III Logic

It is now a good time to be more specific about the precise meaning of mathematical statements. They are governed by the rules of logic.

8 Boolean Algebra

Logic is generally considered to lie in the intersection between Philosophy and Mathematics. It studies the meaning of statements and the relationship between them.

Logical statements in computer programs. Programming languages provide all the tools to be excessively precise. This includes *logical statements* which are used to construct loops, among other things. As an example, consider a while loop that exchanges adjacent array elements until some condition expressed by a logical statement is satisfied. Putting the while loop inside a for loop we get a piece of code that sorts an array $A[1..n]$:

```
for i = 1 to n do j = i;
  while j > 1 and A[j] > A[j − 1] do
    a = A[j]; A[j] = A[j − 1]; A[j − 1] = a;
    j = j − 1
  endwhile
endfor.
```

This particular method for sorting is often referred to as insertion sort because after $i-1$ iterations, $A[1..i-1]$ is sorted, and the i-th iteration inserts the i-th element such that $A[1..i]$ is sorted. We illustrate the algorithm in Figure 7. Here we focus on the logic that controls the while loop.

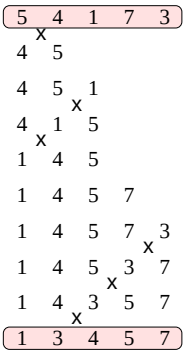

Figure 7: The insertion sort algorithm applied to an unsorted sequence of five integers.

The iteration is executed as long as two conditions hold, namely "$j > 1$" and "$A[j] > A[j-1]$". The first prevents we step beyond the left end of the array. The second condition limits the exchanges to cases in which adjacent elements are not yet in non-decreasing order. The two conditions are connected by a logical and, which requires both to be true.

Boolean operations. A logical statement is either true (T) of false (F). We call this the *truth value* of the statement. We will frequently represent the statement by a *variable* which can be either true or false. A *boolean operation* takes one or more truth values as input and produces a new output truth value. It thus functions very much like an arithmetic operation. For example, *negation* is a unary operation. It maps a truth value to the opposite; see Table 6. Much more common are binary operations; such as

p	$\neg p$
T	F
F	T

Table 6: Truth table for negation ($\neg$).

and, or, and exclusive or. We use a truth table to specify the values for all possible combinations of inputs; see Table 7. Binary operations have two input variables, each in one of two states. The number of different inputs is therefore only four. We have seen the use of these particular

p	q	$p \wedge q$	$p \vee q$	$p \oplus q$
T	T	T	T	F
T	F	F	T	T
F	T	F	T	T
F	F	F	F	F

Table 7: Truth table for and ($\wedge$), or ($\vee$), and exclusive or ($\oplus$) operations.

boolean operations before, namely in the definition of the common set operations; see Figure 8.

$$\begin{aligned} A^c &= \{x \mid x \notin A\}; \\ A \cap B &= \{x \mid x \in A \text{ and } x \in B\}; \\ A \cup B &= \{x \mid x \in A \text{ or } x \in B\}; \\ A \oplus B &= \{x \mid x \in A \text{ xor } x \in B\}; \\ A - B &= \{x \mid x \in A \text{ and } x \notin B\}. \end{aligned}$$

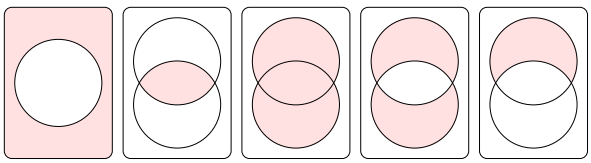

Figure 8: From left to right: the complement of one set and the intersection, union, symmetric difference, and difference of two sets.

Algebraic properties. We observe that boolean operations behave very much like ordinary arithmetic operations. For example, they follow the same kind of rules when we transform them.

- All three binary operations are commutative, that is,

$$\begin{aligned} p \wedge q \quad &\text{iff} \quad q \wedge p; \\ p \vee q \quad &\text{iff} \quad q \vee p; \\ p \oplus q \quad &\text{iff} \quad q \oplus p. \end{aligned}$$

- The and operation distributes over the or operation, and vice versa, that is,

$$\begin{aligned} p \wedge (q \vee r) \quad &\text{iff} \quad (p \wedge q) \vee (p \wedge r); \\ p \vee (q \wedge r) \quad &\text{iff} \quad (p \vee q) \wedge (p \vee r). \end{aligned}$$

Similarly, negation distributes over and and or, but it changes one into the other as it does so. This is known as de Morgan's Law.

