# Introduction to Discrete Mathematics

Viktor Verbovskiy

# Chapter 1

# Introduction to Mathematical Logic

Человек любит размышлять. Но как проверить, что в его рассуждениях нет ошибки. Например, где ошибка в следующем софизме[1]: «Что ты не потерял, ты имеешь. Рогов ты не терял. Стало быть, ты рогат.»

И следующее рассуждение кажется правдоподобным: «Вор не желает приобрести ничего дурного. Приобретение хорошего есть дело хорошее. Следовательно, вор желает хорошего.» Но вряд ли кому-нибудь нравится заключение этого рассуждения. Думаю, что даже вору не нравится заключение, особенно, если под заключением понимать заключение в тюремной камере.

В предыдущем предложении слово „заключение“ имело два разных смысла. Такая двусмысленность естественных языков — дело вполне обычное. И если в литературе такое положение вещей является благом, то в науке, наоборот, двусмысленность может привести к ошибке. Представьте, что создан робот, понимающий человеческую речь, и вы ему говорите: „Завтра Борька пойдёт на шашлыки. Сделай всё необходимое.“ Если Борька человек, то роботу надо подготовить для него мангал, шампуры и прочее. А если Борька — свинья (в смысле, животное, а не человек, ведущий себя по-свински), то что надо будет сделать роботу, чтобы свинью к шашлыку подготовить? Что будет, если робот не так вас поймет? Как построить язык, в котором нет двусмысленностей, в котором всё точно, — один из вопросов, на который даёт ответ математическая логика.

Математическая логика объясняет, как по примеру одного предложения строить другое, вроде бы совсем на него не похожее по смыслу, но имеюшее ту же самую истинность. Переделаем предложение про вора в подобное предложение про студента: «Если студент списывает на экзамене, то потому, что он желает получить хорошую оценку. Получить хорошую оценку есть дело хорошее. Следовательно, когда студент списывает, то он поступает хорошо». Действительно ли хорошо списывать, вы можете узнать на экзамене, а как узнать, где допушена ошибка в данном рассуждении вам поможет предмет под название „Математическая логика“.

Математическая логика изучает такие основополагающие для математики вещи, как математический язык, включая такие понятия как множество, алгоритм, доказательство и другие. В математической логике изучают как математически точно записать то или иное выражение, как избавиться от многозначности и пародоксальности естественных языков. И еще, математическая логика помогает избегать ошибок в рассуждениях.

[1]Софизм (от греч. $\sigma\acute{o}\phi\iota\sigma\mu\alpha$, «мастерство, умение, хитрая выдумка, уловка, мудрость») — ложное высказывание, которое, тем не менее, при поверхностном рассмотрении кажется правильным, софизм основан на преднамеренном, сознательном нарушении правил логики.

## 1.1 Propositions

### 1.1.1 Basic definition

Одна из главных задача математики — как по данному предположению определить, истинное оно или ложное. Истина и ложь — основные понятия в математике. А тот объект, который может быть либо истинным, либо ложным, называется *высказыванием*. Высказывания являются основной рассуждений.

**Definition 1.1.1** A sentence which is either true or false (but not both) is called *a proposition*.

Предложение, которое может быть либо истинным, либо ложным, называется *высказыванием*.

Понятно, что высказванием может быть только повествовательное предложение. Повелительные предложения, например „Пойдём погуляем!“, или вопросительные, скажем, „Идёт ли дождь?“, не могут быть ни истинными, ни ложными.

Но и не всякое повествовательное предложение может быть высказыванием. Знаменитый парадокс „Лжец“, сформулированный древнегреческим мыслителем Евбулидом[2], — отличный тому пример. В простейшем варианте этого парадокса человек произносит всего одну фразу: „Я лгу“. Или говорит: „Высказывание, которое я сейчас произношу, является ложным“. Или: „Это высказывание ложно“.

Требуется решить, является ли это предложение высказыванием, то есть, может ли оно быть истинным или ложным. Рассмотрим обе возможности. Если высказывание ложно, то говорящий сказал правду и, значит, сказанное им не является ложью. Если же высказывание не является ложным, а говорящий утверждает, что он солгал, то это его высказывание ложно. Оказывается, если говорящий лжет, то он говорит правду, и наоборот. Таким образом, поскольку данное предложение не может быть ни истинным, ни ложным, оно не является высказыванием.

Consider the sentence “This sentence is false“. If this sentence is false, then what it says is true, but if it it true, then we must believe what it says, and it says that it is false, so, it must be false. Thus, the sentence “This sentence is false“ is neither true nor false. Hence, it is not a proposition.

Иногда бывает невозможно решить, является ли данное предложение истинным или ложным. Например, „Ровно 30000 лет назад в полдень в этом месте на небе не было ни облачка.“ Мы не знаем ответа, но, тем не менее, это предложение — высказывание, поскольку либо на небе было хотя бы одно облако, либо их не было вовсе.

Заметим, что здесь речь идёт о классической логике, где каждое утверждение либо истинно, либо ложно, а не о квантовой, где всё намного запутаннее. Так, обычный кот либо жив, либо мёртв, то есть фраза „Этот кот жив“ является высказыванием, а вот кот Шрёдингера и не жив, и не мёртв, и с точки зрения квантовой физики фраза „Кот Шрёдингера жив“ ни истинна, ни ложна.

**Glossary**

| | | |
|---|---|---|
| A definition — определение | A sentence — предложение | True — истина |
| A proposition — высказывание | | False — ложь |

[2] Евбулид (из Милета; Эвбулид, Eubulides, $E\upsilon\beta o\upsilon\lambda\iota\delta\eta\varsigma$; IV век до н. э.) — древнегреческий философ-идеалист, представитель мегарской школы, известен своими парадоксами или „апориями“: „Лжец“, „Куча“, „Плешивый“, „Рогатый“ и др.

### 1.1.2 Elementary propositions

Рассмотрим простое предложение: „Если идёт дождь, то я беру зонт.“ Действительно ли оно такое простое? Нет, оно состоит из двух элементарных высказываний „идёт дождь“ и „беру зонт“ и логической связки между ними: „если ..., то ...“.

А что можно сказать о высказывании „Руслан и Людмила любят цветы“? Оно состоит из высказываний „Руслан любит цветы“, „Людмила любит цветы“ и логической связки „и“. То есть его можно переписать в виде „Руслан любит цветы, и Людмила любит цветы“. Звучит коряво, но смысл тот же.

Но не всё так просто. Слегка переделаем последнее высказывание: „Руслан и Людмила родили ребёнка.“ Можно ли сказать, что предложение „Руслан родил ребёнка, и Людмила родила ребёнка“ несёт тот же смысл? Не совсем так. В первом случае мы точно понимаем, что они родили ребёнка вместе, это их общий ребенок, тогда как во втором случае может так получится, что каждый родил по ребёнку от кого-то другого, и в результате родилось два ребёнка.

