall x B \Leftrightarrow \forall x(A \vee B)$$

$$A \wedge \exists x B \Leftrightarrow \exists x(A \wedge B), \quad A \vee \exists x B \Leftrightarrow \exists x(A \vee B)$$

Note that the variable x must not be free in A in any of these cases.[9] To see why, suppose that A is the formula $P(x)$ and B is $Q(x)$. Going from $\forall x[P(x) \wedge \exists x Q(x)]$ to $\forall x \exists x[P(x) \wedge Q(x)]$ changes the meaning of the formula completely, and the universal quantifier becomes obsolete. We do not want to bind more or different variables than in the original formula to each quantifier. Therefore, it will sometimes be necessary to change the variables' names before moving a quantifier from the inside of a formula to the outside. Going from

[9] If A is $P(x)$, then x is free in A but bound in $\forall x A$.

$\forall x[P(x) \land \exists x Q(x)]$ to $\forall x[P(x) \land \exists y Q(y)]$ and only then to $\forall x \exists y[P(x) \land Q(y)]$ keeps everything in good order.

In principle, we just showed that the following equivalences also hold:

$$\mathcal{Q}_1 x A \land \mathcal{Q}_2 x B \Leftrightarrow \mathcal{Q}_1 x \mathcal{Q}_2 y (A \land B),$$

$$\mathcal{Q}_1 x A \lor \mathcal{Q}_2 x B \Leftrightarrow \mathcal{Q}_1 x \mathcal{Q}_2 y (A \lor B).$$

Here, $\mathcal{Q}_i$ stands for any quantifier, so $\mathcal{Q}_1 x \mathcal{Q}_2 y$ could stand for any of $\forall x \forall y$, $\forall x \exists y$, $\exists x \forall y$, and $\exists x \exists y$. The chain of equivalences

$$\begin{aligned} \mathcal{Q}_1 x A \land \mathcal{Q}_2 x B &\Leftrightarrow \mathcal{Q}_1 x A \land \mathcal{Q}_2 y B \\ &\Leftrightarrow \mathcal{Q}_1 x (A \land \mathcal{Q}_2 y B) \\ &\Leftrightarrow \mathcal{Q}_1 x \mathcal{Q}_2 y (A \land B) \end{aligned}$$

shows how to go from left to right for the $\land$-connective. The rest follows similarly.

5.8 Logical consequence

Logical consequence in predicate logic is similar to logical consequence in propositional logic, except in that we must refer to interpretations in case of the latter—again, without interpretations, we cannot speak about true or false predicate logic formulas. In predicate logic, a formula is a logical consequence of a set of premises if and only if it is true in every interpretation in which the conjunction of premises is true.

Definition 5.11. $A_1, \ldots, A_n \models B$ if and only if B is true in at least every interpretation in which $A_1 \land \cdots \land A_n$ is true.

In addition to what we have already seen in connection to propositional logic, the quantifiers let us draw the following types of conclusions from sets of premises.

1. $\forall x A \models A[a/x]$
2. $A[a/x] \models \exists x A$
3. If $\mathcal{A} \models B[a/x]$ then $\mathcal{A} \models \forall x B$
4. If $\mathcal{A}, B[a/x] \models C$ then $\mathcal{A}, \exists x B \models C$

For (3) and (4), it is assumed that the constant a is not in any of the formulas of $\mathcal{A}$. We will use these types of logical consequences in later chapters of deduction in predicate logic.

Appending $[a/x]$ to a formula, as in $A[a/x]$, indicates that any free occurrence of x in the preceding formula is substituted with the term a. Hence if A is the formula $\exists y[P(x) \to Q(y, x)$, then $A[c/x]$ is equal to $\exists y[P(c) \to Q(y, c)$. Ensure that the substituting term c is **free for x in** A—a substituting term must not become bound by a quantifier. Hence, $A[y/x]$ would **not** have been correct since, upon replacement, the y would have become bound by the existential quantifier as in $\exists y[P(y) \to Q(y, y)$.

5.9 Prenex Normal Form

A formula in the prenex normal form has all the quantifiers to the left of the rest of the formula, which is either in the conjunctive normal form or disjunctive normal form. In this setting, we will only focus on formulas in conjunctive normal form.

Turning a formula to prenex conjunctive normal form follows a relatively mechanical process similar to the one for propositional logic, except this time, we need to take care to move all the quantifiers in front: To transform a formula into conjunctive normal form, follow these steps:

1. Move all negations inward to that only prime formulas are negated.[10] To deal with negated quantifiers, observe the equivalences
   (a) $\neg\forall xA \Leftrightarrow \exists x\neg A$
   (b) $\neg\exists xA \Leftrightarrow \forall x\neg A$
2. Transform any conditional or biconditional into their equivalent disjunctions and conjunctions.
3. Move the quantifiers to in front of the formula using the equivalences[11]
   (a) $\forall xA \land \forall xB \Leftrightarrow \forall x(A \land B)$
   (b) $\exists xA \lor \exists xB \Leftrightarrow \exists x(A \lor B)$
   (c) $A \land \forall xB \Leftrightarrow \forall x(A \land B)$
   (d) $A \lor \forall xB \Leftrightarrow \forall x(A \lor B)$
   (e) $A \land \exists xB \Leftrightarrow \exists x(A \land B)$
   (f) $A \lor \exists xB \Leftrightarrow \exists x(A \lor B)$
4. Distribute any disjunction over conjunctions so that any disjunction is a disjunction of literals.
5. Simplify the clauses such that each clause contains at most one instance of each formula letter, i.e., a clause with $P \lor P \lor \dots$ becomes $P \lor \dots$, and a clause with $P \lor \neg P \lor \dots$ become $\top$. If at least one clause distinct from $\top$ remains, remove all the clauses with $\top$. Otherwise, remove all but one of the $\top$s.

Example 5.6. Transform the formula

$$\exists z\,(\exists xQ(x,z) \lor \exists xP(x)) \to \neg\,(\neg\exists xP(x) \land \forall x\exists zQ(z,x))$$

to prenex conjunctive normal form. Starting by replacing the conditional leaves us with a negation in front of both conjuncts to move inward simultaneously.

[10] Depending on the formula, it may be easier to start with step 2 and only after performing step 1.

[11] Make sure to rename variables as necessary along the way or rename variables such that no two quantifiers bind variables with the same name.

Once the negations are moved so that each is directly in front of a prime formula, the quantifiers are moved outwards according to the previously presented rules.

$$\begin{gathered}
\exists z\,(\exists x Q(x,z) \lor \exists x P(x)) \to \neg\,(\neg\exists x P(x) \land \forall x \exists z Q(z,x)) \\
\neg\exists z\,(\exists x Q(x,z) \lor \exists x P(x)) \lor \neg\,(\neg\exists x P(x) \land \forall x \exists z Q(z,x)) \\
\forall z \neg\,(\exists x Q(x,z) \lor \exists x P(x)) \lor \neg\,(\neg\exists x P(x) \land \forall x \exists z Q(z,x)) \\
\forall z\,(\neg\exists x Q(x,z) \land \neg\exists x P(x)) \lor (\neg\neg\exists x P(x) \lor \neg\forall x \exists z Q(z,x)) \\
\forall z\,(\forall x \neg Q(x,z) \land \forall x \neg P(x)) \lor (\exists x P(x) \lor \exists x \neg\exists z Q(z,x)) \\
\forall z\,(\forall x \neg Q(x,z) \land \forall x \neg P(x)) \lor (\exists x P(x) \lor \exists x \forall z \neg Q(z,x)) \\
\forall z \forall x\,(\neg Q(x,z) \land \neg P(x)) \lor \exists x\,(P(x) \lor \forall z \neg Q(z,x)) \\
\forall z \forall x\,(\neg Q(x,z) \land \neg P(x)) \lor \exists y \forall z\,(P(y) \lor \neg Q(z,y)) \\
\forall z \forall x \exists y \big(\,(\neg Q(x,z) \land \neg P(x)) \lor \forall u\,(P(y) \lor \neg Q(u,y))\,\big) \\
\forall z \forall x \exists y \forall u \big(\,(\neg Q(x,z) \land \neg P(x)) \lor P(y) \lor \neg Q(u,y)\big) \\
\forall z \forall x \exists y \forall u \big(\,[(\neg Q(x,z) \lor P(y)) \land (\neg P(x) \lor P(y))] \lor \neg Q(u,y)\big) \\
\forall z \forall x \exists y \forall u \big([\neg Q(x,z) \lor P(y) \lor \neg Q(u,y)] \land [\neg P(x) \lor P(y) \lor \neg Q(u,y)]\big).
\end{gathered}$$

To be on the safe side, one can make a formula *clean* by giving all bound variables different names and not having any free variables with the same name as any bound variable. The formula $(\exists x P(x) \land \exists x Q(x)) \lor R(x)$ is not clean, but by changing the name of the second and third instances of x into, for example, y and z we obtain the clean formula $(\exists x P(x) \land \exists y Q(y)) \lor R(z)$. It is not a must to turn every formula into its clean counterpart every time quantifiers are to be moved, but it makes the process less error-prone. That is true regardless of the type of quantifier to be moved.

Example 5.7. The formula $\forall x G(x) \land \forall y F(y)$ is a clean formula so no change of names is needed before turning it into prenex conjunctive normal form by moving the quantifiers to the front of the formula to get $\forall x \forall y [G(x) \land F(y)]$ or $\forall y \forall x [G(x) \land F(y)]$.

Example 5.8. In the formula $\forall x G(x) \land \forall y F(x,y)$, the second instance of x is a free variable. None of the quantifiers binds it. The scope of the leftmost universal quantifier extends only over $G(x)$. However, when the leftmost quantifier is moved to in front of the formula, its scope will extend over the whole formula and thus bind the second instance of x. Such an operation would alter the formula. Before moving that first universal quantifier, we must change the name of either the first instance of x or the second instance of x (that constitutes the free variable). If we replace the first instance of x to z, we get $\forall z G(z) \land \forall y F(x,y)$. From there, we can move the quantifiers to in front of the formula without any issues to get $\forall z \forall y [G(z) \land F(x,y)]$; the x will remain a free variable.

Example 5.9. Going from $\forall x G(x) \land \forall y \forall x F(x,y)$ to $\forall x [G(x) \land \forall y \forall x F(x,y)]$ poses no problems, the second instance of x would still be bound by the third universal quantifier. Continuing to $\forall x \forall y [G(x) \land \forall x F(x,y)]$ is fine too. However, taking another step to $\forall x \forall y \forall x [G(x) \land F(x,y)]$ would be incorrect because, at that point, the third universal quantifier binds both xs—adding parentheses makes it easier to see that that is indeed the case: $\forall x (\forall y (\forall x [G(x) \land F(x,y)]))$. The solution is renaming the second instance of x to get $\forall x \forall y \forall z [G(x) \land F(z,y)]$, which is a correct prenex conjunctive normal form of the original formula.

5.10 Skolemization

The word Skolemization is due to the Norwegian logician Thoralf Skolem. To skolemize a predicate logic formula in prenex normal form means eliminating all the existential quantifiers by replacing them with constants and functions—these are called Skolem functions and Skolem constants.

Here we will refer to a specific quantifier by its position. In the formula $\forall x \exists y \forall z A$, the $\forall x$ is in the first position, $\exists y$ is in the second position, and $\forall z$ is in the third position.

A variable bound to an existential quantifier not preceded by a universal quantifier is replaced with a Skolem constant. A Skolem constant is essentially a label—it is not a specific individual in the domain of discourse. In a formula such as $\exists x \forall y P(x, y)$, the variable x is replaced by a Skolem constant, e.g., a, and we end up with $\forall y P(a, y)$. Again, we do not know what a points to, but if the original formula is true, there has to be some individual, represented by a, such that the solemnized formula is true as well.

Variables bound to existential quantifiers preceded by universal quantifiers are replaced by Skolem functions. A Skolem function takes as many arguments as there are universal quantifiers to the left of the existential quantifier in question. In a formula with an existential quantifier preceded by two universal quantifiers, such as $\forall x \forall y \exists z P(x, y, z)$, the variable bound to the existential quantifier is replaced by a function with two arguments consisting of the variables bound to the preceding universal quantifiers, as in $\forall x \forall y P(x, y, f(x, y))$.

Here is an example of a real-life Skolem function that was presented to me by David Sundgren. Suppose you work at an auto parts store. You have a catalog for carburetors with two columns. The left column contains the car models, and the right column contains the carburetors, with the carburetor for a specific model placed on the same line as that model. The models and the carburetors constitute the domain of discourse. Suppose that we have two predicates. The predicate P is true of all car models, and the predicate Q is true of all pairs with the model in the first position and the corresponding carburetor in the second position. To state that every model has a corresponding carburetor, we write $\forall x \exists y\,(P(x) \rightarrow Q(x, y))$. A Skolem function is created such that it maps a car model to some carburetor. Substituting such a Skolem function for the variable bound by the existential quantifier yields $\forall x\,(P(x) \rightarrow Q(x, f(x)))$. The function x takes a car model as an argument and outputs the correct carburetor—the one on the same line as the car model in the catalog.

Example 5.10. Take the formula

$$\forall z \forall x \exists y \forall u \big([\neg Q(x, z) \vee P(y) \vee \neg Q(u, y)] \wedge [\neg P(x) \vee P(y) \vee \neg Q(u, y)]\big)$$

in prenex conjunctive normal form and skolemize it. Any instance of the variable y will be substituted by a Skolem function f that takes two arguments z and x—the variable x is bound by an existential quantifier that is preceded by universal quantifiers binding z and x respectively. By skolemizing the formula we obtain

$$\forall z \forall x \forall u \big([\neg Q(x, z) \vee P(f(z, x)) \vee \neg Q(u, f(z, x))] \wedge \\ [\neg P(x) \vee P(f(z, x)) \vee \neg Q(u, f(z, x))]\big).$$

Example 5.11. A formula $\exists x \forall y \exists z [(P(x) \wedge Q(y)) \vee R(z)]$ will have the variable bound by the existential quantifier at the very beginning of the formula substituted by a Skolem constant. The variable bound by the second existential quantifier is substituted by a function that takes the variable y, bound by the preceding universal quantifier, as its argument. Hence we obtain $\forall y[(P(a) \wedge Q(y)) \vee R(f(y))]$ from skolemizing the formula.

6 Resolution in Predicate Logic

Resolution in predicate logic are essentially the same as in propositional logic. The fundamental idea behind the *resolution rule*

$$(A \vee P), (B \vee \neg P) \vdash A \vee B$$

is the same as for propositional logic. However, the formulas involved usually need some preparation, and the resolution rule for predicate logic requires additional or auxiliary steps because variables, functions, and constants are involved.

Unless otherwise stated, we assume that all formulas are in prenex conjunctive normal form and skolemized. Furthermore, we assume that any variable is universally quantified even though no quantifiers will be written out explicitly. Thus, we write $P(x)$ instead of $\forall x P(x)$.

Since two formulas $P(x)$ and $P(y)$ are similar but not equal, a resolution such as

(1)	$A \vee P(x)$	Given
(2)	$B \vee \neg P(y)$	Given
(3)	$A \vee B$	Resolvent 1, 2

is not permitted—the formulas that are resolved have to be equal. To make formulas equal, if at all possible, we will use *substitution* such that formulas can be *unified*. Substitution is replacing variables with other variables, functions, or constants. Two formulas are said to be unified by a substitution if that substitution makes them equal.