DE MORGAN'S LAW. Letting p and q be two variables,

$$\begin{aligned} \neg(p \wedge q) \quad &\text{iff} \quad \neg p \vee \neg q; \\ \neg(p \vee q) \quad &\text{iff} \quad \neg p \wedge \neg q. \end{aligned}$$

PROOF. We construct the truth table, with a row for each combination of truth values for p and q; see Table 8. Since

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

Table 8: The truth table for the expressions on the left and the right of the first de Morgan Law.

the two relations are symmetric, we restrict our attention to the first. We see that the truth values of the two expressions are the same in each row, as required. □

Implications. The *implication* is another kind of binary boolean operation. It frequently occurs in statements of lemmas and theorems. An example is Fermat's Little Theorem. To emphasize the logical structure, we write A for the statement "n is prime" and B for "$a^{n-1} \bmod n = 1$ for every non-zero $a \in \mathbb{Z}_n$". There are different, equivalent ways to restate the theorem, namely "if A then B"; "A implies B"; "A only if B"; "B if A". The operation is

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

Table 9: The truth table for the implication ($\Rightarrow$).

defined in Table 9. We see the contrapositive in the second column on the right, which is equivalent, as expected. We also note that q is forced to be true if p is true and that q can be anything if p is false. This is expressed in the third column on the right, which relates to the last column by de Morgan's Law. The corresponding set operation is the complement of the difference, $(A - B)^c$; see Figure 9 on the left.

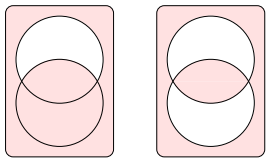

Figure 9: Left: the complement of the difference between the two sets. Right: the complement of the symmetric difference.

We recall that a logical statement is either true or false. This is referred to as the law of the *excluded middle*. In other words, a statement is true precisely when it is not false. There is no allowance for ambiguities or paradoxes. An example is the sometimes counter-intuitive definition that false implies true is true. Write A for the statement "it is raining", B for "I use my umbrella", and consider $A \Rightarrow B$. Hence, if it is raining then I use my umbrella. This does not preclude me from using the umbrella if it is not raining. In other words, the implication is not false if I use my umbrella without rain. Hence, it is true.

Equivalences. If implications go both ways, we have an *equivalence*. An example is the existence of a multiplicative inverse iff the multiplication permutes. We write A for the statement "a has a multiplicative inverse in $\mathbb{Z}_n$" and B for "the function $M : \mathbb{Z}_n \to \mathbb{Z}_n$ defined by $M(x) = a \cdot_n x$ is bijective". There are different, equivalent ways to restate the claim, namely "A if and only if B" and "A and B are equivalent". The operation is defined in Table 10. The last column shows that equivalence is the opposite of the exclusive or operation. Figure 9 shows the corresponding set operation on the right.

Recalling the definition of a group, we may ask which

p	q	$p \Leftrightarrow q$	$(p \Rightarrow q) \wedge (q \Rightarrow p)$	$\neg(p \oplus q)$
T	T	T	T	T
T	F	F	F	F
F	T	F	F	F
F	F	T	T	T

Table 10: The truth table for the equivalence ($\Leftrightarrow$).

of the binary operations form an Abelian group. The set is $\{\mathsf{F}, \mathsf{T}\}$. One of the two must be the neutral element. If we choose F then $\mathsf{F} \circ \mathsf{F} = \mathsf{F}$ and $\mathsf{F} \circ \mathsf{T} = \mathsf{T} \circ \mathsf{F} = \mathsf{T}$. Furthermore, $\mathsf{T} \circ \mathsf{T} = \mathsf{F}$ is necessary for T to have an inverse. We see that the answer is the exclusive or operation. Mapping F to 0 and T to 1, as it is commonly done in programming languages, we see that the exclusive or can be interpreted as adding modulo 2. Hence, $(\{\mathsf{F}, \mathsf{T}\}, \oplus)$ is isomorphc to $(\mathbb{Z}_2, +_2)$.