Значит, высказвание „Руслан и Людмила родили ребёнка“ является элементарным, а высказывание „Руслан и Людмила любят цветы“ — нет. Мы поговорим об этом более точно, когда будем изучать предикаты и их местность.

Как видим, бывает сложно определить, является ли высказывание элементарным или нет. Пока мы остановимся на следующем определении.

**Definition 1.1.2** A proposition which does not contain logical connectivities is called *an elementary proposition.* A proposition which does contain a logical connectivity is called *a compound proposition.*[3]

Высказывание, которое не содержит логических связок, называется *элементарным.* Высказывание, которое содержит логические связки, называется *составным.*

Это определение не самое удачное, потому что мы ещё не знаем, что такое „логическая связка“. Более того, любое определение, которое содержит отрицание, в данном случае мы отрицаем наличие логических связок, не может быть хорошим, если это только не отрицание конкретного объекта. Логических связок можно придумать бесконечно много, не можем же мы перечислять все эти логические связки до бесконечности!

Поэтому мы будем пользоваться другим определением.

**Definition 1.1.3** A proposition which is considered to be elementary is called *an elementary proposition.*

Высказывание, которое мы считаем элементарным, называется *элементарным высказыванием.*

Таким образом, выбор элементарных высказываний лежит на совести человека. Как он захочет, так и будет. Единственно, хорошо если его желания достаточно разумны, то есть если он старается придерживаться определения 1.1.2.

**Glossary**

A logical connectivity — логическая связка
An elementary proposition — элементарное высказывание
A compound proposition — составное (сложное) высказывание

[3] Обычно, служебное слово „does“ используется для построения отрицания, мы говорим „does not“. В данном случае, когда мы просто добавляем слово „does“, оно усиливает следующее за ним сказуемое. Это предложение можно перевести как „Высказывание, которое, действительно, содержит...“

### 1.1.3 A couple of words on names and notations

A system of mathematical notations is quite complicated and intricated. Its main aim is making writing mathematical laws shorter, because it is easier to remember just a single line than a whole page. Observe that programmers have another way of thinking up a notation for something. Indeed, it is not possible to make a program code sufficiently short in order to remember it, so it is better make a code readable, for instance, a programmer never says as a mathematician ‘Two people $A$ and $B$’, he says ‘Two people Alice and Bob’.

Система обозначений в математике достаточно сложная и запутанная, потому что главная её цель — укоротить запись математических законов, ведь запомнить одну строчку проще, чем целую страницу. Например, тождество $(a+b)^2 = a^2 + 2ab + b^2$ выглядет проще и короче своего словесного описания: квадрат суммы двух чисел равен сумме квадратов этих чисел и удвоенного их произведения.

Следует отметить, что у программистов задачи совсем другие. Как правило, код программы очень длинный, укоротить его до таких размеров, что можно было бы его запомнить, невозможно, поэтому они используют обозначения, по которым можно восстановить смысл, и их обозначения, как правило, длинные. Скажем, программист никогда не скажет, как математик: „В комнату вошли два человека: $A$ и $B$“, он скажет: „В комнату вошли два человека: Алиса и Боб“. Выражение „$A$ и $B$“ короче, но „Алиса и Боб“ — понятнее.

Обозначение, в некотором смысле, является именем объекта. Следует различать сам объект и его имя, обозначение. Пишем обозначение объекта, но думаем о самом объекте.

Для того, чтобы почувствовать разницу между именем объекта и самим объектом, давайте вспомним фрагмент из сказки Л. Кэррола „Алиса в зазеркалье“.

‘The name of the song is called “Haddocks’ Eyes.”’

‘Oh, that’s the name of the song, is it?’ Alice said, trying to feel interested.

‘No, you don’t understand,’ the Knight said, looking a little vexed. ‘That’s what the name is called. The name really is “The Aged Aged Man.”’

‘Then I ought to have said “That’s what the song is called”?’ Alice corrected herself.

‘No, you oughtn’t: that’s quite another thing! The song is called “Ways and Means”: but that’s only what it’s called, you know!’

‘Well, what is the song, then?’ said Alice, who was by this time completely bewildered.

‘I was coming to that,’ the Knight said. ‘The song really is “A-sitting On A Gate”.

— Заглавие этой песни называется „Пуговки для сюртуков“.

— Вы хотите сказать — песня так называется? — спросила Алиса, стараясь заинтересоваться песней.

— Нет, ты не понимаешь, — ответил нетерпеливо Рыцарь. — Это заглавие так называется. А песня называется „Древний старичок“.

— Мне надо было спросить: это у песни такое заглавие? — поправилась Алиса.

— Да нет! Заглавие совсем другое. „С горем пополам!“ Но это она только так называется!

— А песня эта какая? — спросила Алиса в полной растерянности.

— Я как раз собирался тебе об этом сказать. „Сидящий на стене“! Вот какая это песня![4]

[4]Вообще говоря, на этом месте правильнее было бы спеть саму песню, то есть показать сам объект, а не сказать: „Сидящий на стене“, что опять является лишь именем объекта.

### 1.1.4 A couple of words on names and notations. Continued

What is wrong in the picture 1.1?

Obviously, the teacher, who gave this exercise, expected another kind of answer.

Nevertheless, is it possible to say, that the answer is incorrect?

What does really mean ‘Find $x$’?

If we consider $x$ as an object, then the answer in the picture 1.1 is correct.

The problem is that in this exercise $x$ is not an object, it is a name of some object, in fact, this object is the number 5.

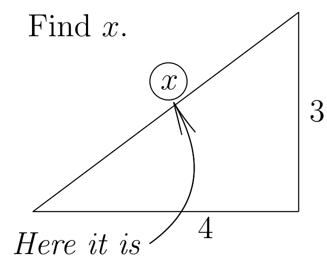


Figure 1.1: Find $x$

More correct formulation of ‘Find $x$’ are the following:

— Find such a value of the variable $x$ that substituting this value to $x$ in the picture gives a correct triangle.

— Find a real number such that replacing $x$ with this number gives a correct length of the hypotenuse of the given triangle.

As you can see the correct formulations much longer and less understandable.

Understanding the fact that $x$ is not an object, it is a name of some unknown object, and the aim of the given exercise is finding this object, allows us to give a simpler formulation of the same question.

There is one more convention.

Let we denote the proposition ‘Paris is the capital of France’ by $p$. We know that this proposition is true, so the truth value of $p$ is 1. But we write a simpler phrase: $p = 1$.

So, we obtain that at the same time $p =$‘Paris is the capital of France’ and $p = 1$.

There is no contradiction here, because ‘$p = 1$’ stands for ‘the truth value of $p$ is 1’.

Посмотрите на рисунок 1.1. Что в нём неправильно?