6.1 Substitution

Assume a formula $P(x)$. Substituting a variable y for x yields $P(y)$. Substituting a function $f(z)$ for x in $P(x)$ yields $P(f(z))$, and so forth. To indicate such substitutions we write $[y/x]$ and $[f(z)/x]$ respectively. Hence, $P(x)[y/x] = P(y)$. Sometimes we will see expressions like $A[y/x]$, with A seemingly without any variables. However, here A is just a short name for a formula such as $P(x)$ or $P(x) \vee Q(x)$.

A substitution can involve multiple and simultaneous replacements of variables. For example, substituting y for x, and a for y; that would be written as $[y/x, a/y]$, and $P(x) \vee Q(y)[y/x, a/y] = P(y) \vee Q(a)$. Note that only the original y is substituted with a as the substitution is done to all variables simultaneously. The y is substituted for x at the same exact time as a is substituted for y.

Substitutions can be chained. Here is an example: $[f(a)/x] = [f(y)/x][a/y]$. Start by substituting $f(y)$ for x, e.g., $P(x)[f(y)/x] = P(f(y))$, and then sustitute a for y in $P(f(y))$. In this example, we would obtain $P(f(a))$.

To save space or to make it easier to reason at a more general level, we can assign variables to substitutions. For example, let $\alpha = [f(y)/x]$ and $\beta = [a/y]$, then we have $\gamma = [f(a)/x] = \alpha\beta$ as above.

A substitution such as $[y/x]$, applied to a formula $P(y)$ or $P(z)$ that does not have the variables to be substituted—in this case x—does not change anyting. Hence, $P(y)[y/x] = P(y)$ and $P(z)[y/x] = P(z)$.

6.2 Unification

A substitution that makes two formulas A and B equal is called a *unifier* of A and B. The formulas $P(x)$ and $P(y)$ are not equal but $P(x)[y/x] = P(y)$ and $P(y)[y/x] = P(y)$ are. Hence $[y/x]$ is a unifier of $P(x)$ and $P(y)$. Similarly, the substitution $[z/x, z/y$ is also a unifier of $P(x)$ and $P(y)$ since $P(x)[z/x] = P(z) = P(y)[z/y]$.

Since only variables can be substituted, for the formulas $P(a)$ and $P(x)$, with a a constant, the only unifier that is not a chain of substitutions is $[a/x]$. Any composite substitution such as $[y/x][z/y][a/z]$ would in the end, be equal to $[a/x]$. Furthermore, $P(a)$ and $P(b)$, with a and b constants, do not have a unifier.

A unifying substitution can be more or less general. The meaning of *general* can intuitively be understood in terms of the constraint put on the resolvent in a subsequent resolution. For example, if the formulas $\neg P(x) \vee Q(x)$ and $P(y)$ are unified by the substitution $[a/x, a/y]$, then we obtain the following resolvent $Q(a)$, from $\neg P(a) \vee Q(a)$ and $P(a)$, which says that Q is true only of a. However, the unifying substitution $[x/y]$ yields $\neg P(x) \vee Q(x)$ and $P(x)$, with the resolvent $Q(x)$, which says that Q is true of all individuals of the domain of discourse, which is quite a difference from being true only a—$Q(a)$ is more constrained than $Q(x)$.

Definition 6.1. A unifier β is *at least as general* as α if and only if there exists a unifier γ such that $\alpha = \beta\gamma$.

Definition 6.2. A unifier is a *most general* unifier if it is at least as general as every other unifier (of the same terms).

There is an algorithm called the *most general unifier algorithm* that unifies two formulas in the most general way—the algorithm does not result in a most general unifier; it rather acts as a most general unifier. Assume two formulas A and B that are to be unified. Then the algorithm is as follows:

1. START: Point to the leftmost symbols of A and B respectively.
2. IF the pointers are equal (i.e., they point to the same symbols in each formula) THEN
   - (a) IF the pointers are past the rightmost positions, THEN GOTO (4)
   - (b) ELSE move the pointers one step to the right and GOTO (2)
3. ELSE
   - (a) IF both pointers are at non-variables, THEN halt and fail
   - (b) ELSE
     - i. IF both pointers are variables, THEN substitute one for the other throughout both formulas, move the pointers one step to the right, and GOTO (2)

   ii. IF one pointer is a function without the other variable as an argument THEN substitute the variable with the function throughout both formulas, move one step to the right and GOTO (2)
   iii. IF one pointer is a constant, then substitute that constant for the variable throughout both formulas, move one step to the right, and GOTO (2)

4. The formulas are identical. HALT.

Example 6.1. Here is one application of the algorithm to the formulas $P(x, x)$ and $P(f(y), f(b))$ with x and y variables and b a constant. The overbar, e.g., $\bar{P}$, will serve as a pointer representation.

1. $\bar{P}(x, x)$ and $\bar{P}(f(y), f(b))$
2. $P(\bar{x}, x)$ and $P(\bar{f}(y), f(b))$
3. $P(f(\bar{y}), f(y))$ and $P(f(\bar{y}), f(b))$
4. $P(f(y), \bar{f}(y))$ and $P(f(y), \bar{f}(b))$
5. $P(f(y), f(\bar{y}))$ and $P(f(y), f(\bar{b}))$
6. $P(f(b), f(b)\bar{)}$and $P(f(b), f(b)\bar{)}$

6.3 The Rules Involved

The basic principles are the same as for propositional logic, but the rules for predicate logic need to account for unification, both when it comes to the *resolution rule* and the simplification of formulas which are here done by applying the factorization rule.

The resolution rule. Assume two clauses $\mathcal{A} \vee A$ and $\mathcal{B} \vee \neg B$, and a unifier γ such that $A\gamma = B\gamma$, then

$$\mathcal{A} \vee A, \mathcal{B} \vee \neg B \vdash (\mathcal{A} \vee \mathcal{B})\gamma.$$

Example 6.2. Resolve $Q(y) \vee P(x)$ and $R(t) \vee \neg P(y)$[12]. Choose $\gamma = [x/y]$ as the unifier of $P(x)$ and $\neg P(y)$.

(1)	$Q(y) \vee P(x)$	Given
(2)	$R(t) \vee \neg P(y)$	Given
(3)	$Q(x) \vee R(t)$	Resolvent 1, 2 with MGU $\gamma = [x/y]$

The process involved, going from line 2 to line 3 is as follows:

1. We find a most general unifier of $P(x)$ and $P(y)$ since these are the ones to be the resolving literals. We choose $[x/y]$ as the most general unifier.[13]
2. Applying the most general unifier $[x/y]$ to $Q(y) \vee P(x)$ and $R(t) \vee \neg P(y)$, we obtain $Q(x) \vee P(x)$ and $R(t) \vee \neg P(x)$ as the result.

[12] This particular example is only intended to show a single application of the resolution rule.

[13] The substitution $[y/x]$ is also a most general unifier, and we could have chosen that instead.

3. Resolving the clauses $Q(x) \lor P(x)$ and $R(t) \lor \neg P(x)$ finally yields the resolvent $Q(x) \lor R(t)$.

It may seem unnecessary to apply the substitution $[y/x]$ also to $Q(y) \lor P(x)$ since the resolving literal $P(x)$ already has the needed variable name x. However, a unifier is always applied to both formulas that are to be unified, regardless of the precise substitutions involved. In this case, specifically, we needs to make sure that also the y in $Q(y)$ is replaced with an x—subsequent resolution steps involving a previous resolvent has to take any previous unifying substitutions into account.

The factorization rule. Assume a clause $A_1 \lor A_2 \lor \mathcal{A}$ and a unifier β such that $A_1\beta = A_2\beta$. Then, given the substitution β, we can simplify the formula by removing one of $A_i\beta$. The resulting formula will retain the substitution acting as a unifier. The rule is

$$A_1 \lor A_2 \lor \mathcal{A} \vdash (A_i \lor \mathcal{A})\beta \quad \text{with } A_1\beta = A_2\beta.$$

6.4 Examples

Example 6.3. Show that $P(x, f(x))$, $\neg P(s, y) \lor P(z, s)$, and $\neg P(a, b)$ are inconsistent by applying resolution.

(1)	$P(x, f(x))$	Given
(2)	$\neg P(s, y) \lor P(z, s)$	Given
(3)	$\neg P(a, b)$	Given
(4)	$P(z, s)$	Resolvent 1, 2 with MGU $[s/x, f(s)/y]$
(5)	$\perp$	Resolvent 3, 4 with MGU $[a/z, b/s]$

On line 4, the substitution $[s/x, f(s)/y]$ unifies $P(x, f(x))$ and $P(s, y)$ such we can resolve based on $P(s, f(s))$ and $\neg P(s, f(s))$.

6.5 The (hidden) Universal Quantifiers

It may seem odd that the quantifiers suddenly are removed once we have the formulas on prenex conjunctive normal form and skolemized. However, since the universal quantifier distributes over $\land$, it poses no problem—in a sense, we can consider them to be there implicitly, although they are not visible. For example, consider the formula

$$\forall s \forall x \forall y \forall z [P(x, f(x)) \land (\neg P(s, y) \lor P(z, s)) \land \neg P(a, b)]$$

which is already skolemized and on prenex conjunctive normal form. We are allowed to distribute the universal quantifiers over the conjunctions, and that yields

$$\forall s \forall x \forall y \forall z P(x, f(x)) \land \forall s \forall x \forall y \forall z (\neg P(s, y) \lor P(z, s)) \land \forall s \forall x \forall y \forall z \neg P(a, b).$$

A quantifier that binds a variable not part of the formula in its scope can be removed. Doing so gives us the shorter version

$$\forall x P(x, f(x)) \land \forall s \forall y \forall z (\neg P(s, y) \lor P(z, s)) \land \neg P(a, b).$$

Since we are working toward resolving clauses, we can already substitute x for s in the second clause and obtain

$$\forall xP(x,f(x)) \land \forall x\forall y\forall z(\neg P(x,y) \lor P(z,x)) \land \neg P(a,b).$$

Now, since $\neg P(x,y)$ is true for any individual substituted for y, it must also be true for $f(x)$—the result of applying the function f to some individual in the domain of discourse will have as output some individual in the domain of discourse. Therefore, we can substitute $f(x)$ for y to obtain

$$\forall xP(x,f(x)) \land \forall x\forall z(\neg P(x,f(x)) \lor P(z,x)) \land \neg P(a,b).$$

Note that the universal quantifier that bound y became obsolete and was removed in the process. Next, we can resolve $P(x,f(x))$ and $\neg P(x,f(x))$ resulting in

$$\forall x\forall zP(z,x) \land \neg P(a,b).$$

Finally, since $P(z,x)$ is true of any choice of individuals substituting these variables, we can safely assume that if $P(z,x)$, then also $P(a,b)$. Hence,

$$P(a,b) \land \neg P(a,b) \Leftrightarrow \bot.$$

Example 6.4. Show that $\exists xP(x,x) \to \exists x\exists yP(x,y)$ is valid. Since that formula is valid if and only if its counterexample set is inconsistent, we try to resolve $\bot$ from $\neg[\exists xP(x,x) \to \exists x\exists yP(x,y)]$.

Start by writing the formula (i.e., the negation of the original formula that we assume to be valid) on prenex conjunctive normal form.

$$\begin{gathered}
\neg[\exists xP(x,x) \to \exists x\exists yP(x,y)] \\
\Leftrightarrow \neg[\neg\exists xP(x,x) \lor \exists x\exists yP(x,y)] \\
\Leftrightarrow \neg\neg\exists xP(x,x) \land \neg\exists x\exists yP(x,y) \\
\Leftrightarrow \exists xP(x,x) \land \forall x\forall y\neg P(x,y) \\
\Leftrightarrow \exists x[P(x,x) \land \forall z\forall y\neg P(z,y)] \\
\Leftrightarrow \exists x\forall z\forall y[P(x,x) \land \neg P(z,y)].
\end{gathered}$$

Then skolemize the formula which in this case amounts to replacing the x with a constant:

$$\forall z\forall y[P(a,a) \land \neg P(z,y)].$$

The resulting clauses are $P(a,a)$ and $\neg P(z,y)$; apply resolution to these.

(1)	$P(a,a)$	Given
(2)	$\neg P(z,y)$	Given
(5)	$\bot$	Resolvent 1, 2 with MGU $[a/z, a/y]$

We resolved $\bot$ from the counterexample set and thus concludes the original formula $\exists xP(x,x) \to \exists x\exists yP(x,y)$ to be valid.

Example 6.5. Show that $\forall x[F(x) \to G(x)] \vdash \neg\forall x\neg F(x) \to \neg\forall x\neg G(x)$. The counterexample set consists of the premises and the negation of the conclusion:

$$\forall x[F(x) \to G(x)] \land \neg[\neg\forall x\neg F(x) \to \neg\forall x\neg G(x)].$$

Start by writing the formula of the counterexample set on prenex conjunctive normal form:

$$\begin{gathered}
\forall x[F(x) \to G(x)] \land \neg[\neg\forall x\neg F(x) \to \neg\forall x\neg G(x)] \\
\Leftrightarrow \forall x[\neg F(x) \lor G(x)] \land [\neg\forall x\neg F(x) \land \neg\neg\forall x\neg G(x)] \\
\Leftrightarrow \forall x[\neg F(x) \lor G(x)] \land \exists x\neg\neg F(x) \land \forall x\neg G(x) \\
\Leftrightarrow \forall y[\neg F(y) \lor G(y)] \land \exists x F(x) \land \forall z\neg G(z) \\
\Leftrightarrow \exists x\{\forall y[\neg F(y) \lor G(y)] \land F(x) \land \forall z\neg G(z)\} \\
\Leftrightarrow \exists x\forall y\{[\neg F(y) \lor G(y)] \land F(x) \land \forall z\neg G(z)\} \\
\Leftrightarrow \exists x\forall y\forall z\{[\neg F(y) \lor G(y)] \land F(x) \land \neg G(z)\}.
\end{gathered}$$

Skolemize the resulting formula by replacing x with a Skolem constant to get

$$\forall y\forall z\{[\neg F(y) \lor G(y)] \land F(a) \land \neg G(z)\}.$$

Apply resolution to the clauses $\neg F(y) \lor G(y)$, $F(a)$, and $\neg G(z)$:

(1)	$\neg F(y) \lor G(y)$	Given
(2)	$F(a)$	Given
(3)	$\neg G(z)$	Given
(4)	$G(a)$	Resolvent 1, 2 with MGU $[a/y]$
(5)	$\bot$	Resolvent 3, 4 with MGU $[a/z]$

Since we could derive $\bot$ from the counterexample set we have shown that

$$\forall x[F(x) \to G(x)] \vdash \neg\forall x\neg F(x) \to \neg\forall x\neg G(x).$$

Example 6.6. Show that $\exists x\forall y P(x, y) \to \forall y\exists x P(x, y)$ is valid.