Summary. We have learned about the main components of logical statements, boolean variables and operations. We have seen that the operations are very similar to the more familiar arithmetic operations, mapping one or more boolean input variable to a boolean output variable.

9 Quantifiers

Logical statements usually include *variables*, which range over sets of possible instances, often referred to as *universes*. We use quantifiers to specify that something holds for all possible instances or for some but possibly not all instances.

Universal and existential quantifiers. We introduce the concept by taking an in-depth look at a result we have discussed in Chapter II.

EUCLID'S DIVISION THEOREM. Letting n be a positive integer, for every integer m there are unique integers q and r, with $0 \leq r < n$, such that $m = nq + r$.

In this statement, we have n, m, q, r as variables. They are integers, so $\mathbb{Z}$ is the universe, except that some of the variables are constrained further, that is, $n \geq 1$ and $0 \leq r < n$. The claim is "for all" m "there exist" q and r. These are quantifiers expressed in English language. The first is called the *universal quantifier*:

$\forall x \ [p(x)]$**:** for all instantiations of the variable x, the statement $p(x)$ is true.

For example, if x varies over the integers then this is equivalent to

$$\ldots \wedge p(-1) \wedge p(0) \wedge p(1) \wedge p(2) \wedge \ldots$$

The second is the *existential quantifier*:

$\exists x \ [q(x)]$**:** there exists an instantiation of the variable x such that the statement $q(x)$ is true.

For the integers, this is equivalent to

$$\ldots \vee q(-1) \vee q(0) \vee q(1) \vee q(2) \vee \ldots$$

With these quantifiers, we can restate Euclid's Division Theorem more formally:

$$\forall n \geq 1 \ \forall m \ \exists q \ \exists 0 \leq r < n [m = nq + r].$$

Negating quantified statements. Recall de Morgan's Law for negating a conjunction or a disjunction:

$$\begin{aligned} \neg(p \wedge q) &\quad \Leftrightarrow \quad \neg p \vee \neg q; \\ \neg(p \vee q) &\quad \Leftrightarrow \quad \neg p \wedge \neg q. \end{aligned}$$

The corresponding rules for quantified statements are

$$\begin{aligned} \neg\,(\forall x \ [p(x)]) &\quad \Leftrightarrow \quad \exists x \ [\neg p(x)]; \\ \neg\,(\exists x \ [q(x)]) &\quad \Leftrightarrow \quad \forall x \ [\neg q(x)]. \end{aligned}$$

We get the first line by applying de Morgan's first Law to the conjunction that corresponds to the expression on the left hand side. Similarly, we get the second line by applying de Morgan's second Law. Alternatively, we can derive the second line from the first. Since both sides of the first line are equivalent, so are its negations. Now, all we need to do it to substitute $\neg q(x)$ for $p(x)$ and exchange the two sides, which we can because $\Leftrightarrow$ is commutative.

Big-Oh notation. We practice the manipulation of quantified statements by discussing the big-Oh notation for functions. It is commonly used in statements about the convergence of an iteration or the running time of an algorithm. We write $\mathbb{R}^+$ for the set of positive real numbers.

DEFINITION. Let f and g be functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. Then $f = \mathrm{O}(g)$ if there are positive constants c and n_0 such that $f(x) \leq cg(x)$ whenever $x > n_0$.

This notation is useful in comparing the asymptotic behavior of the functions f and g, that is, beyond a constant n_0. If $f = \mathrm{O}(g)$ then f can grow at most a constant times as fast as g. For example, we do not have $f = \mathrm{O}(g)$ if $f(x) = x^2$ and $g(x) = x$. Indeed, $f(x) = xg(x)$ so there is no constant c such that $f(x) \leq cg(x)$ because we can always choose x larger than c and n_0 and get a contradiction. We rewrite the definition in more formal notation. The statement $f = \mathrm{O}(g)$ is equivalent to

$$\exists c > 0 \ \exists n_0 > 0 \ \forall x \in \mathbb{R} \ [x > n_0 \Rightarrow f(x) \leq cg(x)].$$