Очевидно, что учитель подразумевал другой ответ, что $x = 5$.

Но можно ли сказать, что данный ответ абсолютно некорректен?

Что в действительности обозначает фраза „Найдите $x$”?

Если мы будем рассматривать $x$ как объект, в данном случае, как букву алфавита, то ответ на приведенном рисунке правильный, там, действительно, найден объект $x$.

Но учитель имел в виду, что $x$ — это имя неизвестного объекта и что цель задачи — найти этот объект, которым, на самом деле, является некоторое число.

Более корректная, но и более нудная и труднопонимаемая формулировка этой задачи может звучать следующим образом:

— Найдите такое значение переменной $x$, подстановка которого вместо переменной $x$ на данном рисунке даёт возможный треугольник.

— Найдите такое вещественное число, что если $x$ заменить этим числом, то получим правильную длину гипотенузы в данном треугольнике.

Таким образом, понимание того, что $x$ — это всего лишь имя, ссылка на некоторый объект, позволяет гораздо проще формулировать постановку задачи.

И еще одно соглашение. Отныне мы не будем больше писать „истинностное значение высказывания $p$ равно 1”, поскольку это очень длинная запись.

Мы будем писать проще: „$p = 1$”, но подразумевать под этим вышеуказанную запись.

### 1.1.5 Logical connectives

Основыных логических связок не так уж и много. Мы уже встречали с вами такие связки, как логическое отрицание, логическое „и“, „если ..., то ...“.

К ним можно добавить логическое „или“ и „исключающее или“. Разница между ними следующая. Если мы говорим, что пойдёт дождь или снег, то имеем в виду, что может пойти дождь, может пойти снег, а может пойти и снег с дождём. А когда мы говорим, что пациент либо жив, либо мёртв, то подразумеваем, что пациент может быть жив, может быть мёртв, но он не может быть и жив, и мёртв одновременно. Следует учитывать тот факт, что очень часто, когда люди говорят „или”, они подразумевают исключающее „Или”. Если вас спрашивают: „Вам чай с молоком или без?”, то понятно, что речь идёт об исключающем „Или”. Но какое „Или” подразумевается во фразе „Вам чай или кофе?”

И ещё есть логическая эквиваленция. Она встречается, например, когда мы хотим сказать, что два слова являются синонимами. Можно сказать, что „я получаю зарплату“ и „я получаю жалованье“ — это одно и тоже. В математических текстах логическая эквивалентность проявляется в фразах „тогда и только тогда“, „если и только если“.

Таким образом, все союзы, при помощи которых мы образуем сложносочиненные и сложноподчиненные предложения, являются логическими связками, которые ещё называют логическими операторами, или логическими функциями. „Потому что“, „по той причине“, „из-за того, что“, „в следствии того, что“, „и“, „но“, „если“, „ни то, ни другое“, „когда“, „лишь“, „несмотря на то“, „хотя“ и многие другие — это всё логические связки.

Но если союзов очень много, то основных логических связок всего 6:

— отрицание — negation;

— логическое „И“, конъюнкция — logical “And”, conjunction;

— логическое „ИЛИ“, дизъюнкция — logical “Or”, inclusive “Or”, disjunction;

— исключающее „ИЛИ“ — exclusive “Or”;

— импликация — implication;

— логическая эквиваленция — logical biconditional, logical equality.

There are a lot of conjunctions, that is words used to connect clauses or sentences or to coordinate words in the same clause (for instance, and, but, if, because, since, and so on). But there are just six basic logical operations, which are used for constructing compound propositions:

— negation;

— logical “And”, which is also called conjunction;

— logical “Or”, which is also called disjunction;

— exclusive “Or”;

— implication;

— logical biconditional, logical equality.

### 1.1.6 Negation

Можно ли сказать, что предложение „Я не люблю фиалки“ состоит из элементарного высказывания „Я люблю филаки“ и логической связки „не“, которая называется логическим отрицанием. Если переписать предложение таким образом, то получим: „Не верно, что я люблю фиалки.“

Действительно ли мы не исказили смысл? Это достаточно тонкий вопрос, который зависит от нашей точки зрения. Если фразу „Я не люблю фиалки“ произнести достаточно экспрессивно, громко, то её можно понять как „испытываю отрицательные эмоции при виде фиалок“. В то же время, высказывание „Не верно, что люблю филаки“ может быть истинным, потому что человек равнодушен к фиалкам, а не потому, что испытываю отрицательные эмоции при их виде.

Рассмотрим еще два предложения: „Чайник долго закипает“ и „Чайник долго не закипает“. Является ли второе предложение отрицанием первого? Сказать сложно, особенно если принять во внимание тот факт, что смысл у этих двух предложений один и тот же.

Мы ещё раз убеждаемся в нелогичности человеческого языка, где из наличия отрицания может и не следовать само отрицание. Но для точного анализа подобное положение вещей недопустимо. Поэтому самым простым решением будет читать отрицание как „Не верно, что ...“

Отрицание в логике — унарная, то есть применимая к одному высказыванию, логическая операция над высказываниями, результатом которой является высказывание „противоположное“ исходному, то есть, отрицанием высказывания $p$ является высказывание, которое обозначается $\neg p$ и читается „не верно, что $p$“ или, для краткости, „не $p$“, при этом, если высказывание $p$ истинно, то $\neg p$ — ложно, а если $p$ — ложно, то $\neg p$ — истинно. Синоним: логическое „НЕ“.

**Definition 1.1.4** *Negation is a unary[5] logical operation on a proposition $p$, that produces the proposition 'It is not true, that $p$', which is denoted $\neg p$ (pronounced "not A"), and $\neg p$ is true when $p$ is false and $\neg p$ is false when $p$ is true.*

Логические операции удобно задавать в виде таблиц, подобным таблицам сложения и умножения. Такие таблицы называются таблицами истинности (truth table). Итак, приведём таблицу истинности для отрицания.

| truth value of $p$ | truth value of $\neg p$ |
|---|---|
| $True$ | $False$ |
| $False$ | $True$ |

or

| truth value of $p$ | truth value of $\neg p$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

где 1 обозначает $True$, а 0 — $False$.

В зависимости от контекста, можно встретить несколько разных способов обозначения отрицания:

$\neg p$ (not $p$), $\bar{p}$ ($p$ bar, bar $p$), $\tilde{p}$ (not $p$), $p'$ ($p$ prime, $p$ complement), $!p$ (bang $p$, not $p$).

**Glossary**

Truth value — истинностное значение

Negation — отрицание

Truth table — таблица истинности

[5] That is applicable to one proposition. Here 'uno' is 'one' from Latin.

### 1.1.7 Conjunction

Следующей достаточно простой логической операцией является конъюнкция[6], или, говоря более простым языком, логическое „И". Иногда коъюнкцию еще называют логическим умножением.