To show the validity of said formula using resolution, we try to derive a contradiction from its counterexample set $\neg[\exists x\forall y P(x, y) \to \forall y\exists x P(x, y)]$. Start by writing the formula of the counterexample set on prenex normal conjunctive normal form:

$$\begin{gathered}
\neg[\exists x\forall y P(x, y) \to \forall y\exists x P(x, y)] \\
\Leftrightarrow \exists x\forall y P(x, y) \land \neg\forall y\exists x P(x, y) \\
\Leftrightarrow \exists x\forall y P(x, y) \land \exists y\neg\exists x P(x, y) \\
\Leftrightarrow \exists x\forall y P(x, y) \land \exists y\forall x\neg P(x, y) \\
\Leftrightarrow \exists x\forall y P(x, y) \land \exists s\forall t\neg P(t, s) \\
\Leftrightarrow \exists x[\forall y P(x, y) \land \exists s\forall t\neg P(t, s)] \\
\Leftrightarrow \exists x\exists s[\forall y[P(x, y) \land \forall t\neg P(t, s)] \\
\Leftrightarrow \exists x\exists s\forall y[P(x, y) \land \forall t\neg P(t, s)] \\
\Leftrightarrow \exists x\exists s\forall y\forall t[P(x, y) \land \neg P(t, s)]
\end{gathered}$$

Skolemize the formula by replacing x and s with Skolem constants to get

$$\forall y\forall t[P(a, y) \land \neg P(t, b)].$$

Apply resolution to the clauses $P(a, y)$ and $\neg P(t, b)$:

(1)	$P(a, y)$	Given
(2)	$\neg P(t, b)$	Given
(3)	$\perp$	Resolvent 1, 2 with MGU $[a/t, b/y]$

Since we could derive $\perp$ from the counterexample set, the original formula $\exists x \forall y P(x, y) \to \forall y \exists x P(x, y)$ is valid—that, we can also write as

$$\vdash \exists x \forall y P(x, y) \to \forall y \exists x P(x, y).$$

7 Semantic Tableaux in Predicate Logic

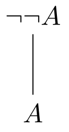
The method of semantic tableaux for predicate logic rests on the same fundamental idea as the one behind semantic tableaux for propositional logic, namely by creating a tableau, or tree, with each branch corresponding to a case. Suppose we have a formula $A \vee B$. Then, if both A and B are inconsistent, $A \vee B$ is false. Here, we consider A a case and B another—we test whether A and B are inconsistent separately.

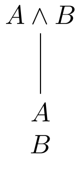
To test if $\mathcal{A} \models \mathcal{B}$, with $\mathcal{A}$ possibly empty, we construct a semantic tableau of the sentences in $\mathcal{A}$ together with $\neg B$ (i.e., a semantic tableau of the counterexample set). If all the branches close—meaning all branches contain inconsistencies—that shows the counterexample set to be inconsistent, and therefore $\mathcal{A} \models \mathcal{B}$ to be valid.

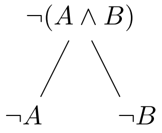
If we can show that $A_1, \dots, A_n$ implies B using a semantic tableau, we write $A_1, \dots, A_n \vdash B$ (note the use of $\vdash$ instead of $\models$ to underline the use of a deductive system of rules rather than reasoning based on semantics). Hence, if the semantic tableau of $A_1, \dots, A_n, \neg B$ have all branches closed, then $A_1, \dots, A_n \vdash B$. Just as with resolution, we prove the validity of a formula or logical implication by showing that the corresponding counterexample set is inconsistent—that goes back to Chapter 1.

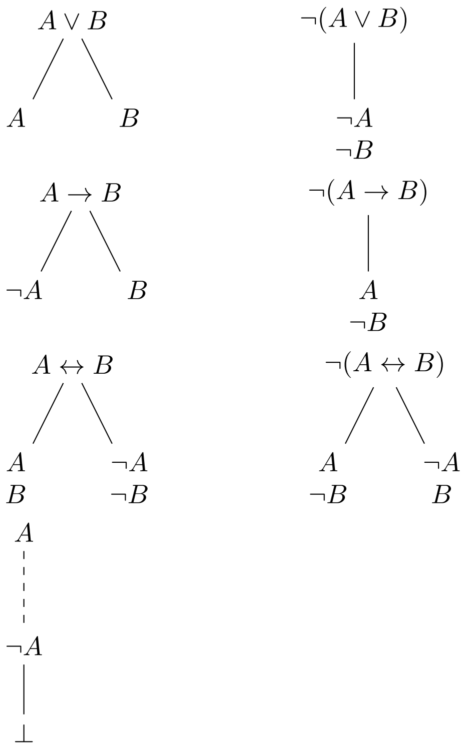
The method of semantic tableaux for predicate logic is sound and complete, and the same is true for resolution. It is sound because anytime we can show that $\mathcal{A} \vdash \mathcal{B}$, we also have $\mathcal{A} \models \mathcal{B}$. It is complete since whenever $\mathcal{A} \models \mathcal{B}$, we can show that $\mathcal{A} \vdash \mathcal{B}$. In other words, we will have a closed tableau of $A_1, \dots, A_n, \neg B$ if and only if $A_1, \dots, A_n \models B$.

The rules for the logical connectives are the same in both propositional and predicate logic semantic tableaux. They are repeated here for easy access.

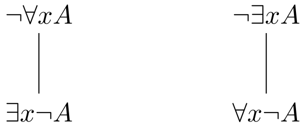




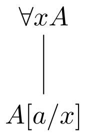




In addition to the above, we need rules to handle quantified formulas. First, we have rules corresponding to some of the equivalences concerning negated predicate logic formulas, where the quantifier is flipped as the negation symbol is moved inward.



Next, we need rules for eliminating the quantifiers and substituting constant symbols for the variables. We want to instantiate as few constant symbols as possible throughout a semantic tableau. It is contradictions that will close the branches, and while $P(a) \land \neg P(a)$ is a contradiction, $P(a) \land \neg P(b)$ is not since a and b are potential names of different individuals in the domain of discourse. So, even if $\forall xP(x)$ means that P is true of any individual in the domain of discourse, we want to constrain the number of individuals we refer to. If we already have $P(a)$, then we want to instantiate $\neg P(a)$ from $\forall x\neg P(x)$ rather than $\neg P(b)$. Therefore, the basic rule for the universal quantifier is



where the constant symbol a already occuring on the same branch.

In other words, we will refer to individuals already referred to as far as possible.

The existential quantifier works differently. After all, we cannot assume that just because $P(a)$ and $\exists x \neg P(x)$, it follows that $\neg P(a)$. It might as well be that $\neg P(b)$ or $\neg P(c)$, given $\exists x \neg P(x)$. Whenever we instantiate an existentially quantified variable, we pick a constant symbol not used above on the same branch, assuming that that constant symbol is a label for an object of which the formula in question is true. If $\exists x Q(x)$, we might say, e.g., $Q(b)$, assuming that the constant symbol b points to an individual of which Q is true, even without knowing exactly which individual that is. Therefore, the rule for eliminating an existential quantifier is

$$\begin{array}{c} \exists x A \\ \Big| \\ A[b/x] \end{array}$$

with b not occuring anywhere above on the same branch.

As a consequence of the above, an existential quantifier should be eliminated before a universal quantifier when there is a choice. However, sometimes eliminating a universal quantifier is the only possibility. Assuming that the domain of discourse has at least one individual, the additional rule for eliminating a universal quantifier poses no problem.

$$\begin{array}{c} \forall x A \\ \Big| \\ A[c/x] \end{array}$$

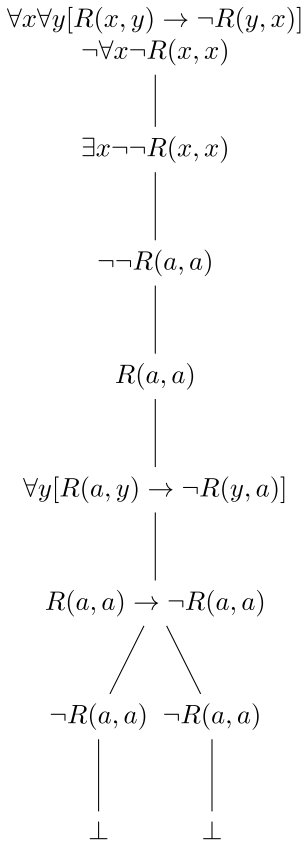
for any choice of c, only if no constant symbol has been introduced so far on the same branch.

Here are some examples of semantic tableaux for valid arguments and formulas.

Example 7.1. Prove that $\forall x \forall y [R(x,y) \rightarrow \neg R(y,x)] \models \forall x \neg R(x,x)$ using the method of semantic tableaux. The counterexample set is

$$\{\forall x \forall y [R(x,y) \rightarrow \neg R(y,x)], \neg \forall x \neg R(x,x)\}.$$

The resulting semantic tableau has all its branches closed, which proves the inconsistency of the counterexample set. If the counterexample set is inconsistent, then the original argument is valid.

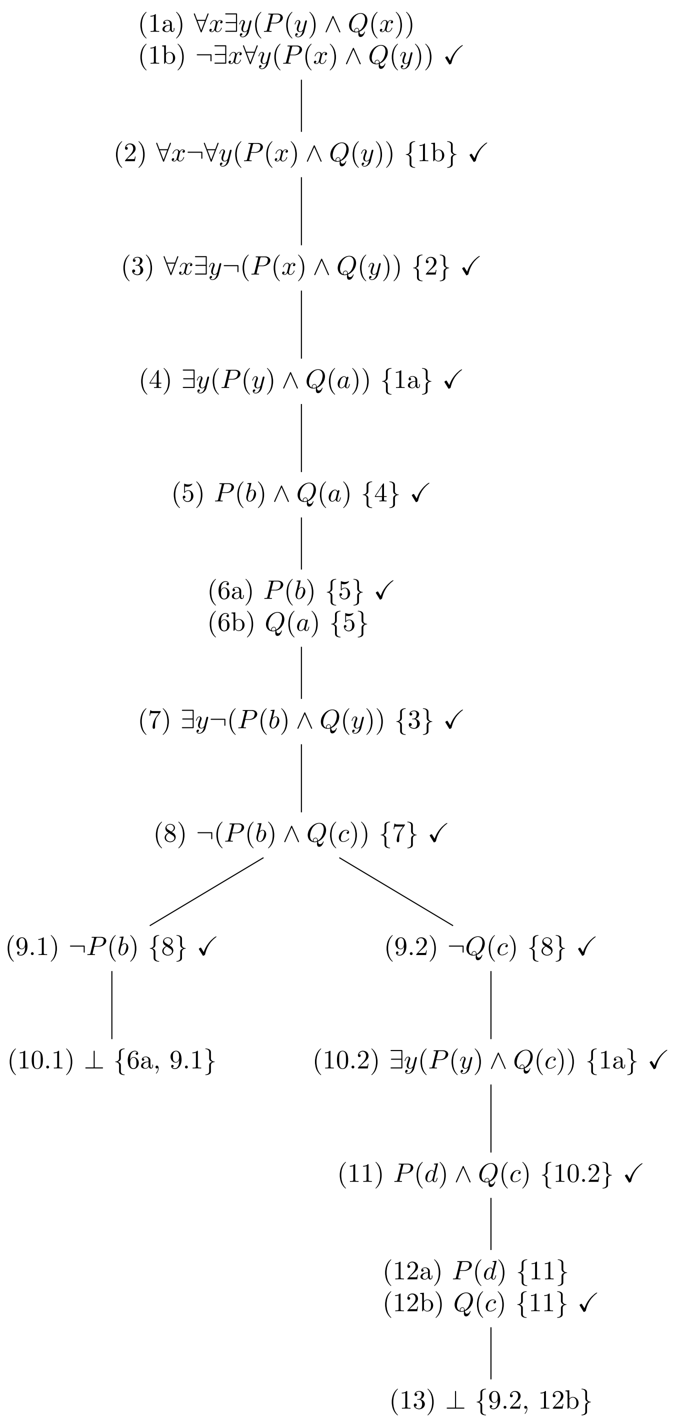

$\forall x \forall y[R(x,y) \to \neg R(y,x)]$
$\neg \forall x \neg R(x,x)$
$\exists x \neg\neg R(x,x)$
$\neg\neg R(a,a)$
$R(a,a)$
$\forall y[R(a,y) \to \neg R(y,a)]$
$R(a,a) \to \neg R(a,a)$
$\neg R(a,a)$
$\neg R(a,a)$
$\perp$
$\perp$

Because the instantiation of the same bound variables can be performed multiple times, we will only mark quantified formulas as discharged once it is evident they will not be instantiated again or if the quantifiers are interspersed with one or several negation signs that we are to move inward.[14] However, it can be helpful to make explicit the origin of an instantiation; and we can still put checkmarks after each formula without variables. To do that, label each node in the tableau with its corresponding node position[15]. When applying a quantifier rule, note which node the rule is applied to. Let us look at an example with such labels.

Example 7.2. Show that $\forall x \exists y(P(y) \land Q(x)) \models \exists x \forall y(P(x) \land Q(y))$.

[14] The use of checkmarks is not necessary and can be done in different ways due to personal preference, but it can support the process by keeping track of which formulas have been dealt with so far.

[15] We can label nodes in several different ways. Here we adopt a labeling system (x, y), where x is the depth and y is the number of nodes from the left (for each level). When multiple formulas share the same node, we add a letter, as in (1.2b).

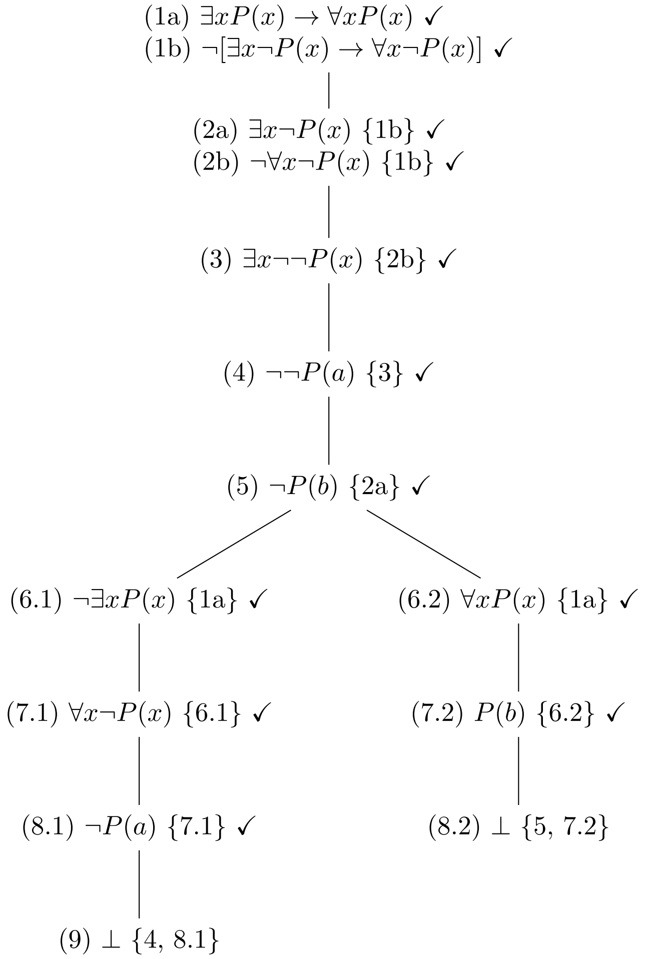

(1a) $\forall x \exists y (P(y) \land Q(x))$
(1b) $\neg \exists x \forall y (P(x) \land Q(y))$ ✓
(2) $\forall x \neg \forall y (P(x) \land Q(y))$ {1b} ✓
(3) $\forall x \exists y \neg (P(x) \land Q(y))$ {2} ✓
(4) $\exists y (P(y) \land Q(a))$ {1a} ✓
(5) $P(b) \land Q(a)$ {4} ✓
(6a) $P(b)$ {5} ✓
(6b) $Q(a)$ {5}
(7) $\exists y \neg (P(b) \land Q(y))$ {3} ✓
(8) $\neg (P(b) \land Q(c))$ {7} ✓
(9.1) $\neg P(b)$ {8} ✓
(9.2) $\neg Q(c)$ {8} ✓
(10.1) $\perp$ {6a, 9.1}
(10.2) $\exists y (P(y) \land Q(c))$ {1a} ✓
(11) $P(d) \land Q(c)$ {10.2} ✓
(12a) $P(d)$ {11}
(12b) $Q(c)$ {11} ✓
(13) $\perp$ {9.2, 12b}

Since both formulas begin with a universal quantifier once the negation sign in (1b) has been moved inward, starting by immediately instantiating x in (1a) would certainly have been in accordance with the rules. By moving any negations inward before instantiating any variables, however, we are sure not to miss any formulas that start with an existential quantifier by accident—it is simply a matter of practice and not something called upon by some rule.[16]

By eliminating the quantifiers in (1a) before continuing with (3), we maintain the single branch as long as possible. To minimize the growth of the tree—breadth-wise, that is—it is generally good practice to postpone any application of a rule that splits the current branch into two branches.