We can simplify by absorbing the constraint of x being larger than the constant n_0 into the last quantifying statement:

$$\exists c > 0 \ \exists n_0 > 0 \ \forall x > n_0 \ [f(x) \leq cg(x)].$$

We have seen above that negating a quantified statement reverses all quantifiers and pulls the negation into the unquantified, inner statement. Recall that $\neg(p \Rightarrow q)$ is equivalent to $p \wedge \neg q$. Hence, the statement $f \neq \mathrm{O}(g)$ is equivalent to

$$\forall c > 0 \ \forall n_0 > 0 \ \exists x \in \mathbb{R} \ [x > n_0 \wedge f(x) > cg(x)].$$

We can again simplify by absorbing the constraint on x into the quantifying statement:

$$\forall c > 0 \ \forall n_0 > 0 \ \exists x > n_0 \ [f(x) > cg(x)].$$

Big-Theta notation. Recall that the big-Oh notation is used to express that one function grows asymptotically at most as fast as another, allowing for a constant factor of difference. The big-Theta notation is stronger and expresses that two functions grow asymptotically at the same speed, again allowing for a constant difference.

DEFINITION. Let f and g be functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. Then $f = \Theta(g)$ if $f = \mathrm{O}(g)$ and $g = \mathrm{O}(f)$.

Note that in big-Oh notation, we can always increase the constants c and n_0 without changing the truth value of the statement. We can therefore rewrite the big-Theta statement using the larger of the two constants c and the larger of the two constants n_0. Hence, $f = \Theta(g)$ is equivalent to

$$\exists c > 0\, \exists n_0 > 0\, \forall x > n_0\ [f(x) \leq cg(x) \wedge g(x) \leq cf(x)].$$

Here we can further simplify by rewriting the two inequalities by a single one: $\frac{1}{c}g(x) \leq f(x) \leq cg(x)$. Just for practice, we also write the negation in formal notation. The statement $f \neq \Theta(f)$ is equivalent to

$$\forall c > 0\, \forall n_0 > 0\, \exists x > n_0\ [cg(x) < f(x) \vee cf(x) < g(x)].$$

Because the two inequalities are connected by a logical or, we cannot simply combine them. We could by negating it first, $\neg(\frac{1}{c}g(x) \leq f(x) \leq cg(x))$, but this is hardly easier to read.

Big-Omega notation. Complementary to the big-Oh notation, we have

DEFINITION. Let f and g be functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. Then $f = \Omega(g)$ if $g = \mathrm{O}(f)$.

In formal notation, $f = \Omega(g)$ is equivalent to

$$\exists c > 0\, \exists n_0 > 0 \forall x > n_0\ [f(x) \geq cg(x)].$$

We may think of big-Oh like a less-than-or-equal-to for functions, and big-Omega as the complementary greater-than-or-equal-to. Just as we have $x = y$ iff $x \leq y$ and $x \geq y$, we have $f = \Theta(g)$ iff $f = \mathrm{O}(g)$ and $f = \Omega(g)$.

Little-oh and little-omega notation. For completeness, we add notation that corresponds to the strict less-than and greater-than relations.

DEFINITION. Let f and g be functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. Then $f = \mathrm{o}(g)$ if for all constants $c > 0$ there exists a constant $n_0 > 0$ such that $f(x) < cg(x)$ whenever $x > n_0$. Furthermore, $f = \omega(g)$ if $g = \mathrm{o}(f)$.

This is not equivalent to $f = \mathrm{O}(g)$ and $f \neq \Omega(g)$. The reason for this is the existence of functions that cannot be compared at all. Consider for example $f(x) = x^2(\cos x + 1)$. For $x = 2k\pi$, k a non-negative integer, we have $f(x) = 2x^2$, while for $x = (2k+1)\pi$, we have $f(x) = 0$. Let $g(x) = x$. For even multiples of π, f grows much fast than g, while for odd multiples of π it grows much slower than g, namely not at all. We rewrite the little-Oh notation in formal notation. Specifically, $f = \mathrm{o}(g)$ is equivalent to

$$\forall c > 0\, \exists n_0 > 0\, \forall x > n_0\ [f(x) < cg(x)].$$

Similarly, $f = \omega(g)$ is equivalent to

$$\forall c > 0\, \exists n_0 > 0\, \forall x > n_0\ [f(x) > \frac{1}{c}g(x)].$$

In words, no matter how small our positive constant c is, there always exists a constant n_0 such that beyond that constant, $f(x)$ is larger than $g(x)$ over c. Equivalently, no matter how big our constant c is, there always exists a constant n_0 such that beyond that constant, $f(x)$ is larger than c times $g(x)$. We can thus simplify the formal statement by substituting $[f(x) > cg(x)]$ for the inequality.