**Definition 1.1.5** *The conjunction* is a binary[7] logical operation of two propositions $p$ and $q$ that produces the proposition $p \wedge q$ which is true if and only if both propositions are true.

*Конъюнкция* — это бинарная (то есть применяемая к двум высказываниям) логическая операция, которая из двух данных высказываний $p$ и $q$ производит высказывание $p \wedge q$, которое истинно тогда и только тогда, когда оба высказывания истинны.

Задаётся конъюнкция при помощи следующей таблицы истинности:

| truth value of $p$ | truth value of $q$ | truth value of $p \wedge q$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

Встречается несколько обозначений конъюнкции: $p \wedge q$, $p\&q$, $p \cdot q$, $pq$.

Обозначение $\wedge$ происходит, возможно, от первой буквы слова „And": „$\bigwedge$nd".

Второе обозначение — от французского слова „*et*". Действительно, если наложить буквы „*e*" и „*t*" друг на друга[8], то получится знак &.

Последние два обозначения происходят от другого названия конъюнкции — „логического умножения". Кстати, почему конъюнкция называется логическим умножением?

Рассмотрим предложение „На улице идёт дождь, и я беру зонт".
Обозначим символом $p$ первую часть предложения — „На улице идёт дождь", а
символом $q$ вторую — „Я беру зонт".
Тогда данное предложение можно символически записать как $p \wedge q$.

А теперь рассмотрим похожее предложение „На улице идёт дождь, но я не беру зонт". Какой логической связке соответствует союз „но"? Оказывается, тоже конъюнкции. С точки зрения логики между союзами „и" и „но" нет никакой разницы, но если в сложном предложении вторая часть противопоставляется первой, начинается с отрицания, то вместо союза „и" говорят „но". Данное предложение можно записать как $p \wedge (\neg q)$.

Consider the following sentence—"It is raining and I am taking my umbrella."
Denote by the symbol $p$ the first part of the sentence—"It is raining", and
by the symbol $q$ the second part—"I am taking my umbrella".
Then we can write symbolically our sentence as $p \wedge q$.

In the sentence "It is raining but I am not taking my umbrella" we have another conjunction "but", which logically means "and", but is used in presence of a contrast or exception. So we may write this sentence as $p \wedge (\neg q)$.

**Glossary**

Conjunction — в логике — конъюнкция, логическое „И", а в литературе — союз.

---

[6]От латинского coniunctionem — союз, связь.

[7]i.e. operation that is applicable to two propositions (here i.e. is an abbreviation of "id est" ("it is"). It is usually spoken as "that is").

[8]Наложение одного знака или буквы на другой знак называется *лигатурой*.

### 1.1.8 Inclusive and exclusive logical “Or”

Следующими логическими операциями являются дизъюнкция[9], или, говоря более простым языком, логическое „Или“, и исключающее „Или“.

**Definition 1.1.6** *The disjunction* is a binary logical operation of two propositions $p$ and $q$ that produces the proposition $p \vee q$, which is true if and only if at least one of these propositions is true.

*Дизъюнкция* — это бинарная логическая операция, которая по двум высказываниям $p$ и $q$ даёт высказывание $p \vee q$, которое истинно тогда и только тогда, когда хотя бы одно высказывание из двух истинно.

**Definition 1.1.7** *The exclusive or* is a binary logical operation of two propositions $p$ and $q$ that produces the proposition $p \oplus q$, which is true if and only if exactly one of these propositions is true.

*Исключающее или* (*строгая дизъюнкция*) — это бинарная логическая операция, которая по двум высказываниям $p$ и $q$ даёт высказывание $p \oplus q$, которое истинно тогда и только тогда, когда ровно одно высказывание из двух истинно.

Приведём таблицу истинности для дизъюнкции и исключающего „Или“:

| truth value of $p$ | truth value of $q$ | truth value of $p \vee q$ | truth value of $p \oplus q$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

В русском языке исключающему „Или“ соответствуют обороты „или. . . , или. . . “ и „либо. . . , либо. . . “. Сам по себе союз „или“ может означать как дизъюнкцию, так и исключающее „Или“ в зависимости от ситуации.

Рассмотрим следующий фрагмент из сказки „Винни Пух и все, все, все“ А. Милна:

А когда Кролик спросил: „Тебе чего намазать — мёду или сгущённого молока?“ — Пух пришел в такой восторг, что выпалил: „И того, и другого!“

Несомненно, что Кролик имел в виду исключающее „Или“: „Намазать мёду“ $\oplus$ „намазать сгущённого молока“, в то время как Винни Пух понял союз „или“ как дизъюнкцию.

In English, the construct “either. . . or” is usually used to indicate exclusive or and “or” generally used for inclusive, but it is not a strict rule.

Recall the following fragment of “Winnie the Pooh” by A. Milne.

[. . . ] and when Rabbit said, “Honey or condensed milk with your bread?” he (Pooh) was so excited that he said, “Both!”

Obviously, Rabbit meant exclusive or: “Honey with your bread” $\oplus$ “condensed milk with your bread?”, but Winnie understood this “or” as inclusive or, as disjunction.

In the proposition $p \vee q$ the propositions $p$ and $q$ are called *disjuncts*.

In the proposition $p \wedge q$ the propositions $p$ and $q$ are called *conjuncts*.

**Glossary**

Exclusive “Or” — исключающее „Или“. Disjunction — дизъюнкция, логическое „Или“.

[9] От латинского disjunctionem — разделение, знак $\vee$ происходит от латинского слова „Vel“ — или.

### 1.1.9 Implication

Одной из наиболее важных в математике логических связок является импликация.

**Definition 1.1.8** *Implication* is a binary logical operation of two propositions $p$ and $q$ that produces the proposition $p \to q$, which is false if and only if $p$ is true and $q$ is false.

*Импликация* — это бинарная логическая операция, которая по двум высказываниям $p$ и $q$ даёт высказывание $p \to q$, которое ложно тогда и только тогда, когда $p$ истинно, а $q$ ложно.

Таблица истинности для импликации выглядит следующим образом:

| truth value of $p$ | truth value of $q$ | truth value of $p \to q$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

Запомнить эту таблицу можно при помощи такого объяснения:
— если вам что-то хорошее обещали и подарили, то вы довольны;
— если же обещали, но не подарили, то есть повод расстроиться;
— если не обещали, но, вдруг, подарили, то это приятно;
— а если и не обещали, и не подарили, то и причин расстраиваться нет.

Таким образом, расстроиться можно только в одном случае: когда обещали, но не подарили, то есть когда из истины следует ложь.

Another notation for implication is $p \supset q$. Other names of implication are the *material conditional*, *material implication*,

In the proposition $p \to q$ the left part $p$ is called the *hypothesis*, or the *antecedent* and the right part $q$ is called the *conclusion*, or the *consequent*.