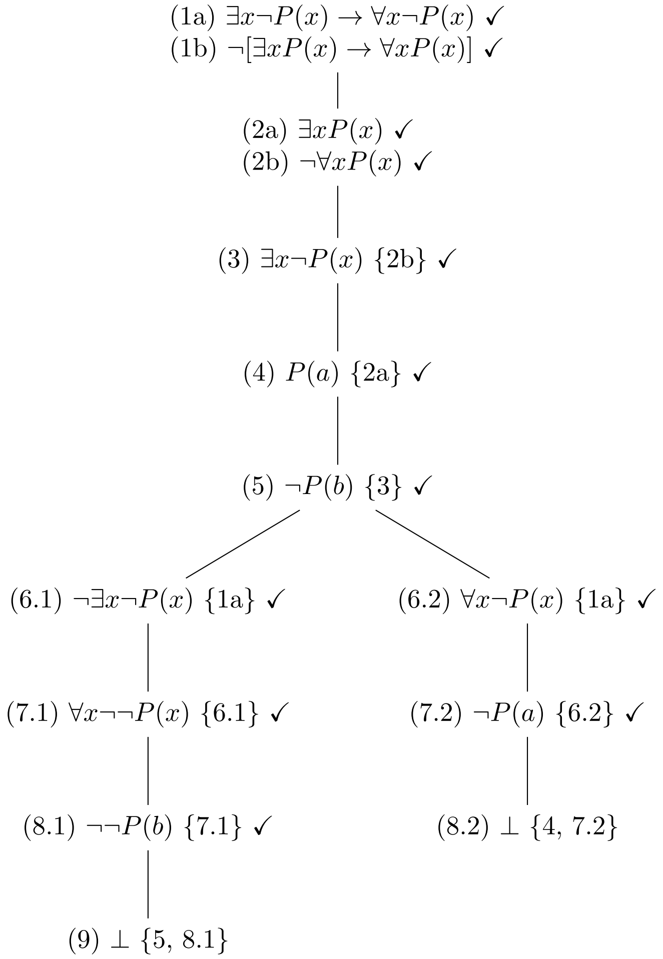
To obtain $Q(c)$ in (12b), we had to reinstantiate the universally quantified variable x in (1a). As a consequence, we ended up with the superfluous $P(d)$ in (12a) since we had to eliminate the existential quantifier in (10.2) before being able to get $Q(c)$ separately. That unused $P(d)$, however, did not keep the branch from closing—it caused no problems.

Example 7.3. To check if $\exists x P(x) \to \forall x P(x) \dashv\vdash \exists x \neg P(x) \to \forall x \neg P(x)$, we need to check whether the left side implies and right side, and then whether the right side implies the left side. We start by checking whether the left side implies the right side.


(1a) $\exists x P(x) \to \forall x P(x)$ ✓
(1b) $\neg[\exists x \neg P(x) \to \forall x \neg P(x)]$ ✓
(2a) $\exists x \neg P(x)$ {1b} ✓
(2b) $\neg \forall x \neg P(x)$ {1b} ✓
(3) $\exists x \neg\neg P(x)$ {2b} ✓
(4) $\neg\neg P(a)$ {3} ✓
(5) $\neg P(b)$ {2a} ✓
(6.1) $\neg \exists x P(x)$ {1a} ✓
(6.2) $\forall x P(x)$ {1a} ✓
(7.1) $\forall x \neg P(x)$ {6.1} ✓
(7.2) $P(b)$ {6.2} ✓
(8.1) $\neg P(a)$ {7.1} ✓
(8.2) $\perp$ {5, 7.2}
(9) $\perp$ {4, 8.1}

[16] Some authors state the rules for $\neg \forall x A$ and $\neg \exists y B$ such that they result in $\neg A[a/x]$ with a not already occurring on the same branch, and $\neg B[b/y]$ with b already occurring on the same branch, respectively. It can lessen the number of steps, but the result is the same as when using the rules defined in this text.

All the branches close when going from left to right, thus we have so far showed that $\exists x P(x) \to \forall x P(x) \vdash \exists x \neg P(x) \to \forall x \neg P(x)$. Let us check the other direction to see if the right side implies the left side.

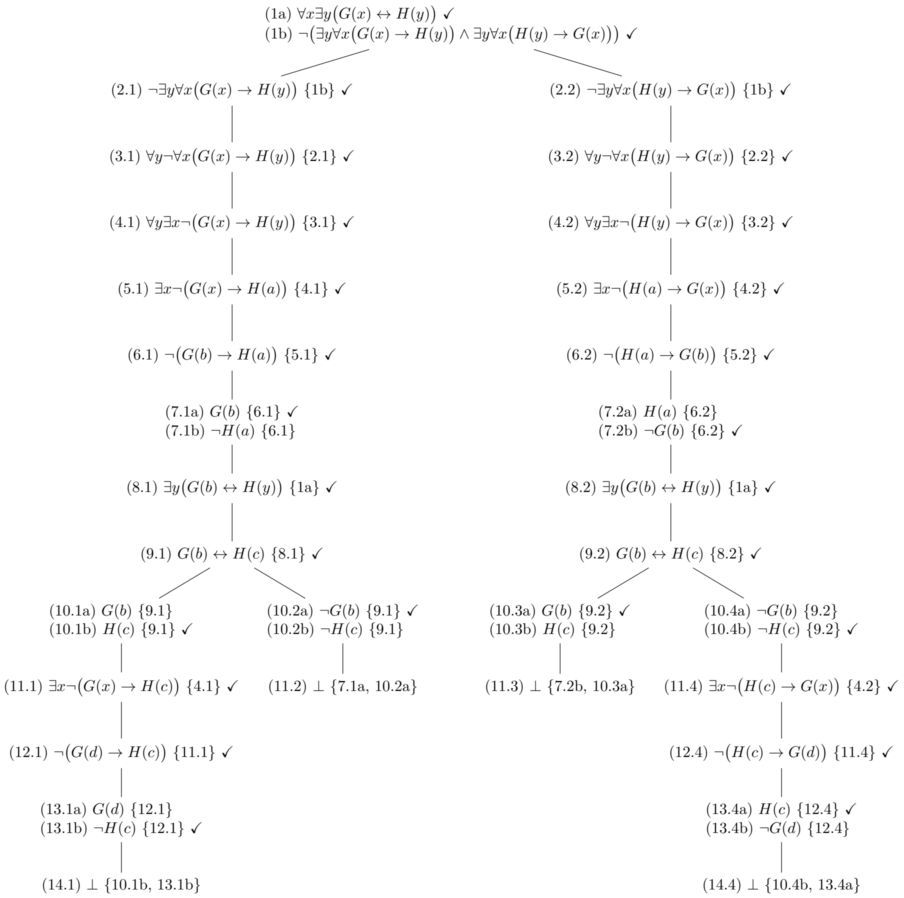

(1a) $\exists x \neg P(x) \to \forall x \neg P(x)$ ✓
(1b) $\neg[\exists x P(x) \to \forall x P(x)]$ ✓
(2a) $\exists x P(x)$ ✓
(2b) $\neg \forall x P(x)$ ✓
(3) $\exists x \neg P(x)$ {2b} ✓
(4) $P(a)$ {2a} ✓
(5) $\neg P(b)$ {3} ✓
(6.1) $\neg \exists x \neg P(x)$ {1a} ✓
(6.2) $\forall x \neg P(x)$ {1a} ✓
(7.1) $\forall x \neg\neg P(x)$ {6.1} ✓
(7.2) $\neg P(a)$ {6.2} ✓
(8.1) $\neg\neg P(b)$ {7.1} ✓
(8.2) $\bot$ {4, 7.2}
(9) $\bot$ {5, 8.1}

In this case also, all branches close, hence $\exists x \neg P(x) \to \forall x \neg P(x) \vdash \exists x P(x) \to \forall x P(x)$. Considering the result of both tableaux, we have shown that indeed $\exists x P(x) \to \forall x P(x) \dashv\vdash \exists x \neg P(x) \to \forall x \neg P(x)$.

Example 7.4. To show that

$$\forall x \exists y \big(G(x) \leftrightarrow H(y)\big) \vdash \exists y \forall x \big(G(x) \to H(y)\big) \land \exists y \forall x \big(H(y) \to G(x)\big)$$

holds, using the method of semantic tableaux, requires considerably more space than the previous examples. With large tableaux like this, choosing which formula to start with can have significant implications on the growth of the resulting tableau.

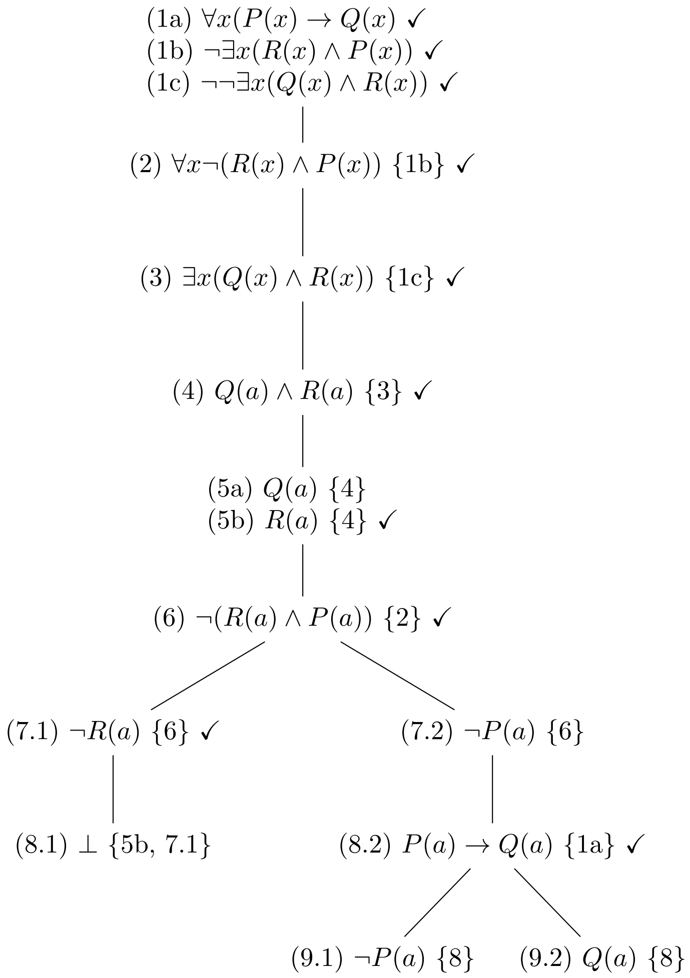

(1a) ∀x∃y(G(x) ↔ H(y)) ✓
(1b) ¬(∃y∀x(G(x) → H(y)) ∧ ∃y∀x(H(y) → G(x))) ✓
(2.1) ¬∃y∀x(G(x) → H(y)) {1b} ✓
(2.2) ¬∃y∀x(H(y) → G(x)) {1b} ✓
(3.1) ∀y¬∀x(G(x) → H(y)) {2.1} ✓
(3.2) ∀y¬∀x(H(y) → G(x)) {2.2} ✓
(4.1) ∀y∃x¬(G(x) → H(y)) {3.1} ✓
(4.2) ∀y∃x¬(H(y) → G(x)) {3.2} ✓
(5.1) ∃x¬(G(x) → H(a)) {4.1} ✓
(5.2) ∃x¬(H(a) → G(x)) {4.2} ✓
(6.1) ¬(G(b) → H(a)) {5.1} ✓
(6.2) ¬(H(a) → G(b)) {5.2} ✓
(7.1a) G(b) {6.1} ✓
(7.1b) ¬H(a) {6.1}
(7.2a) H(a) {6.2}
(7.2b) ¬G(b) {6.2} ✓
(8.1) ∃y(G(b) ↔ H(y)) {1a} ✓
(8.2) ∃y(G(b) ↔ H(y)) {1a} ✓
(9.1) G(b) ↔ H(c) {8.1} ✓
(9.2) G(b) ↔ H(c) {8.2} ✓
(10.1a) G(b) {9.1}
(10.1b) H(c) {9.1} ✓
(10.2a) ¬G(b) {9.1} ✓
(10.2b) ¬H(c) {9.1}
(10.3a) G(b) {9.2} ✓
(10.3b) H(c) {9.2}
(10.4a) ¬G(b) {9.2}
(10.4b) ¬H(c) {9.2} ✓
(11.1) ∃x¬(G(x) → H(c)) {4.1} ✓
(11.2) ⊥ {7.1a, 10.2a}
(11.3) ⊥ {7.2b, 10.3a}
(11.4) ∃x¬(H(c) → G(x)) {4.2} ✓
(12.1) ¬(G(d) → H(c)) {11.1} ✓
(12.4) ¬(H(c) → G(d)) {11.4} ✓
(13.1a) G(d) {12.1}
(13.1b) ¬H(c) {12.1} ✓
(13.4a) H(c) {12.4} ✓
(13.4b) ¬G(d) {12.4}
(14.1) ⊥ {10.1b, 13.1b}
(14.4) ⊥ {10.4b, 13.4a}

Note in particular how (4.1) and (4.2) are re-instantiated at (11.1) and (11.4) to enable the first and fourth branches to close.

7.1 Countermodels

If a tableau does not close, it has at least one open branch. An open branch is either saturated or not. A saturated open branch completely informs a countermodel, while an unsaturated open branch only partially informs a countermodel. A countermodel of a formula A is an interpretation that falsifies A. Remember that an open branch does not constitute an actual countermodel but informs us of an interpretation that would make the counterexample set true. Let us look at an example to see how it works in practice.

Example 7.5. Here is a tableau with open but saturated branches, showing that $\forall x(P(x) \to Q(x), \neg\exists x(R(x) \wedge P(x)) \models \neg\exists x(Q(x) \wedge R(x))$ is invalid.