10 Inference

In this section, we discuss the application of logic to proving theorems. In principle, every proof should be reducible to a sequence of simple logical deductions. While this is not practical for human consumption, there have been major strides toward that goal in computerized proof systems.

Modus ponens. This is an example of *direct inference*, the cornerstone of logical arguments.

PRINCIPLE OF MODUS PONENS. From p and $p \Rightarrow q$, we may conclude q.

We read this as a recipe to prove q. First we prove p, then we prove that p implies q, and finally we conclude q. Let us take a look at Table 11 to be sure. We see that modus

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

Table 11: The truth table for modus ponens.

ponens is indeed a tautology, that is, it is always true. Every theorem is this way, namely always true.

Other methods of direct inference. There are many other direct proof principles, all easy to verify. Some are straighforward re-interpretations of logical formulas and others use logical equivalences we have learned about. Here are but a few:

$$\begin{aligned} p \text{ and } q \quad &\text{then} \quad p \wedge q; \\ p \text{ or } q \quad &\text{then} \quad p \vee q; \\ q \text{ or } \neg p \quad &\text{then} \quad p \Rightarrow q; \\ \neg q \text{ and } p \quad &\text{then} \quad p \not\Rightarrow q; \\ p \Rightarrow q \text{ and } q \Rightarrow p \quad &\text{then} \quad p \Leftrightarrow q; \\ p \Rightarrow q \text{ and } q \Rightarrow r \quad &\text{then} \quad p \Leftrightarrow r. \end{aligned}$$

The last principle is perhaps more interesting than the others because it is the only one among the six that is not an equivalence; see Table 12.

Contrapositive. This is the first example of an *indirect inference* method.

p	q	r	$((p \Rightarrow q) \wedge (q \Rightarrow r))$	$\Rightarrow$	$(p \Rightarrow r)$
T	T	T	T T T	T	T
T	T	F	T F F	T	F
T	F	T	F F T	T	T
T	F	F	F F T	T	F
F	T	T	T T T	T	T
F	T	F	T F F	T	T
F	F	T	T T T	T	T
F	F	F	T T T	T	T

Table 12: The truth table for reasoning by transitivity.

PRINCIPLE OF CONTRAPOSITION. The statements $p \Rightarrow q$ and $\neg q \Rightarrow \neg p$ are equivalent, and so a proof of one is a proof of the other.

We have seen a truth table that shows the equivalence of the two statements earlier, in Section 8. Let us look at an example.

CLAIM. If n is a positive integer with $n^2 > 25$ then $n > 5$.

PROOF. The statement p is that n is a positive integer whose square is larger than 25. The statement q is that n is larger than 5. We could argue directly but then we would need to know something about talking square roots. Instead, let us argue indirectly. Suppose $\neg q$, that is, $n \leq 5$. By monotonicity of multiplication, we have

$$n^2 \;\leq\; 5n \;\leq\; 5 \cdot 5 \;\leq\; 25.$$

Now, by transitivity of the smaller-than-or-equal-to relation, we have $n^2 \leq 25$. Thus $\neg q$ implies $\neg p$. □

Example: Chinese remainders. Another instructive example is a result we have seen in Section 6. Let m and n be relative prime, positive integers. We map each integer in $\mathbb{Z}_{mn}$ to the pair of remainders, that is, for $0 \leq x < mn$ we define $f(x) = (x \bmod m, x \bmod n)$.

CHINESE REMAINDER THEOREM. If $x \neq y$ both belong to $\mathbb{Z}_{mn}$ then $f(x) \neq f(y)$.

PROOF. We use again the indirect approach by contraposition. Assume $f(x) = f(y)$. Then

$$\begin{aligned} x \bmod m &= y \bmod m; \\ x \bmod n &= y \bmod n. \end{aligned}$$

Hence,

$$\begin{aligned}(x-y) \bmod m &= 0;\\ (x-y) \bmod n &= 0.\end{aligned}$$

Therefore, $x-y$ is a multiple of both m and n. Hence, $(x-y) \bmod mn = 0$ and therefore $x \bmod mn = y \bmod mn$, which contradicts that $x \neq y$ in $\mathbb{Z}_{mn}$. □

Reduction to Absurdity. Another powerful indirect proof technique is by contradiction.