В высказывании $p \to q$ левая часть называется посылкой, или гипотезой, а правая — заключением.

Существует множество способов выразить импликацию $p \to q$:
— если $p$, то $q$;
— $q$, если $p$;
— $p$ влечёт $q$;
— $q$ следует из $p$ и другие.

There are a lot of ways in order to express implication:
— if $p$ then $q$;
— $p$ if $q$;
— $p$ implies $q$;
— $q$ follows from $p$, and many others.

Implication can yield some unexpected truths when expressed in natural language, because any implication statement with a false hypothesis is true, for example the proposition "If 2+2 = 3 then 2+2 = 5" is true. In English this fact is used in the saying "When hell freezes over."[10] So the sentence "I'll pay him back when hell freezes over" is true even you never do it.

[10] На русском языке этой поговорке соответствует „Когда рак на горе свистнет". Таким образом, если вы скажете: "Я верну ему деньги, когда рак на горе свистнет", то будете правы, поскольку это выражение истинно независимо от того, вернёте вы деньги или нет.

### 1.1.10 Logical biconditional, or logical equality

Последней в ряду изучаемых нами логических операций стоит эквивалентность. Она похожа на равенство и истинна, когда два высказывания принимают одинаковые истинностные значения.

**Definition 1.1.9** *Biconditional* is a binary logical operation of two propositions $p$ and $q$ that produces the proposition $p \leftrightarrow q$, which is true if and only if the truth values of $p$ and $q$ are the same.

*Эквиваленция*, или *равнозначность* — это бинарная логическая операция, которая по двум высказываниям $p$ и $q$ даёт высказывание $p \leftrightarrow q$, которое истинно тогда и только тогда, когда истинностные значения высказываний $p$ и $q$ совпадают.

Таблица истинности для эквиваленции выглядит следующим образом:

| truth value of $p$ | truth value of $q$ | truth value of $p \leftrightarrow q$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

We pronounce $p \leftrightarrow q$ as "$p$ if and only if $q$", or "$p$ iff $q$" for short. The plain English "if" may sometimes be used as a biconditional. One should carefully consider the context. For example, "I will buy you a new smartphone if you need one" may be meant as a biconditional, since the speaker implicitly says "If you do not need a new smartphone, then I will not buy it". However, "it is cloudy if it is raining" is not meant as a biconditional, since it can be cloudy while not raining.

Also we can state a biconditional in plain English in the form "$p$ if $q$ and $q$ if $p$". Slightly more formally, one could say "$p$ implies $q$ and $q$ implies $p$". Below we construct the truth table explaining why we can express a biconditional in this way.

In this table we replace the phrase "truth value of $p$", which is quite long, with a shorter phrase "$p$".

| $p$ | $q$ | $p \to q$ | $q \to p$ | $(p \to q) \wedge (q \to p)$ | $p \leftrightarrow q$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 |

As you can see the truth values of $(p \to q) \wedge (q \to p)$ and $p \leftrightarrow q$ are the same.

Высказывание $p \leftrightarrow q$ можно прочитать, как „$p$ тогда и только тогда, когда $q$“ или, что то же самое, „$p$, если и только если $q$“. Кроме того, данное высказывание можно прочитать и как „$p$ влечёт $q$, и $q$ влечёт $p$“. Для того, чтобы понять, почему это возможно, надо обратиться к таблице истинности, приведённой выше. Видно, что столбцы истинностных значений у высказываний $(p \to q) \wedge (q \to p)$ и $p \leftrightarrow q$ совпадают.

**Glossary**

Biconditional — эквиваленция, равнозначность
Iff — if and only if — если и только если

### 1.1.11 Propositions and electrical circuits

From Physics we know that components of an electrical circuit can be connected in many different ways. The two simplest ways of these are called series and parallel.

Below we give two examples: of a series circuit and of a parallel circuit.

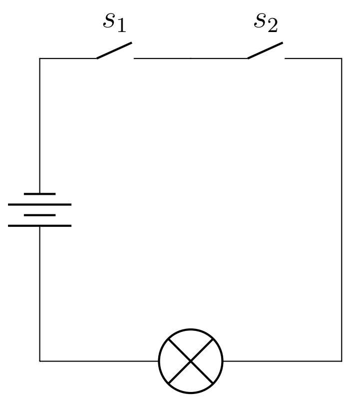


Figure 1.2: Series circuit

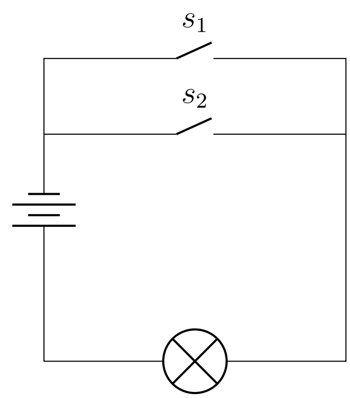


Figure 1.3: Parallel circuit

What is a relation between these electrical circuits and logical operations?

Let us answer the question: when the lamp is on?

First, we consider the series circuit on Figure 1.2: the lamp is on if and only if both switches $s_1$ and $s_2$ are closed.

Let $p_i$ stands for the proposition 'the switch $s_i$ is closed' for $i = 1, 2$.

Then the lamp is on if 'the switch $s_1$ is closed' and 'the switch $s_2$ is closed'.

But it is exactly the conjunction: $p_1 \wedge p_2$.

Second, we consider the parallel circuit on Figure 1.3: the lamp is on if and only if at least one of $s_1$ and $s_2$ is closed, that is when the disjunction $p_1 \vee p_2$ is true.

So, the series circuit corresponds to $p_1 \wedge p_2$, and the parallel circuit corresponds to $p_1 \vee p_2$.

Что общего между последовательным и параллельным соединением двух выключателей в электрических цепях и логическими операциями?

На рисунке 1.2 изображено последовательное подсоединение выключателей. Понятно, что лампочка будет гореть только тогда, когда оба выключателя замкнуты, то есть когда высказывание „выключатель $s_1$ замкнут и выключатель $s_2$ замкнут" истинно.

На рисунке 1.3 изображено параллельное подсоединение выключателей. Понятно, что лампочка будет гореть только тогда, когда хотя бы один выключатель замкнут, то есть когда высказывание „выключатель $s_1$ замкнут или выключатель $s_2$ замкнут" истинно.

Пусть $p_i$ обозначает высказывание „выключатель $s_i$ замкнут", где $i = 1, 2$.

Последовательное соединение выключателей $s_1$ и $s_2$ соответствует $p_1 \wedge p_2$.

Параллельное соединение выключателей $s_1$ и $s_2$ соответствует $p_1 \vee p_2$.