(1a) $\forall x(P(x) \to Q(x)$ ✓
(1b) $\neg\exists x(R(x) \wedge P(x))$ ✓
(1c) $\neg\neg\exists x(Q(x) \wedge R(x))$ ✓
(2) $\forall x \neg(R(x) \wedge P(x))$ {1b} ✓
(3) $\exists x(Q(x) \wedge R(x))$ {1c} ✓
(4) $Q(a) \wedge R(a)$ {3} ✓
(5a) $Q(a)$ {4}
(5b) $R(a)$ {4} ✓
(6) $\neg(R(a) \wedge P(a))$ {2} ✓
(7.1) $\neg R(a)$ {6} ✓
(7.2) $\neg P(a)$ {6}
(8.1) $\bot$ {5b, 7.1}
(8.2) $P(a) \to Q(a)$ {1a} ✓
(9.1) $\neg P(a)$ {8}
(9.2) $Q(a)$ {8}

Step (2) is the result of moving the negation sign inward, and step (3) is the elimination of the double negation. Since (4) is the only formula having an existential quantifier at the head, we instantiate x with a and remove the quantifier, followed by applying the $\wedge$-rule to obtain the prime formulas $Q(a)$ and $R(a)$ in step (5). Next, we instantiate the variable in (2). The other possibility would have been to instantiate the variable in (1a). In that case, however, we would not have been able to close either of the resulting branches as quickly since neither $\neg P(a)$ nor $Q(a)$ would have resulted in a contradiction. With the current choice, the left branch closes immediately. On the right branch, step (8.2) is got by instantiating (1a)—at this point, we have discharged all quantified formulas as we have no other individual than a to consider. Finally, by applying the rule for $\to$, we obtain (9.1) and (9.2), neither of which closes.

The open branches ending with (9.1) and (9.2) suggest the same countermodel. They involve only the constant a, of which only Q and R are true. To construct a countermodel, let the domain of discourse be $\{a\}$ (or any other constant letter of choice), with $P = \varnothing$, $Q = \{a\}$, and $R = \{a\}$. Then $\forall x(P(x) \to Q(x))$ is vacuously true, $\neg\exists x(R(x) \wedge P(x))$ is true because no individual satisfies both R and P. The consequence, however, is false since a

satisfies both Q and R.

Example 7.6. This example involves a tableau with an unsaturated branch. It shows that $\forall x \exists y P(x, y) \rightarrow \forall y \exists x P(x, y)$ is not valid.

$$(1)\ \neg[\forall x \exists y P(x, y) \rightarrow \forall y \exists x P(x, y)]\ \checkmark$$

$$\text{(2a)}\ \forall x \exists y P(x, y)\ \{1\}\ \checkmark$$
$$\text{(2b)}\ \neg \forall y \exists x P(x, y)\ \{1\}\ \checkmark$$

$$(3)\ \exists y \neg \exists x P(x, y)\ \{\text{2b}\}\ \checkmark$$

$$(4)\ \exists y \forall x \neg P(x, y)\ \{3\}\ \checkmark$$

$$(5)\ \forall x \neg P(x, a)\ \{4\}\ \checkmark$$

$$(6)\ \neg P(a, a)\ \{5\}$$

$$(7)\ \exists y P(a, y)\ \{\text{2a}\}\ \checkmark$$

$$(8)\ P(a, b)\ \{7\}$$

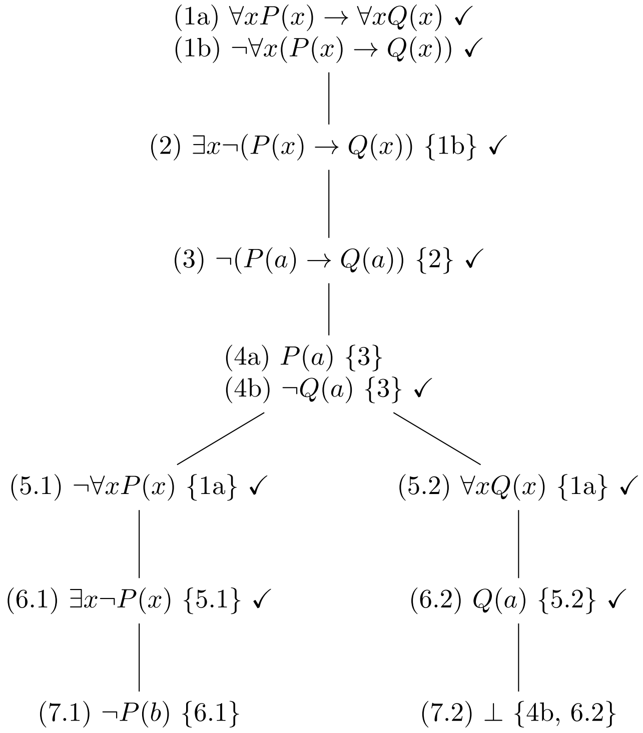
This branch can be infinitely extended without becoming saturated. Thereby it cannot inform a countermodel completely. Nevertheless, it does provide sufficient information for constructing a countermodel. Two things, in particular, are worth noticing. The $\neg P(a, a)$ in (6) tells us that P is not reflexive, i.e., there is at least one individual a such that P is not true of (a, a). Also, the domain of discourse must contain at least two distinct individuals.

One way of constructing a countermodel is to start with an interpretation with the domain of discourse $\{a, b\}$ and $P = \{(a, b)\}$. That will not be enough because to falsify the formula, we need an interpretation that makes the antecedent true but the consequent false. The given interpretation does not make $\forall x \exists y P(x, y)$ true, for when b is the first argument to P, there exists no individual y such that $(b, y) \in P$. The antecedent becomes true, however, if we extend P to also include (b, b). If $P = \{(a, b), (b, b)\}$, then regardless of whether we make a or b the first argument to P, there exists some individual y such that both $P(a, y)$ and $P(b, y)$ are true, and thus $\forall x \exists y P(x, y)$ holds while the consequent $\forall y \exists x P(x, y)$ does not.

While the domain of discourse $\{a, b\}$ together with $P = \{(a, b), (b, b)\}$ indeed is a countermodel, we can still try to attach some meaning to it. Instead of a and b, let $\{0, 1\}$ make up the domain of discourse. Is there a relation R between 0 and 1 such that only $0R1$ and $1R1$?[17] Yes, there is. Let xRy stand for 'x is divisable by y.' Zero is not divisible by itself but divisible by one, and one is only divisible by itself (given the current domain of discourse). Hence, it is true that for every number in $\{0, 1\}$, there is another number in $\{0, 1\}$ such that the first number is divisible by the second, so by setting $P = R$ and the domain of discourse $\{0, 1\}$, the antecedent $\forall x \exists y P(x, y)$ is true. However, since zero is not a divisor of anything, the consequent $\forall y \exists x P(x, y)$ is false.

Yet a third countermodel is given by interpreting P as $<$ (i.e., the ordinary 'less than' relation) and letting the domain of discourse be the natural numbers. Then we see that $\forall x \exists y(x < y)$ is true—for each natural number, there exists a natural number that is even greater—but it is not the case that $\forall y \exists x(x < y)$—for each natural number there is another natural number that is even smaller.

Example 7.7. Show that $\forall x P(x) \to \forall x Q(x) \vdash \forall x(P(x) \to Q(x))$ does not hold and create a countermodel.

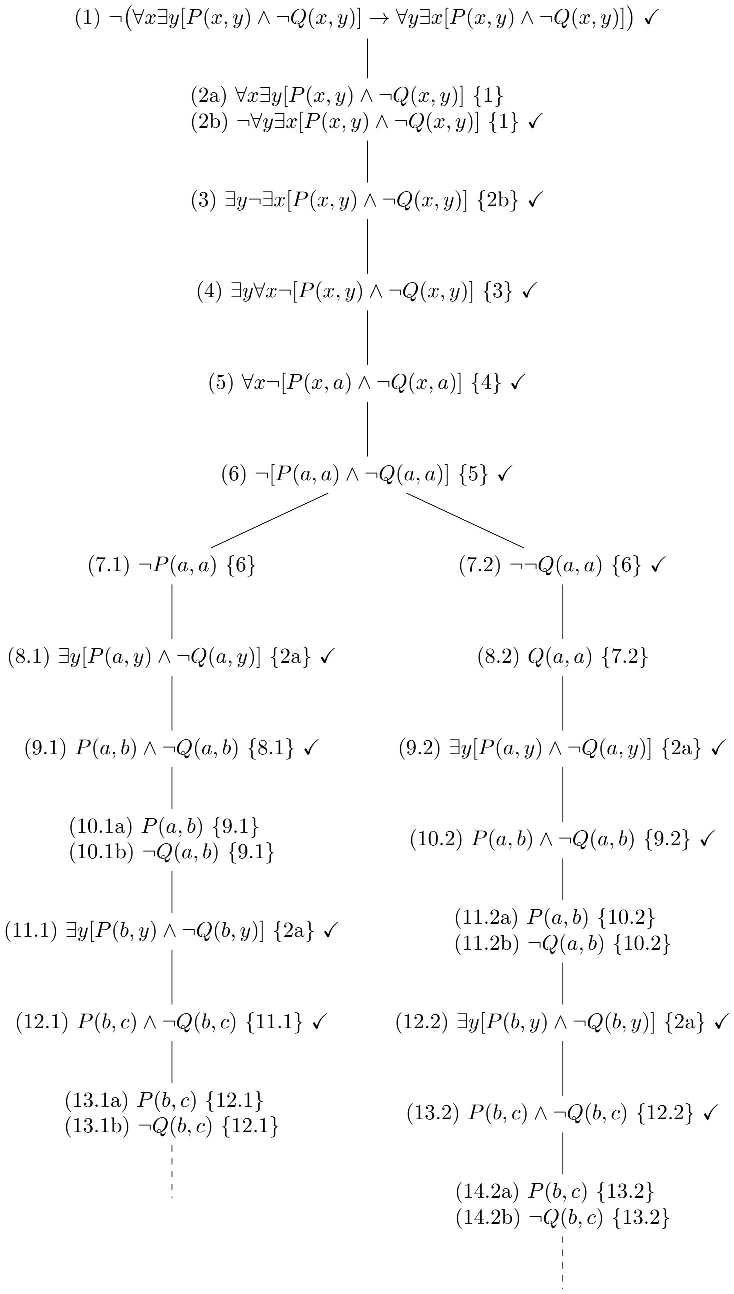

(1a) $\forall x P(x) \to \forall x Q(x)$ ✓
(1b) $\neg\forall x(P(x) \to Q(x))$ ✓
(2) $\exists x \neg(P(x) \to Q(x))$ {1b} ✓
(3) $\neg(P(a) \to Q(a))$ {2} ✓
(4a) $P(a)$ {3}
(4b) $\neg Q(a)$ {3} ✓
(5.1) $\neg\forall x P(x)$ {1a} ✓
(5.2) $\forall x Q(x)$ {1a} ✓
(6.1) $\exists x \neg P(x)$ {5.1} ✓
(6.2) $Q(a)$ {5.2} ✓
(7.1) $\neg P(b)$ {6.1}
(7.2) $\perp$ {4b, 6.2}

The left branch remains open with individual constant letters a and b, and with P true of a. Hence, we can construct a countermodel with the domain of discourse $\{a, b\}$, $P = \{a\}$, and $Q = \varnothing$. Given that countermodel, the left side will be true—since the antecedent $\forall x P(x)$ will be false, the conditional $\forall x P(x) \to \forall x Q(x)$ will be true—but the right side will be false since a is in P but not in Q.

Example 7.8. Show that $\forall x \exists y[P(x, y) \land \neg Q(x, y)] \to \forall y \exists x[P(x, y) \land \neg Q(x, y)]$

[17] These are different ways of writing $R(0, 1)$ and $R(1, 1)$ or $R = \{(0, 1), (1, 1)\}$.

is not valid and provide a countermodel.


(1) $\neg(\forall x\exists y[P(x,y)\wedge\neg Q(x,y)]\rightarrow\forall y\exists x[P(x,y)\wedge\neg Q(x,y)])$ ✓
(2a) $\forall x\exists y[P(x,y)\wedge\neg Q(x,y)]$ {1}
(2b) $\neg\forall y\exists x[P(x,y)\wedge\neg Q(x,y)]$ {1} ✓
(3) $\exists y\neg\exists x[P(x,y)\wedge\neg Q(x,y)]$ {2b} ✓
(4) $\exists y\forall x\neg[P(x,y)\wedge\neg Q(x,y)]$ {3} ✓
(5) $\forall x\neg[P(x,a)\wedge\neg Q(x,a)]$ {4} ✓
(6) $\neg[P(a,a)\wedge\neg Q(a,a)]$ {5} ✓
(7.1) $\neg P(a,a)$ {6}
(8.1) $\exists y[P(a,y)\wedge\neg Q(a,y)]$ {2a} ✓
(9.1) $P(a,b)\wedge\neg Q(a,b)$ {8.1} ✓
(10.1a) $P(a,b)$ {9.1}
(10.1b) $\neg Q(a,b)$ {9.1}
(11.1) $\exists y[P(b,y)\wedge\neg Q(b,y)]$ {2a} ✓
(12.1) $P(b,c)\wedge\neg Q(b,c)$ {11.1} ✓
(13.1a) $P(b,c)$ {12.1}
(13.1b) $\neg Q(b,c)$ {12.1}
(7.2) $\neg\neg Q(a,a)$ {6} ✓
(8.2) $Q(a,a)$ {7.2}
(9.2) $\exists y[P(a,y)\wedge\neg Q(a,y)]$ {2a} ✓
(10.2) $P(a,b)\wedge\neg Q(a,b)$ {9.2} ✓
(11.2a) $P(a,b)$ {10.2}
(11.2b) $\neg Q(a,b)$ {10.2}
(12.2) $\exists y[P(b,y)\wedge\neg Q(b,y)]$ {2a} ✓
(13.2) $P(b,c)\wedge\neg Q(b,c)$ {12.2} ✓
(14.2a) $P(b,c)$ {13.2}
(14.2b) $\neg Q(b,c)$ {13.2}

Neither of the branches close nor do they become saturated, for anytime we have a new instantiation of the universally bound variable x in (2a), we end up with an existentially quantified variable. Once that has been instantiated, we would need yet another instantiation of the universally quantified x in (2a).

A countermodel should make the antecedent $\forall x\exists y[P(x,y)\wedge\neg Q(x,y)]$ true, but the consequent $\forall y\exists x[P(x,y)\wedge\neg Q(x,y)]$ false. Considering the left branch, we have a domain of discourse with $\{a,b,c\}$. We also see that $P=\{(a,b),(b,c)\}$

in (10.1a) and (13.1a). That, however, is not enough to make the antecedent true. For any choice of x, $\neg Q(x, y)$ is true since Q is empty, but $P(x, y)$ cannot be true whenever c is substituted for x. Therefore, let us add (c, c) to P such that $P = \{(a, b), (b, c), (c, c)\}$. With that addition to P, the antecedent is true, but the consequent is false since there is no x such that $P(x, a)$ is true. Hence, one possible countermodel is $P = \{(a, b), (b, c), (c, c)\}$, $Q = \varnothing$, with $\{a, b, c\}$ as the domain of discourse.