PRINCIPLE OF REDUCTION TO ABSURDITY. If from assuming p and $\neg q$ we can derive r as well as $\neg r$ then $p \Rightarrow q$.

Here r can be any statement. Often we use a statement r that is always true (or always false) so that we only need to derive $\neg r$ (or r) from p and $\neg q$. Let us take a look at Table 13. As with all the proof methods, it is best to see examples. There are many and a large variety because different principles are combined, or made more complicated, etc.

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

Table 13: The truth table for the reduction to absurdity.

Example: irrational numbers. A real number u is *rational* if there are integers m and n such that $u = \frac{m}{n}$ and *irrational* otherwise. The set of rational numbers is denoted as $\mathbb{Q}$. For any two different rational numbers, $u < w$, we can always find a third that lies strictly between them. For example, if $w = \frac{k}{l}$ then

$$\begin{aligned}v &= \frac{u+w}{2}\\ &= \frac{ml+nk}{nl}\end{aligned}$$

lies halfway between u and w. This property is sometimes expressed by saying the rational numbers are *dense* in the set of real numbers. How do we know that not all real numbers are rational?

CLAIM. $\sqrt{5}$ is irrational.

PROOF. Assume the square root of 5 is rational, that is, there exist integers m and n such that $\sqrt{5} = \frac{m}{n}$. Squaring the two sides, we get

$$5 = \frac{m^2}{n^2}$$

or, equivalently, $5n^2 = m^2$. But m^2 has an even number of prime factors, namely each factor twice, while $5n^2$ has an odd number of prime factors, namely 5 together with an even number of prime factors for n^2. Hence, $5n^2 = m^2$ is not possible, a contradiction. □

We take a look at the logic structure of this proof. Let p be the statement that $\sqrt{5}^2 = 5$ and q the statement that $\sqrt{5}$ is irrational. Thus $\neg q$ is the statement that $\sqrt{5} = \frac{m}{n}$. From assuming p and $\neg q$, we derive r, that is the statement $5n^2 = m^2$. But we also have $\neg r$, because each integer has a unique decomposition into prime factors. We thus derived r and $\neg r$. But this cannot be true. Using the Principle of Reduction to Absurdity, we conclude that p implies q. By modus ponens, assuming p gives q.

Summary. We have learned that theorems are tautologies and there are different ways to prove them. As applications of logic rules we have discussed direct methods (Principle of Modus Ponens) and indirect methods (Principle of Contrapositive and Principle of Reduction to Absurdity).

Third Homework Assignment

Write the solution to each problem on a single page. The deadline for handing in solutions is February 23.

Question 1. ($20 = 10 + 10$ points). (Problem 3.1-6 in our textbook). Show that $p \oplus q$ is equivalent to $(p \wedge \neg q) \vee (\neg p \wedge q)$. State the corresponding relation in terms of sets and set operations.

Question 2. ($20 = 10 + 10$ points). (Problem 3.2-14 in our textbook). Let x, y, z be variables and p, q logical statements that depend on one variable.

(a) Are the following two compound logical statements equivalent?

1. $(\exists x \in \mathbb{R}\ [p(x)]) \wedge (\exists y \in \mathbb{R}\ [q(y)])$;
2. $\exists z \in \mathbb{R}\ [p(z) \wedge q(z)]$.

(Justify your answer.)

(b) Are the following two compound logical statements equivalent?

1. $(\exists x \in \mathbb{R}\ [p(x)]) \vee (\exists y \in \mathbb{R}\ [q(y)])$;
2. $\exists z \in \mathbb{R}\ [p(z) \vee q(z)]$.

(Justify your answer.)

Question 3. (20 points). (Problem 3.3-6 in our textbook). Is the statement $p \Rightarrow q$ equivalent to the statement $\neg p \Rightarrow \neg q$? (If yes, why? If no, why not?)

Question 4. (20 points). (Problem 3.3-14 in our textbook). Prove that there is no largest prime number. In other words, for every prime number there is another, larger prime number.