**Glossary**

Series circuit — последовательное подключение

Parallel circuit — параллельное подключение

### 1.1.12 Logic gates

As we have seen logical operations are used to model the circuitry of electronic devices. Each input and each output of such a device can be thought of as a bit, that is as a member of the set $\{0, 1\}$. A computer and other electronic devices are made up of circuits. Each circuit can be designed using the logical operations which we have studied.

Below we give the graphical representation of the basic logical operations. These pictures, which represent logical operations, are called *logic gates*.

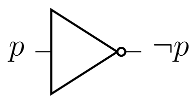


Figure 1.4: Inverter

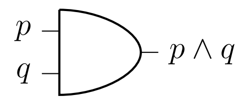


Figure 1.5: AND gate

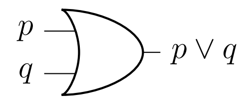


Figure 1.6: OR gate

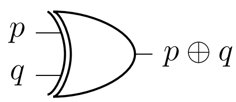


Figure 1.7: Exclusive OR gate

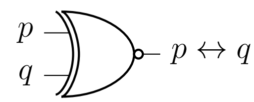


Figure 1.8: XNOR gate

Как мы видели, логические операции можно использовать для описания электрических устройств. Каждый вход и каждый выход такого устройства может быть рассмотрен как бит, то есть как элемент множества $\{0, 1\}$. Компьютер и другие электронные устройства сделаны из электрических цепей. Каждая цепь может быть описана при помощи логических операций, которые мы изучали.

Выше были представлены графические представления основных логических операций, которые называются *логическими вентилями*.

We can combine gates in order to describe a compound propositions.

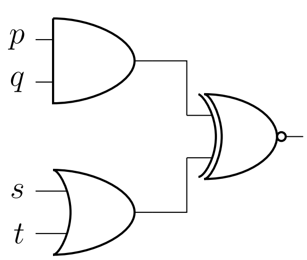


Figure 1.9: $(p \wedge q) \leftrightarrow (s \vee t)$

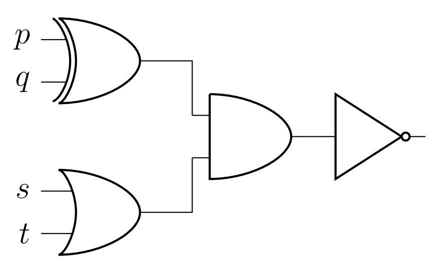


Figure 1.10: $\neg((p \oplus q) \wedge (s \vee t))$

Для того, чтобы описать составные высказывания, мы можем соединять логические вентили, как показано на рисунках выше.

**Glossary**

Logic gate — логический вентиль
Describe — описывать

### 1.1.13 The number of rows in a truth table

Before we start to construct truth tables for various propositions, we have to be sure, that a given thruth table is complete, i.e. consists of all possible cases for elementary propositions it depends on. So it is necessary to calculate the number of all possible cases.

Since we denote elementary propositions by letters $p$, $q$, $r$, …, we can think of these letters as of variables, which we call *propositional variables*, or *propositional letters*.

Recall, that a propositional variable may have just 2 values: 0 and 1.

For two propositional letters $p$ and $q$ there are 4 possible cases:

$$\begin{aligned} p = 1 \quad &\text{and} \quad q = 1, \\ p = 1 \quad &\text{and} \quad q = 0, \\ p = 0 \quad &\text{and} \quad q = 1, \\ p = 0 \quad &\text{and} \quad q = 0. \end{aligned}$$

For three propositional letters $p$, $q$ and $r$ there are 4 possible case for $p$ and $q$ when $r = 1$ and there are 4 possible case for $p$ and $q$ when $r = 0$. So totally we have 8 possible cases.

We can add one more propositional letter $s$: there are 8 possible case for $p$, $q$ and $r$ when $s = 1$ and there are 8 possible case for $p$, $q$ and $r$ when $s = 0$. So totally we have 16 possible cases for 4 propositional letters.

Continuing reasoning in this way we obtain

**Theorem 1.1.10** *The number of all possible cases for $n$ propositional letters is equal to $2^n$, or, that is the same, any truth table in $n$ propositional letters consists of $2^n$ rows.*

*Proof.* For small $n$'s we already checked the statement of the theorem.

We postpone proving this theorem for each $n$ untill we study mathematical induction. □

Прежде, чем мы начнём строить таблицы истинности, было бы хорошо удостовериться в том, что данная таблица истинности полна, содержит всю информацию, то есть в ней рассмотрены все возможные случаи для используемых элементарных высказываний.

Буквы $p$, $q$, $r$, …, которые мы используем для обозначения элементарных высказываний, также называются *пропозициональными буквами*, или *пропозициональными переменными*.

Вспомним, что пропозициональная буква принимает только два значения: 0 и 1.

Две пропозициональные переменные могут принимать уже 4 значения, все возможные случае приведены на этой странице выше.

Для трёх пропозициональных букв $p$, $q$ и $r$ существует 4 возможных случая для $p$ и $q$, когда $r = 1$, и 4 возможных случая для $p$ и $q$, когда $r = 0$. Итого, у нас есть 8 случаев.

Продолжая наши рассуждения, мы получаем следующую теорему.

**Теорема 1.1.10** *Число всех возможных случаев для $n$ пропозицональных переменных равно $2^n$, или, что тоже самое, число строк в таблице истинности с $n$ пропозициональными переменными равно $2^n$.*

*Доказательство.* Для небольших значения $n$ теорему мы уже доказали.

Доказательство для всех значений $n$ мы отложим до тех пор, пока не изучим математическую индукцию. □

**Glossary**

Propositional letter — пропозициональная буква

Propositional variable — пропозициональная переменная

### 1.1.14 Some agreements and remarks

It does not matter in which order we put all possible variants of elementary propositions in a truth table. All of the following truth table are the same.

| $p$ | $q$ | $p \to q$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

| $p$ | $q$ | $p \to q$ |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| $p$ | $q$ | $p \to q$ |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |

| $p$ | $q$ | $p \to q$ |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |

But if everybody use the same ordering of all possible cases of elementary propositions, then it is easier to compare the results.

Не имеет значения в каком порядке расположены все возможные случаи для элементарных высказываний, все четыре таблицы истинности, приведённые выше, на самом деле, одинаковые.

Но если все используют один и тот же порядок расположения всех возможных случаев для элементарных высказываний, то сравнение таблиц истинности происходит гораздо быстрее и легче.

The order of operations (or operator precedence) is a convention for the order in which logical operations are performed when parentheses are omitted.

For instance, we understand that $2 + 2 \times 3 = 8$, because we do multiplication first.

The order of logical operation is the following: 1) negation; 2) conjunction and disjunction; 3) exclusive or; 4) implication; 5) biconditional.

For example, $p \wedge q \to \neg s \vee r$ stands for $(p \wedge q) \to ((\neg s) \vee r)$.

One more rule: $p \to q \to r \to s$ stands for $p \to (q \to (r \to s))$.