Another countermodel is to let the domain of discourse consist of the natural numbers and let P be $\leq$, and Q be $=$. Then the antecedent would read "for every natural number there is a greater number," which is true, while the consequent would be "for any natural number there is a smaller number," which is false. That is a different way of representing the relation $<$ as in Example 7.6.

8 Natural Deduction in Predicate Logic

As for natural deduction in propositional logic, and in contrast to the methods of resolution, and semantic tableaux, we will now try to derive the conclusion from the premises rather than trying to show the counterexample set to be inconsistent.

In addition to the rules for the logical connectives, there are four additional rules concerning the quantifiers.

The rules for eliminating a universal quantifier and introducing an existential quantifier are the most straightforward. Let us start with the elimination of a universal quantifier. If some predicate P is true of all individuals in the domain of discourse, then P will be true specifically of any individual we pick at random from the domain of discourse. In symbols, if $\forall x P(x)$, then $P(a)$ for any constant a. Stated as a rule, we have

$$\text{if } \mathcal{A} \vdash \forall x A \text{ then } \mathcal{A} \vdash A[a/x]$$

for any a free for x in A;[18] we label the rule by ∀E. Note that the set of assumptions $\mathcal{A}$ stays the same after applying the ∀E rule. Here is a short example of its use where $P(a, a)$ is derived from $\forall x \forall y P(x, y)$.

$\{1\}$	(1)	$\forall x \forall y P(x, y)$	Premise
$\{1\}$	(2)	$\forall y P(a, y)$	∀E 1
$\{1\}$	(3)	$P(a, a)$	∀E 2

Of course, we could have derived $P(a, b)$, $P(b, a)$, $P(b, b)$, or any other combination of constants other than a and b.

The introduction rule for the existential quantifier says that if we can show that some formula is true of some specific individual, then we have shown that there exists at least one individual of which the formula in question is true.[19]

[18] To reiterate, if a is free for x in A, it means that we can safely substitute a for x without a being bound to an already existing quantifier. In $\forall x \exists y P(x, y)$, the variable y is not free for x because if we substituted y for x, it would become bound to the existential quantifier. Therefore, keeping variable and constant names separate is a good practice.

[19] Because we will not always know which individual we are referring to precisely, only that it has specific properties or stand in certain relations to other individuals—which essentially defines the type of individual we are referring to—we are informally speaking about a type of

The rule says that

$$\text{if } \mathcal{A} \vdash P(a) \text{ then } \mathcal{A} \vdash \exists x P(x)$$

with x not already in the scope of another quantifier in the scope of $\exists x$. Hence, if we have $\exists x P(a, x)$, we cannot choose x as a new variable and get $\exists x \exists x P(x, x)$. Instead, we must pick another variable, such as y, yielding $\exists y \exists x P(y, x)$, which correctly would let us distinguish between the variables.

As with the ∀E rule, when we apply the ∃I rule, the set of premises does not change. Here is a short example of how to derive $\exists x \exists y P(x, y)$ from $P(a, b)$.

$\{1\}$	(1)	$P(a, b)$	Premise
$\{1\}$	(2)	$\exists y P(a, y)$	∃I 1
$\{1\}$	(3)	$\exists x \exists y P(x, y)$	∃I 2

We can introduce a universal quantifier once we have shown that something holds for an arbitrary individual that is not part of the set of assumptions—we can only obtain an individual that is not part of the assumptions by applying the universal quantifier elimination rule. Thus, we cannot go from a set of assumptions $P(a)$ to $\forall x P(x)$ since a is in the assumption $P(a)$. We could, however, go from the assumption $\forall x P(x)$ to $P(a)$ and then to $\forall x P(x)$ because $P(a)$ is, in this case, not part of the assumptions. In list form, we write the assumptions used to derive the formula on the current line at the start of that line within curly brackets.

$\{1\}$	(1)	$\forall x P(x)$	Premise
$\{1\}$	(2)	$P(a)$	∀E 1
$\{1\}$	(3)	$\forall x P(x)$	∀I 2

The rule says that if we can obtain $P(a)$ from a set of premises where a does not occur, we can conclude that $\forall x P(x)$ can be obtained from that same set of assumptions. In symbols, we have

$$\text{if } \mathcal{A} \vdash P(a) \text{ then } \mathcal{A} \vdash \forall x P(x)$$

given that a is not present among the premises in $\mathcal{A}$.

A formalization of the argument "All coders are humans. All humans need water. Hence, all coders need water" can be formalized as

$\{1\}$	(1)	$\forall x[C(x) \to H(x)]$	Premise
$\{2\}$	(2)	$\forall x[(H(x) \to W(x)]$	Premise
$\{3\}$	(3)	$C(a)$	Assumption
$\{1\}$	(4)	$C(a) \to H(a)$	∀E 1
$\{1, 3\}$	(5)	$H(a)$	→E 3, 4
$\{2\}$	(6)	$H(a) \to W(a)$	∀E 2
$\{1, 2, 3\}$	(7)	$W(a)$	→E 5, 6
$\{1, 2\}$	(8)	$C(a) \to W(a)$	→I 3-7
$\{1, 2\}$	(9)	$\forall x[C(x) \to W(x)]$	∀I 8

individual rather than a specific individual. Thus when we write $P(a)$, we usually see a as a constant reference to one of the individuals of which we assume P to be true without knowing exactly which one.

In the derivation, we assumed that C was true of some individual a. It is important to note that $C(a)$ is an assumption, not part of the premises. On line 8, we discharge that assumption by introducing the conditional. We cannot say that C is true of any particular individual from the premises alone. Still, whenever—if ever—C is true of a specific individual, then W will be true of that same individual as well. Hence, from line 8, we can derive line 9, using the $\forall$I rule, as a is not in any of the premises on which line 8 rests—only lines 1 and 2 are in the set of assumptions of line 8, and a is not present anywhere in either 1 or 2.

The fourth and last quantifier rule is that of eliminating an existential quantifier. It is a bit more involved than the previous rules and reads

$$\text{if } \mathcal{A} \vdash \exists x P(x) \text{ and } P(a), \mathcal{B} \vdash A \text{ then } \mathcal{A}, \mathcal{B} \vdash A$$

given that a is not present in $\mathcal{A}$, $\exists x P(x)$, $\mathcal{B}$, or A. The assumption $P(a)$ is discharged upon application of the rule.

The of premises $\mathcal{B}$ may be empty, and since any formula can be derived from itself, a minimal example with the premise set $\mathcal{B}$ empty is

$\{1\}$	(1)	$\exists x P(x)$	Premise (this corresponds to $\mathcal{A}$ in the rule)
$\{2\}$	(2)	$P(a)$	Assumption (to be discharged below)
$\{2\}$	(3)	$\exists y P(y)$	$\exists$I 2 (using y only to make it distinct from line 1)
$\{1\}$	(4)	$\exists y P(y)$	$\exists$E 1, 2-3 (the assumption on line 2 is discharged)

A derivation using the $\exists$E rule will typically result in an existentially quantified formula, but as the following derivation shows, there are exceptions.

$\{1\}$	(1)	$\forall x[P(x) \rightarrow Q]$	Premise
$\{2\}$	(2)	$\exists x P(x)$	Premise
$\{3\}$	(3)	$P(a)$	Assumption
$\{1\}$	(4)	$P(a) \rightarrow Q$	$\forall$E 1
$\{1,3\}$	(5)	Q	$\rightarrow$E 3, 4
$\{1,2\}$	(6)	Q	$\exists$E 2, 3-5

Even though lines 5 and 6 both contain Q, they are derived from different sets of premises. Without the application of the $\exists$E rule, the truth of Q would have been justified by $\forall x[P(x) \rightarrow Q]$ and $P(a)$, rather than the actual premises $\forall x[P(x) \rightarrow Q]$ and $\exists x P(x)$.

In addition to the fundamental quantifier rules, we derive the rules corresponding to the equivalences $\neg\forall x P(x) \Leftrightarrow \exists x \neg P(x)$ and $\neg\exists x P(x) \Leftrightarrow \forall x \neg P(x)$ and make them part of the rule set.

$\{1\}$	(1)	$\neg\forall x P(x)$	Premise
$\{2\}$	(2)	$\neg\exists x \neg P(x)$	Assumption
$\{3\}$	(3)	$\neg P(a)$	Assumption
$\{3\}$	(4)	$\exists x \neg P(x)$	$\exists$I 3
$\{2,3\}$	(5)	$\bot$	$\neg$E 2, 4
$\{2\}$	(6)	$P(a)$	RAA' 3-5
$\{2\}$	(7)	$\forall x P(x)$	$\forall$I 6
$\{1,2\}$	(8)	$\bot$	$\neg$E 1, 7
$\{1\}$	(9)	$\exists x \neg P(x)$	RAA' 2-8

$\{1\}$	(1)	$\exists x \neg P(x)$	Premise
$\{2\}$	(2)	$\forall x P(x)$	Assumption
$\{3\}$	(3)	$\neg P(a)$	Assumption
$\{2\}$	(4)	$P(a)$	$\forall$E 2
$\{2,3\}$	(5)	$\bot$	$\neg$E 3, 4
$\{1,2\}$	(6)	$\bot$	$\exists$E 1, 3-5
$\{1\}$	(7)	$\neg\forall x P(x)$	RAA 2-6

$\{1\}$	(1)	$\neg\exists x P(x)$	Premise
$\{2\}$	(2)	$P(a)$	Assumption
$\{2\}$	(3)	$\exists x P(x)$	$\exists$I 2
$\{1,2\}$	(4)	$\bot$	$\neg$E 1, 3
$\{1\}$	(5)	$\neg P(a)$	RAA 2-4
$\{1\}$	(6)	$\forall x \neg P(x)$	$\forall$I 5

$\{1\}$	(1)	$\forall x \neg P(x)$	Premise
$\{2\}$	(2)	$\exists x P(x)$	Assumption
$\{3\}$	(3)	$P(a)$	Assumption
$\{1\}$	(4)	$\neg P(a)$	$\forall$E 1
$\{1,3\}$	(5)	$\bot$	$\neg$E 3, 4
$\{1,2\}$	(6)	$\bot$	$\exists$E 2, 3-5
$\{1\}$	(7)	$\neg\exists x P(x)$	RAA 2-5

Here is the complete set of rules for natural deduction in predicate logic:

(Axiom) $A \vdash A$

($\wedge$I) If $\mathcal{A} \vdash A$ and $\mathcal{B} \vdash B$ then $\mathcal{A}, \mathcal{B} \vdash A \wedge B$

($\wedge$E) If $\mathcal{A} \vdash A \wedge B$ then $\mathcal{A} \vdash A$ and $\mathcal{A} \vdash B$

($\vee$I) If $\mathcal{A} \vdash A$ or $\mathcal{A} \vdash B$ then $\mathcal{A} \vdash A \vee B$

($\vee$E)* If $\mathcal{A} \vdash A \vee B$ and $\mathcal{B}, A \vdash C$ and $\mathcal{C}, B \vdash C$ then $\mathcal{A}, \mathcal{B}, \mathcal{C} \vdash C$

($\rightarrow$I)* If $\mathcal{A}, A \vdash B$ then $\mathcal{A} \vdash A \rightarrow B$

($\rightarrow$E) If $\mathcal{A} \vdash A$ and $\mathcal{B} \vdash A \rightarrow B$ then $\mathcal{A}, \mathcal{B} \vdash B$

($\leftrightarrow$I) If $\mathcal{A} \vdash A \rightarrow B$ and $\mathcal{B} \vdash B \rightarrow A$ then $\mathcal{A}, \mathcal{B} \vdash A \leftrightarrow B$

($\leftrightarrow$E) If $\mathcal{A} \vdash A \leftrightarrow B$ then $\mathcal{A} \vdash A \rightarrow B$ or $\mathcal{A} \vdash B \rightarrow A$

($\neg\neg$E) If $\mathcal{A} \vdash \neg\neg A$ then $\mathcal{A} \vdash A$

($\neg$E) If $\mathcal{A} \vdash \neg A$ and $\mathcal{B} \vdash A$ then $\mathcal{A}, \mathcal{B} \vdash \bot$

(EFQ) If $\mathcal{A}, \mathcal{B} \vdash \bot$ then $\mathcal{A}, \mathcal{B} \vdash C$

(TND)* If $\mathcal{A}, A \vdash B$ and $\mathcal{B}, \neg A \vdash B$ then $\mathcal{A}, \mathcal{B} \vdash B$

(RAA)* If $\mathcal{A}, A \vdash \bot$ then $\mathcal{A} \vdash \neg A$

(RAA')* If $\mathcal{A}, \neg A \vdash \bot$ then $\mathcal{A} \vdash A$

($\forall$E) If $\mathcal{A} \vdash \forall x A$ then $\mathcal{A} \vdash A[a/x]$ for any a free for x in A

(∀I) If $\mathcal{A} \vdash P(a)$ then $\mathcal{A} \vdash \forall x P(x)$ given that a is not present among the premises in $\mathcal{A}$

(∃I) If $\mathcal{A} \vdash P(a)$ then $\mathcal{A} \vdash \exists x P(x)$ with x not already in the scope of another quantifier in the scope of $\exists x$

(∃E)* If $\mathcal{A} \vdash \exists x P(x)$ and $P(a), \mathcal{B} \vdash A$ then $\mathcal{A}, \mathcal{B} \vdash A$ given that a is not present in $\mathcal{A}$, $\exists x P(x)$, $\mathcal{B}$, or A

(¬∀E) If $\mathcal{A} \vdash \neg\forall x P(x)$ then $\mathcal{A} \vdash \exists x \neg P(x)$

(¬∀I) If $\mathcal{A} \vdash \exists x \neg P(x)$ then $\mathcal{A} \vdash \neg\forall x P(x)$

(¬∃E) If $\mathcal{A} \vdash \neg\exists x P(x)$ then $\mathcal{A} \vdash \forall x \neg P(x)$

(¬∃I) If $\mathcal{A} \vdash \forall x \neg P(x)$ then $\mathcal{A} \vdash \neg\exists x P(x)$

*These rules are so-called *discharge rules* because they discharge the assumptions that were made—some authors refer to discharging assumptions as "closing" assumptions. In ∨E, the assumptions A and B are discharged. In →I, the assumption A is discharged but not forgotten as it becomes the antecedent of a conditional instead of being a premise. In TND, the assumptions A and $\neg A$ are removed because the truth value of B is not affected by the truth value of A. In the RAA rule, if the assumption A yields a contradiction, we can derive its negation and scrap the assumption A since it was only used for testing—the RAA' rule works similarly.

8.1 Discharging assumptions

The premises, along with the conclusion, are what constitutes an actual argument. That the conclusion follows from the premises can be far from apparent. In other cases, as in an investigation, one might need to work out what follows from a given set of facts and show that that indeed is the case. To convince someone about the validity of an argument—including oneself sometimes—one may need to involve additional assumptions to support the reasoning process, as we will soon see.

We reason based on assumptions in everyday life, not only in formal settings. Imagine if I had a spaceship, then I would travel to Alderaan. In that case, the assumptions are that I have a spaceship and Alderaan exists. Of course, owning a spaceship and Alderaan's existence does not become a reality just because I assume so. Similarly, the assumptions in a formal derivation do not become part of the supposedly true premises just because they are introduced and temporarily used to prove a point formally.

Let us consider an example involving a suspect seen by several witnesses shortly after a burglary had taken place. At least one of the witnesses has made a correct observation, but the police do not know who. According to the witness interviews, the suspect was running or riding an electric scooter from the scene. Nearby, there is a tunnel, and evidence from a surveillance video shows only cars went thru the tunnel. After some thinking, the investigator concluded the burglar did not escape thru the tunnel. Based on what we know so far, can we prove that using predicate logic and natural deduction?

The following is what we know:

- At least one of the witness observations is correct.
- Any observation either has it that the burglar was running och riding an electric scooter from the scene
- Only cars went thru the tunnel
- Running or riding an electric scooter is not riding a car
- Only cars went thru the tunnel and the suspect went thru the tunnel only if the suspect was observed driving a car

To formalize the facts and the conclusion we define the following predicates to be used in the derivation:

- $O(x)$: 'x is a correct witness observation'
- $R(x)$: 'x is an observation that the suspect was running'
- $S(x)$: 'x is an observation that the suspect was riding an electrical scooter'
- $C(x)$: 'x is an observation that the suspect was riding a car'
- T: 'Only cars went thru the tunnel'
- U: 'The suspect went thru the tunnel'

The facts that make up the premises are formalized as follows:

- That at least one of the witness observations are correct is written as $\exists x O(x)$
- That a correct observation involves the burglar either running or riding an electric scooter becomes $\forall x\big(O(x) \to R(x) \vee S(x)\big)$
- That only cars went thru the tunnel is T
- That running or riding an electric scooter is not riding a car is split into two separate premises $\forall x\big(R(x) \to \neg C(x)\big)$ and $\forall x\big(S(x) \to \neg C(x)\big)$
- That only cars went thru the tunnel and the suspect went thru the tunnel only if the suspect was observed driving a car as written as $T \wedge U \to \forall x C(x)$

Based on the above premises, here is on possible way of deriving the conclusion $\neg U$:

$\{1\}$	(1)	$\forall x\big(O(x) \to R(x) \lor S(x)\big)$	Premise
$\{2\}$	(2)	$\forall x\big(R(x) \to \neg C(x)\big)$	Premise
$\{3\}$	(3)	$\forall x\big(S(x) \to \neg C(x)\big)$	Premise
$\{4\}$	(4)	$\exists x O(x)$	Premise
$\{5\}$	(5)	T	Premise
$\{6\}$	(6)	$T \land U \to \forall x C(x)$	Premise
$\{7\}$	(7)	$O(a)$	Assumption (to be discharged by ∃E)
$\{1\}$	(8)	$O(a) \to R(a) \lor S(a)$	∀E 1
$\{1, 7\}$	(9)	$R(a) \lor S(a)$	→E 7, 8
$\{10\}$	(10)	$R(a)$	Assumption (to be discharged by ∨E)
$\{2\}$	(11)	$R(a) \to \neg C(a)$	∀E 2
$\{2, 10\}$	(12)	$\neg C(a)$	→E 10, 11
$\{13\}$	(13)	$S(a)$	Assumption (to be discharged by ∨E)
$\{3\}$	(14)	$S(a) \to \neg C(a)$	∀E 3
$\{3, 13\}$	(15)	$\neg C(a)$	→E 13, 14
$\{1, 2, 3, 7\}$	(16)	$\neg C(a)$	∨E 9, 10-12, 13-15 (discharges 10 and 13)
$\{17\}$	(17)	U	Assumption (to be discharged by RAA)
$\{5, 17\}$	(18)	$T \land U$	∧I 6, 17
$\{5, 6, 17\}$	(19)	$\forall x C(x)$	→E 6, 18
$\{5, 6, 17\}$	(20)	$C(a)$	∀E 19
$\{1, 2, 3, 5, 6, 7, 17\}$	(21)	$\bot$	¬E 16, 20
$\{1, 2, 3, 5, 6, 7\}$	(22)	$\neg U$	RAA 17-21 (discharges 17)
$\{1, 2, 3, 4, 5, 6\}$	(23)	$\neg U$	∃E 7, 8-22 (discharges 7)

In this example, we used three assumptions to support us in reaching the sought conclusion. First, we assumed that a was a specific correct observation on line 7. Second, we reasoned about the two cases where a involved the suspect running or riding an electric scooter, starting with the assumptions on lines 10 and 13. Third, we tested a hypothesis that the suspect went thru the tunnel by assuming that the suspect indeed did go thru the tunnel on line 17, eventually arriving at a contradiction. None of these assumptions, however, are part of the premises. We cannot tell whether a is a correct witness observation, but based on that assumption, we derived the conclusion—we could have picked any other individual other than a and still reached the same conclusion.

The assumption that a was such a correct witness observation stuck until we reached $\neg U$ on line 23. However, since the derivation of $\neg U$ did not depend specifically on the correct witness observation a—it could have been b or c or any other correct witness observation there might have been—we could discharge the assumption $O(a)$.

At various points in the derivation, we discharged each of the *assumptions*. In other words, the conclusion follows from the premises only, but we used some temporary assumptions to support the actual derivation.

When to include an assumption and what to assume is almost a strategic choice. It is almost necessary to form a general idea of the complete derivation so that one does not start assuming things at random. Of course, sometimes one might have to try an assumption and see where it may lead, but in general, adding an assumption should be based on an idea of what to use it for and, thereby, how it eventually will be discharged. An undischarged assumption will necessarily become part of the premise set, turning the derivation into an argument different from what you had initially.

8.1.1 The rules that discharge assumptions

Suppose we derive a formula from the original set of premises in conjunction with an additional assumption. In that case, the result of the derivation presupposes that that assumption indeed is true; the derivation is contingent on the truth of the assumption, just as if it was a premise. In fact, we can look at assumptions as additional but temporary premises. Under certain conditions, those assumptions can be discharged (i.e., dismissed as temporary premises)—it is the discharging rules that set those conditions.

Unless we somehow discharge an assumption before or at the point of getting to the conclusion[20] in a derivation, it will become part of a newly formed argument with an extended premise set—that is generally not what we want. Trying to show that a conclusion B follows from a set of premises Γ by introducing an assumption A, without at some point discharging it, shows that $\Gamma, A \vdash B$, not that $\Gamma \vdash B$ holds. If we were allowed to let an assumption remain active throughout a derivation, with no rule to discharge it, and then acted as if it magically disappeared on the last line of the derivation, we could easily derive anything from a set of premises—extend the set of premises by assuming the conclusion itself, and we are done. Of course, a logically sound deduction system does not allow for that.

Assumptions play different roles depending on the rules meant to discharge them, but introducing an assumption should always be done with a purpose. For example, one can introduce an assumption intending to reach a contradiction that would prove its negation. Sometimes one wants to show that something else follows from an assumption, hence introducing an assumption that subsequently is discharged by making it the antecedent of a conditional. When dealing with existentially quantified variables, we let an assumption exemplify an individual of which the formula in question would be true—a bit like using one's imagination when reasoning in daily life. When investigating what a disjunction may lead to, we assume each disjunct to be true separately and reason based on those cases. Once we have arrived at a conclusion, no assumption should play an active role—only the premises remain as the fundamental grounds for the argument's conclusion.

Any assumption that has not been discharged is an *open* or *undischarged* assumption. A discharged assumption is also called a *closed* assumption. In what follows, the term premise will be used to denote any original premise as well as any previously introduced assumption.

Let us look at each rule that discharges assumptions to form a better intuitive understanding of their almost strategic role in derivations.

$\rightarrow$I rule. In short, when setting out to derive a conclusion with the form of a conditional, start by assuming the antecedent A of the conditional and attempt to derive the consequent B. Here, A and B are arbitrary formulas. Once you have derived the consequent, you have a derivation from the premises, denoted $\mathcal{A}$, plus the assumption A to B, i.e., $\mathcal{A}, A \vdash B$. At that point, however, we can discharge the assumption A from the premises by introducing a conditional formula and obtain $\mathcal{A} \vdash A \rightarrow B$.

[20] Note that when we refer to conclusions, it not only the main conclusion of the derivation but also intermediate conclusions drawn along the way.

A concise example shows how we can introduce an assumption and reason based on it as if it was a premise before discharging it. Line 3 is just a repetition of the sole premise. In this case, the rule says that if we can derive A from A and B, then we can derive $B \to A$ from A alone.

{1}	(1)	A	Premise
{2}	(2)	B	Assumption (to be discharged by →I)
{1}	(3)	A	Axiom 1
{1}	(4)	$B \to A$	→I 2, 3 (discharges 2)

The discharge of the assumption takes place on line 4, where the number 1 within curly brackets indicates that the only premise needed to conclude line 4 is line 1. However, without introducing B as an assumption, we would not be able to derive $B \to A$; hence, while it is true that $A \models B \to A$, with only A in the set of premises, assuming B was a necessary component of the derivation.

∨E rule. In some cases, the term *elimination* rule may be confusing, particularly when referring to it in a context where an assumption is discharged as a result of applying the rule. Disjunctions and existentially quantified formulas naturally introduce uncertainty: we know something is true, but not precisely which. If $A \vee B$, we know for sure that either of the disjuncts is true, or both are true. Unless there is a conditional $A \vee B \to C$, i.e., a formula with the corresponding disjunction as antecedent, it may be difficult to know what to do with the information given by $A \vee B$.

In natural deduction, we look at two possible cases when encountering a disjunction $A \vee B$: we investigate what can be derived from the premises under the assumption that A is true, and we investigate what can be derived from the premises under the assumption that B is true. If we find that we can derive the same formula C by assuming A and B separately, then we can practically substitute C for $A \vee B$ in the remainder of the derivation. In any case, the disjunction $A \vee B$ remains a premise—we might find a use for $A \vee B$ elsewhere in the derivation. We have *eliminated* the disjunction $A \vee B$ in the sense that we now know C is derivable from it regardless of which of the disjuncts is true.

A rather practical but trivial example could be a situation where you have an orange or an apple in a closed bag; you do not know which one you have or if you have both. However, you do know that if you have an orange, you can make fruit juice, and if you have an apple, you can make fruit juice. By showing that you can make fruit juice regardless of whether there is an orange or an apple in the bag, it does not matter what exactly makes the disjunction true—if it is the orange, the apple, or both. You could make use of that disjunction in deriving the conclusion that you can make fruit juice.

What you eliminate, however, is the assumption that you have an orange and the assumption that you have an apple. These are purely there for the sake of reasoning about the different possible cases. As long as both eventually yield fruit juice, their role is no different from that original disjunction; they don't provide any additional information and can therefore be eliminated.

$\{1\}$	(1)	$A \to F$	Premise
$\{1\}$	(2)	$O \to F$	Premise
$\{3\}$	(3)	$A \lor O$	Premise
$\{4\}$	(4)	A	Assumption (to be discharged by $\lor$E)
$\{1, 4\}$	(5)	F	$\to$E
$\{6\}$	(6)	O	Assumption (to be discharged by $\lor$E)
$\{2, 6\}$	(7)	F	$\to$E
$\{1, 2, 3\}$	(8)	F	$\lor$E 3, 4-5, 6-7 (discharges 4 and 6)

In most derivations, as we have seen, it is not as simple as going from orange to fruit juice or from apple to fruit juice, but the fundamental idea is the same. Also, the assumptions that we discharge are the ones representing the two cases suggested by the disjunction. Once we have derived the same formula from each of them, they can be discharged by applying the $\lor$E rule.

$\exists$E rule. Similar to the $\lor$E rule, this rule is more about putting the existentially quantified formula to good use rather than eliminating it. We use it by instantiating the existentially bound variable in part of the derivation. If we later derive something based on that temporary instantiation, then we can use that toward reaching a conclusion, as long as we do not depend on the specific instantiation. After all, reasoning based on ”Assuming this is a cat, then...” on the one hand and ”This is a cat, so...” on the other, are entirely different situations—at least in this context.

The $\exists$E rule is somewhat similar to the $\lor$E rule in that we can make use of knowing that something is true without knowing exactly which. For the $\lor$E rule, we knew that at least one of the disjuncts is true, and in this case, we know that some predicate P holds for at least one individual—for a domain of discourse with individuals a, b, and c, we know that $P(a) \lor P(b) \lor P(c)$, given the premise $\exists x P(x)$.

Suppose you receive a box. In it are four smaller boxes with unknown content. Then you are told the (big) box contains at least one cake. You pick up one of the smaller boxes and think, ”If this was a box containing a cake, what could I do?” Holding and getting a feel of the box helps your imagination. ”I know! I can throw a cake party and invite the neighbors.” The conclusion that you can throw a cake party is based on premises unknown to the rest of the world—let us not be concerned about those to keep things simple. The point is you imaginatively and temporarily instantiated that one box to think of something to do. Whether the little box in your hand contains a cake is irrelevant as long as some box contains a cake. Making the rest of the underlying premises explicit would, of course, be important in a logical derivation.

The following example is short but shows a typical situation in which the penultimate line is the same as the last one, apart from them being derived using different rules.

$\{1\}$	(1)	$\forall x \big(A(x) \to B\big)$	Premise
$\{2\}$	(2)	$\exists x A(x)$	Premise
$\{3\}$	(3)	$A(c)$	Assumption (to be discharged by $\exists$E)
$\{1\}$	(4)	$A(c) \to B$	$\forall$E 1
$\{1, 3\}$	(5)	B	$\to$E 3, 4
$\{1, 2\}$	(6)	B	$\exists$E 2, 4-5 (discharges 3)

RAA rule and RAA'. These rules work in precisely the same way, the only difference being that the first presupposes a positive assumption and the latter a negative assumption. If we, from a set of consistent premises and an assumption A, derive $\bot$, we can conclude $\neg A$—vice versa for an assumption $\neg A$. In the following, we will only mention RAA—flip the negation, and you have RAA'.

If an assumption turns out to contradict the already present premises, we can apply RAA to discharge that assumption, indirectly having shown that its negation is consistent with the premises. It is a handy rule when trying to prove the truth-value of a certain formula—you might have a sense that a formula has to be true, e.g., that the conclusion you are trying to derive is contingent on the truth of that formula. When needing to show that some formula is true, assume its negation, and derive a contradiction; that is one way of doing it.

$\{1\}$	(1)	$A \to B$	Premise
$\{2\}$	(2)	$\neg B$	Premise
$\{3\}$	(3)	A	Assumption (to be discharged by RAA)
$\{1,3\}$	(4)	B	→E 1, 3
$\{1,2,3\}$	(5)	$\bot$	¬E 2, 4
$\{1,2\}$	(6)	$\neg A$	RAA 3-5 (discharges 3)

TND rule. For all formulas A, either A is true, or its negation $\neg A$ is true. The TND rule is quite similar in nature to the ∨E rule, but since it is based on the fact that $A \vee \neg A$ is true for any A, we can base the derivation on the true assumption $A \vee \neg A$ rather than on some already present premise $A \vee B$.

Having shown that the assumptions A and $\neg A$ yield the same result, A and $\neg A$ can safely be discharged using the TND rule.

Sometimes, when some of the premises are conditional formulas such as $A \to B$, one might have to test what follows from A as well as from $\neg A$. If the same formula can be derived from both A and $\neg A$, the conclusion need not be contingent on the truth value of the antecedent.