Note that there is a problems with $\wedge$ and $\vee$. Some people think that they have the same precedence, some people consider $\wedge$ to have a higher order, while others believe that $\vee$ has a higher order than $\wedge$.

So it is better not to omit the parentheses between $\wedge$ and $\vee$.

Порядок операций — это соглашение о том, в каком порядке выполняются операции при отсутствии скобок.

Например, мы понимаем, что $3 + 4 \times 5 - 6 = 17$, потому что умножение выполняется в первую очередь.

Точно так же есть своя очередность и у логических операций. При опускании скобок в высказываниях следует учитывать приоритет операций: 1) наивысший у отрицания; 2) затем идут $\wedge$ и $\vee$; 3) исключающее или; 4) импликация; 5) эквиваленция.

Например, под $p \wedge q \leftrightarrow t \to \neg s \vee r$ имеется в виду $(p \wedge q) \leftrightarrow (t \to ((\neg s) \vee r))$.

Еще одно правило: $p \to q \to r \to s$ обозначает $p \to (q \to (r \to s))$.

Заметим, что есть небольшая проблема с $\wedge$ и $\vee$. Некоторые люди считают, что они имеют одинаковый приоритет, другие — что $\wedge$ имеет более высокий приоритет, в то время как есть и те, которые верят, что приоритет у $\vee$ выше, чем у $\wedge$.

**Glossary**

Operator precedence — порядок (приоритет) операций

### 1.1.15 Constructing thruth tables. A tautology, a contradiction

Since we have defined the basic logical operations and its truth tables we can construct the truth table for any compound proposition.

Consider the proposition $\neg(p \to (q \to p))$. In order to calculate this we calculate $q \to p$, then $p \to (q \to p)$, and finally $\neg(p \to (q \to p))$.

| $p$ | $q$ | $q \to p$ | $p \to (q \to p)$ | $\neg(p \to (q \to p))$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |

The second example is the following: $((\neg q) \to (\neg p)) \to (p \to q)$.

Slightly more convenient would be to write the value of a logical operation under the symbol of this logical operation.

| $p$ | $q$ | $((\neg q) \to (\neg p)) \to (p \to q)$ |
|---|---|---|
| 1 | 1 | 0 1 0 **1** 1 |
| 1 | 0 | 1 0 0 **1** 0 |
| 0 | 1 | 0 1 1 **1** 1 |
| 0 | 0 | 1 1 1 **1** 1 |

We can see that the propositions $p \to (q \to p)$ and $((\neg q) \to (\neg p)) \to (p \to q)$ have the truth value 1, while the proposition $\neg(p \to (q \to p))$ has the truth value 0 for any truth values of $p$ and $q$.

Such propositions have special names.

**Definition 1.1.11** A compound proposition is called a *tautology* if it has the truth value 1 for each truth value of elementary proposition which it contains.

Roughly speaking, a proposition is called a *tautology*, if it is always true.

**Definition 1.1.12** A compound proposition is called a *contradiction* if it has the truth value 0 for each truth value of elementary proposition which it contains.

Roughly speaking, a proposition is called a *contradiction*, if it is always false.

Выше мы привели примеры как составлять таблицу истинности для различных высказываний, причём, для любых значений аргументов высказывания $p \to (q \to p)$ и $((\neg q) \to (\neg p)) \to (p \to q)$ принимают только значение 1, а высказывание $\neg(p \to (q \to p))$ — только значение 0. У таких высказываний есть специальное имя.

**Определение 1.1.11** Составное высказывание называется *тавтологией*, если оно принимает истинностное значение 1 при любых истинностных значениях входящих в него элементарных высказываний.

Грубо говоря, высказывание называется *тавтологией*, если оно всегда истинно.

**Определение 1.1.12** Составное высказывание называется *противоречием*, если оно принимает истинностное значение 0 при любых истинностных значениях входящих в него элементарных высказываний.

Грубо говоря, высказывание называется *противоречием*, если оно всегда ложно.

**Glossary**

Tautology — тавтология    Contradiction — противоречие

### 1.1.16 Constructing thruth tables. Logical equivalence

Consider the following propositions: $(p \to q)$, $(\neg p) \vee q$, and $(p \to q) \leftrightarrow ((\neg p) \vee q)$.

| $p$ | $q$ | $p \to q$ | $\neg p$ | $(\neg p \vee q)$ | $(p \to q) \leftrightarrow ((\neg p) \vee q)$ |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |

We can see that the propositions $p \to q$ and $((\neg p) \vee q)$ have the same truth tables.

In mathematics equality of the values of two expressions for any instances of variables is called an *identity*, for example $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$.

In logic the same property, that two propositions have the same truth tables, that is have the same values for each instances of variables, is called *logical equivalence*.

In order to distinguish elementary (or, that is the same, atomic) propositions, denoted by propositional variables $p$, $q$, ..., $r$, from compound propositions, we denote compound propositions by capital letters $P$, $Q$, ..., $R$.

**Definition 1.1.13** Two propositions $P$ and $Q$ are *logically equivalent*, if they have the same truth tables. We denote this by $P \Leftrightarrow Q$.

Note that since the truth tables of $(p \to q)$ and $(\neg p) \vee q$ are the same, the proposition $(p \to q) \leftrightarrow ((\neg p) \vee q)$ is a tautology. It is immediate from definition of $\leftrightarrow$.

So, the following theorem is obvious.

**Theorem 1.1.14** *Two propositions $P$ and $Q$ are* logically equivalent *if and only if $P \leftrightarrow Q$ is a tautology.*

Выше мы построили таблицу истинности для высказываний $(p \to q)$, $(\neg p) \vee q$, и $(p \to q) \leftrightarrow ((\neg p) \vee q)$. Как мы видим, высказывания $(p \to q)$ и $(\neg p) \vee q$ имеют одинаковые таблицы истинности.

В математике тот факт, что два выражения имеют одинаковые значению при любых значениях переменных, называется тождеством. Например, $\sin^2 x + \cos^2 x = 1$.

В логике то же самое свойство, что два высказывания имеют одинаковые таблицы истинности, то есть принимают одни и те же истинностные значения при любых значениях пропозициональных переменных, называется *логической эквивалентностью.*

Чтобы отличать элементарные высказывания, обозначаемые пропозициональными переменными $p$, $q$, ..., $r$, от составных высказываний, мы будем обозначать последние при помощи заглавных букв $P$, $Q$, ..., $R$.

**Определение 1.1.13** Два высказывания $P$ и $Q$ являются *логически эквивалентными*, если они имеют одинаковые таблицы истинности.

Мы будем обозначать этот факт следующим образом: $P \Leftrightarrow Q$.

Как видно из примера выше, если два высказывания имеют одинаковые таблицы истинности, то их эквиваленция будет тавтологией. Таким образом, из определения операции $\leftrightarrow$ непосредственно вытекает следующая теорема.