$\{1\}$	(1)	$A \to B$	Premise
$\{2\}$	(2)	$\neg A \to B$	Premise
$\{3\}$	(3)	A	Assumption (to be discharged by TND)
$\{1,3\}$	(4)	B	→E 1, 3
$\{5\}$	(5)	$\neg A$	Assumption (to be discharged by TND)
$\{2,5\}$	(6)	B	→E 2, 5)
$\{1,2\}$	(6)	B	TND 3-4, 5-6 (discharges 3 and 5)

8.2 Examples of derivations

Example 8.1. Show that $\forall x\big(P(x) \to Q(x)\big), \forall x P(x) \models \forall x Q(x)$ using natural deduction.

$\{1\}$	(1)	$\forall x\big(P(x) \to Q(x)\big)$	Premise
$\{2\}$	(2)	$\forall x P(x)$	Premise
$\{1\}$	(3)	$P(a) \to Q(a)$	$\forall$E 1
$\{2\}$	(4)	$P(a)$	$\forall$E 2
$\{1, 2\}$	(5)	$Q(a)$	$\to$E 3, 4
$\{1, 2\}$	(6)	$\forall x Q(x)$	$\forall$I 5

Example 8.2. Show that $\forall x\big(P(x) \to Q(x)\big), \exists x P(x) \models \exists x Q(x)$ using natural deduction.

$\{1\}$	(1)	$\forall x\big(P(x) \to Q(x)\big)$	Premise
$\{2\}$	(2)	$\exists x P(x)$	Premise
$\{3\}$	(3)	$P(a)$	Assumption (to be discharged by $\exists$E)
$\{1\}$	(4)	$P(a) \to Q(a)$	$\forall$E 1
$\{1, 3\}$	(5)	$Q(a)$	$\to$E 3, 4
$\{1, 3\}$	(6)	$\exists x Q(x)$	$\exists$I 5
$\{1, 2\}$	(7)	$\exists x Q(x)$	$\exists$E 2, 3-6 (discharges 3)

The order in which we apply the rules is typically important for obtaining a structure that better corresponds to how we reason. However, as long as one follows the rules correctly, they can be written in any order. While the order makes a derivation more or less readable, it is not the order of the lines that dictate the correctness of a derivation. A different way of writing the previous derivation without affecting the readability much follows here.

$\{1\}$	(1)	$\forall x\big(P(x) \to Q(x)\big)$	Premise
$\{2\}$	(2)	$\exists x P(x)$	Premise
$\{1\}$	(3)	$P(a) \to Q(a)$	$\forall$E 1
$\{4\}$	(4)	$P(a)$	Assumption (to be discharged by $\exists$E)
$\{1, 4\}$	(5)	$Q(a)$	$\to$E 3, 4
$\{1, 4\}$	(6)	$\exists x Q(x)$	$\exists$I 5
$\{1, 2\}$	(7)	$\exists x Q(x)$	$\exists$E 2, 4-6 (discharges 4)

By sticking to the order of the first listing, the elimination of the universal quantifier (on line 4 in the first listing and on line 3 in the second listing) comes after the assumption of $P(a)$ to be discharged by the $\exists$E rule. The reason for eliminating the universal quantifier is to make that conditional available such that we can derive $Q(a)$. So, by eliminating the universal quantifier only after the assumption of $P(a)$ has been made, the listing better conveys the reason for applying that $\forall$E rule at all.

Example 8.3. Multiple premises does not necessarily mean a very long derivation. This example has a constant c, in place of a variable, in one of the premises and in the conclusion. To make it easier to read, predicates are written without parentheses around their arguments. Hence, Sxy is the same as $S(x, y)$.

$$\begin{aligned}&\{\exists x\big(Px \land \forall y[(Qy \land Rxy) \to Sxy]\big),\\ &\forall x\big(Px \to Tx\big),\\ &Qe \land \forall x\big(Tx \to Rxe\big)\}\\ &\ \models \exists x\big(Px \land \exists y(Qy \land Sxy)\big)\end{aligned}$$

$\{1\}$	(1)	$\exists x\big(Px \land \forall y[(Qy \land Rxy) \to Sxy]\big)$	Premise
$\{2\}$	(2)	$\forall x\big(Px \to Tx\big)$	Premise
$\{3\}$	(3)	$Qe \land \forall x\big(Tx \to Rxe\big)$	Premise
$\{4\}$	(4)	$Pa \land \forall y[(Qy \land Ray) \to Say]$	Assumption (to be discharged by ∃E)
$\{4\}$	(5)	$\forall y[(Qy \land Ray) \to Say]$	∧E 4
$\{4\}$	(6)	$(Qe \land Rae) \to Sae$	∀E 5
$\{4\}$	(7)	Pa	∧E 4
$\{2\}$	(8)	$Pa \to Ta$	∀E 2
$\{2, 4\}$	(9)	Ta	→E 7, 8
$\{3\}$	(10)	Qe	∧E 3
$\{3\}$	(11)	$\forall x\big(Tx \to Rxe\big)$	∧E 3
$\{3\}$	(12)	$Ta \to Rae$	∀E 11
$\{2, 3, 4\}$	(13)	Rae	→E 9, 12
$\{2, 3, 4\}$	(14)	$Qe \land Rae$	∧I 10, 13
$\{2, 3, 4\}$	(15)	Sae	→E 6,14
$\{2, 3, 4\}$	(16)	$Qe \land Sae$	∧I 10, 15
$\{2, 3, 4\}$	(17)	$\exists y(Qy \land Say)$	∃I 16
$\{2, 3, 4\}$	(18)	$Pa \land \exists y(Qy \land Say)$	∧I 7, 17
$\{2, 3, 4\}$	(19)	$\exists x\big(Px \land \exists y(Qy \land Sxy)\big)$	∃I 18
$\{1, 2, 3\}$	(20)	$\exists x\big(Px \land \exists y(Qy \land Sxy)\big)$	∃E 1, 4-19 (discharges 4)

Example 8.4.

$$\exists x P(x, x), \exists x \forall z Q(x, z) \models \exists x \exists y\big(P(x, x) \land Q(y, x)\big)$$

$\{1\}$	(1)	$\exists x P(x, x)$	Premise
$\{2\}$	(2)	$\exists x \forall z Q(x, z)$	Premise
$\{3\}$	(3)	$P(a, a)$	Assumption (to be discharged by ∃E)
$\{4\}$	(4)	$\forall z Q(b, z)$	Assumption (to be discharged by ∃E)
$\{4\}$	(5)	$Q(b, a)$	∀E 4
$\{3, 4\}$	(6)	$P(a, a) \land Q(b, a)$	∧I 3, 5
$\{3, 4\}$	(7)	$\exists y\big(P(a, a) \land Q(y, a)\big)$	∃I 6
$\{2\}$	(8)	$\exists y\big(P(a, a) \land Q(y, a)\big)$	∃E 2, 4-7 (discharges 4)
$\{2\}$	(9)	$\exists x \exists y\big(P(x, x) \land Q(y, x)\big)$	∃I 8
$\{1, 2\}$	(10)	$\exists x \exists y\big(P(x, x) \land Q(y, x)\big)$	∃E 1, 3-9 (discharges 3)

In the next set of examples are derivations of some important predicate logic equivalences.

Example 8.5.

$$\exists x(A(x) \land B) \Leftrightarrow \exists x A(x) \land B$$

$\{1\}$	(1)	$\exists x(A(x) \wedge B)$	Premise
$\{2\}$	(2)	$A(a) \wedge B$	Assumption
$\{2\}$	(3)	$A(a)$	$\wedge$E 2
$\{2\}$	(4)	$\exists x A(x)$	$\exists$I 3
$\{2\}$	(5)	B	$\wedge$E 2
$\{2\}$	(6)	$\exists x A(x) \wedge B$	$\wedge$I 4, 5
$\{1\}$	(7)	$\exists x A(x) \wedge B$	$\exists$E 1, 2-6

$\{1\}$	(1)	$\exists x A(x) \wedge B$	Premise
$\{1\}$	(2)	$\exists x A(x)$	$\wedge$E 1
$\{3\}$	(3)	$A(a)$	Assumption
$\{1\}$	(4)	B	$\wedge$E 1
$\{1,3\}$	(5)	$A(a) \wedge B$	$\wedge$I 3, 4
$\{1,3\}$	(6)	$\exists x(A(x) \wedge B)$	$\exists$I 5
$\{1\}$	(7)	$\exists x(A(x) \wedge B)$	$\exists$E 2, 3-6

Example 8.6.

$$\exists x\big(A(x) \vee B\big) \Leftrightarrow \exists x A(x) \vee B$$

$\{1\}$	(1)	$\exists x\big(A(x) \vee B\big)$	Premise
$\{2\}$	(2)	$A(a) \vee B$	Assumption
$\{3\}$	(3)	$A(a)$	Assumption
$\{3\}$	(4)	$\exists x A(x)$	$\exists$I 3
$\{3\}$	(5)	$\exists x A(x) \vee B$	$\vee$I 4
$\{6\}$	(6)	B	Assumption
$\{6\}$	(7)	$\exists x A(x) \vee B$	$\vee$I 6
$\{2\}$	(8)	$\exists x A(x) \vee B$	$\vee$E 2, 3-5, 6-7
$\{1\}$	(9)	$\exists x A(x) \vee B$	$\exists$E 1, 2-8

$\{1\}$	(1)	$\exists x A(x) \vee B$	Premise
$\{2\}$	(2)	$\exists x A(x)$	Assumption
$\{3\}$	(3)	$A(a)$	Assumption
$\{3\}$	(4)	$A(a) \vee B$	$\vee$I 3
$\{3\}$	(5)	$\exists x\big(A(x) \vee B\big)$	$\exists$I 4
$\{2\}$	(6)	$\exists x\big(A(x) \vee B\big)$	$\exists$E 2, 3-5
$\{7\}$	(7)	B	Assumption
$\{7\}$	(8)	$A(a) \vee B$	$\vee$I 7
$\{7\}$	(9)	$\exists x\big(A(x) \vee B\big)$	$\exists$I 8
$\{1\}$	(10)	$\exists x\big(A(x) \vee B\big)$	$\vee$E 1, 2-6, 7-9

Example 8.7.

$$\forall x\big(A(x) \to B\big) \Leftrightarrow \exists x A(x) \to B$$

$\{1\}$	(1)	$\forall x\big(A(x) \to B\big)$	Premise
$\{2\}$	(2)	$\exists x A(x)$	Assumption
$\{3\}$	(3)	$A(a)$	Assumption
$\{1\}$	(4)	$A(a) \to B$	$\forall$E 1
$\{1,3\}$	(5)	B	$\to$E 3, 4
$\{1,2\}$	(6)	B	$\exists$E 2, 3-5
$\{1\}$	(7)	$\exists x A(x) \to B$	$\to$I 2-6

$\{1\}$	(1)	$\exists x A(x) \to B$	Premise
$\{2\}$	(2)	$A(a)$	Assumption
$\{2\}$	(3)	$\exists x A(x)$	∃I 2
$\{1,2\}$	(4)	B	→E 1, 3
$\{1\}$	(5)	$A(a) \to B$	→I 2-4
$\{1\}$	(6)	$\forall x\big(A(x) \to B\big)$	∀I 5

Example 8.8.

$$\exists x\big(A(x) \to B\big) \Leftrightarrow \forall x A(x) \to B$$

$\{1\}$	(1)	$\exists x\big(A(x) \to B\big)$	Premise
$\{2\}$	(2)	$A(a) \to B$	Assumption
$\{3\}$	(3)	$\forall x A(x)$	Assumption
$\{3\}$	(4)	$A(a)$	∀E 3
$\{2,3\}$	(5)	B	→E 2, 4
$\{2\}$	(6)	$\forall x A(x) \to B$	→I 3-5
$\{1\}$	(7)	$\forall x A(x) \to B$	∃E 1, 2-6

$\{1\}$	(1)	$\forall x A(x) \to B$	Premise
$\{2\}$	(2)	$\forall x A(x)$	Assumption
$\{3\}$	(3)	$A(a)$	Assumption
$\{1,2\}$	(4)	B	→E 1, 2
$\{1,2\}$	(5)	$A(a) \to B$	→I 3-4
$\{1,2\}$	(6)	$\exists x\big(A(x) \to B\big)$	
$\{7\}$	(7)	$\neg\forall x A(x)$	Assumption
$\{7\}$	(8)	$\exists x \neg A(x)$	¬∀E 7
$\{9\}$	(9)	$\neg A(a)$	Assumption
$\{10\}$	(10)	$A(a)$	Assumption
$\{9,10\}$	(11)	$\bot$	¬E 9, 10
$\{9,10\}$	(12)	B	EFQ 11
$\{9\}$	(13)	$A(a) \to B$	→I 10-12
$\{9\}$	(14)	$\exists x\big(A(x) \to B\big)$	∃I 13
$\{7\}$	(15)	$\exists x\big(A(x) \to B\big)$	∃E 8, 9-14
$\{1\}$	(16)	$\exists x\big(A(x) \to B\big)$	TND 2-6, 7-14

Example 8.9.

$$\exists x\big(B \to A(x)\big) \Leftrightarrow B \to \exists x A(x)$$

$\{1\}$	(1)	$\exists x\big(B \to A(x)\big)$	Premise
$\{2\}$	(2)	$B \to A(a)$	Assumption (to be discharged by ∃E)
$\{3\}$	(3)	B	Assumption (to be discharged by →I
$\{2,3\}$	(4)	$A(a)$	→E 2-3
$\{2,3\}$	(5)	$\exists x A(x)$	∃I 4
$\{2\}$	(6)	$B \to \exists x A(x)$	→I 3-5 (discharges 3)
$\{1\}$	(7)	$B \to \exists x A(x)$	∃E 1, 2-6 (discharges 2)

$\{1\}$	(1)	$B \to \exists x A(x)$	Premise
$\{2\}$	(2)	B	Assumption (to be discharged by TND)
$\{1, 2\}$	(3)	$\exists x A(x)$	→E 1, 2
$\{4\}$	(4)	$A(a)$	Assumption (to be discharged by ∃E)
$\{5\}$	(5)	B	Assumption (to be discharged by →I)
$\{4\}$	(6)	$A(a)$	Axiom 4
$\{4\}$	(7)	$B \to A(a)$	→I 5-6 (discharges 5)
$\{4\}$	(8)	$\exists x \bigl(B \to A(x)\bigr)$	∃I 7
$\{1, 2\}$	(9)	$\exists x \bigl(B \to A(x)\bigr)$	∃E 3, 4-8 (discharges 4)
$\{10\}$	(10)	$\neg B$	Assumption (to be discharged by TND)
$\{11\}$	(11)	$\neg \exists x \bigl(B \to A(x)\bigr)$	Assumption (to be discharged by RAA')
$\{12\}$	(12)	B	Assumption (to be discharged by →I)
$\{10, 12\}$	(13)	$\bot$	¬E 10, 12
$\{10, 12\}$	(14)	$A(a)$	EFQ 13
$\{10\}$	(15)	$B \to A(a)$	→I 12-14 (discharges 12)
$\{10\}$	(16)	$\exists x \bigl(B \to A(x)\bigr)$	∃I 15
$\{10, 11\}$	(17)	$\bot$	¬E 11, 16
$\{10\}$	(18)	$\exists x \bigl(B \to A(x)\bigr)$	RAA' 11-17 (discharges 11)
$\{1\}$	(19)	$\exists x \bigl(B \to A(x)\bigr)$	TND 2-9, 10-18 (discharges 2 and 10)