**Теорема 1.1.14** *$P \Leftrightarrow Q$ тогда и только тогда, когда $P \leftrightarrow Q$ — тавтология.*

**Glossary**

Logical equivalence — логическая эквивалентность

### 1.1.17 Logical laws

Some of logical equivalences are of great importance, so we call them *logical laws*.

Особо важные логические эквивалентности называют *логическими законами*.

| Name | Equivalence |
|---|---|
| Identity laws<br>Законы сокращения 0 и 1 | $P \wedge 1 \Leftrightarrow P$<br>$P \vee 0 \Leftrightarrow P$ |
| Domination laws<br>Законы преобладания 0 и 1 | $P \wedge 0 \Leftrightarrow 0$<br>$P \vee 1 \Leftrightarrow 1$ |
| Idempotent laws<br>Законы идемпотентности | $P \wedge P \Leftrightarrow P$<br>$P \vee P \Leftrightarrow P$ |
| Double negation law<br>Закон двойного отрицания | $\neg(\neg P) \Leftrightarrow P$ |
| Commutative laws<br>Законы коммутативности | $P \wedge Q \Leftrightarrow Q \wedge P$<br>$P \vee Q \Leftrightarrow Q \vee P$ |
| Associative laws<br>Законы ассоциативности | $(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$<br>$(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$ |
| Distributive laws<br>Законы дистрибутивности | $P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$<br>$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$ |
| De Morgan's laws<br>Законы де Моргана | $\neg(P \wedge Q) \Leftrightarrow (\neg P) \vee (\neg Q)$<br>$\neg(P \vee Q) \Leftrightarrow (\neg P) \wedge (\neg Q)$ |
| Absorption laws<br>Законы поглощения | $P \wedge (P \vee Q) \Leftrightarrow P$<br>$P \vee (P \wedge Q) \Leftrightarrow P$ |
| Negation laws<br>Законы для отрицания | $P \wedge (\neg P) \Leftrightarrow 0$<br>$P \vee (\neg P) \Leftrightarrow 1$ |

All logical laws can be checked by truth tables. Sometimes it is not necessary to compute the whole truth table explicitly, we apply some reasoning to understand how it looks like.

Let us check the associative law $(P \wedge Q) \wedge R \Leftrightarrow P \wedge (Q \wedge R)$.

Consider the left part: $(P \wedge Q) \wedge R$ is true if and only if $(P \wedge Q)$ is true and $R$ is true;
$(P \wedge Q)$ is true if and only if both $P$ and $Q$ are true;
So, $(P \wedge Q) \wedge R$ is true if and only if each of $P$, $Q$, and $R$ is true.

Consider the right part: $P \wedge (Q \wedge R)$ is true iff $P$ is true and $(Q \wedge R)$ is true;
$(Q \wedge R)$ is true iff both $Q$ and $R$ are true;
So, $P \wedge (Q \wedge R)$ is true iff each of $P$, $Q$, and $R$ is true.

Thus, the truth table of these propositions has the truth value 1 just in the row where each of $P$, $Q$, and $R$ are true, and the truth value 0 in all other rows. So, these propositions are equivalent.

Все логические законы могут быть проверены при помощи таблиц истинности. Однако не всегда необходимо явно строить таблицу истинности, иногда можно понять как она выглядит при помощи простых рассуждений.

Например, $P \vee Q$ ложно тогда и только тогда, когда и $P$, и $Q$ ложны. Но то же самое можно сказать и о $Q \vee P$, следовательно, эти высказывания эквивалентны.

**Glossary**

Logical law — логический закон  Iff stands for 'if and only if' — если и только если

### 1.1.18 Cheking logical laws

We continue checking some logical laws.

Below we construct the truth tables for $p \wedge (q \vee r)$ and $(p \wedge q) \vee (p \wedge r)$ as well as for $p \vee (q \vee r)$ and $(p \vee q) \vee r$.

| $p$ | $q$ | $r$ | $p \wedge (q \vee r)$ | $(p \wedge q) \vee (p \wedge r)$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 $_1$ | $_1$ 1 $_1$ |
| 1 | 1 | 0 | 1 $_1$ | $_1$ 1 $_0$ |
| 1 | 0 | 1 | 1 $_1$ | $_0$ 1 $_1$ |
| 1 | 0 | 0 | 0 $_0$ | $_0$ 0 $_0$ |
| 0 | 1 | 1 | 0 $_1$ | $_0$ 0 $_0$ |
| 0 | 1 | 0 | 0 $_1$ | $_0$ 0 $_0$ |
| 0 | 0 | 1 | 0 $_1$ | $_0$ 0 $_0$ |
| 0 | 0 | 0 | 0 $_0$ | $_0$ 0 $_0$ |

| $p$ | $q$ | $r$ | $p \vee (q \vee r)$ | $(p \vee q) \vee r$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 $_1$ | $_1$ 1 |
| 1 | 1 | 0 | 1 $_1$ | $_1$ 1 |
| 1 | 0 | 1 | 1 $_1$ | $_1$ 1 |
| 1 | 0 | 0 | 1 $_0$ | $_1$ 1 |
| 0 | 1 | 1 | 1 $_1$ | $_1$ 1 |
| 0 | 1 | 0 | 1 $_1$ | $_1$ 1 |
| 0 | 0 | 1 | 1 $_1$ | $_0$ 1 |
| 0 | 0 | 0 | 0 $_0$ | $_0$ 0 |

As we can see, $p \wedge (q \vee r)$ and $(p \wedge q) \vee (p \wedge r)$ are equivalent, as well as $p \vee (q \vee r)$ and $(p \vee q) \vee r$.

Выше мы построили таблицы истинности и убедились, что $p \wedge (q \vee r)$ и $(p \wedge q) \vee (p \wedge r)$ эквивалентны, так же как и $p \vee (q \vee r)$ и $(p \vee q) \vee r$.

Also, in order to check some laws we can use another laws, provided that these laws have been already checked.

Consider the absorption law: $p \wedge (p \vee q) \Leftrightarrow p$.

Assume, $q$ is 0. Then $p \wedge (p \vee q) \Leftrightarrow p \wedge (p \vee 0)$, because we substitute 0 for $q$;
$p \wedge (p \vee 0) \Leftrightarrow p \wedge p$, by the identity law $(p \vee 0) \Leftrightarrow p$;
$p \wedge p \Leftrightarrow p$, by the idempotent law.

Assume, $q$ is 1. Then $p \wedge (p \vee q) \Leftrightarrow p \wedge (p \vee 1)$, because we substitute 1 for $q$;
$p \wedge (p \vee 1) \Leftrightarrow p \wedge 1$, by the domination law $(p \vee 1) \Leftrightarrow 1$;
$p \wedge 1 \Leftrightarrow p$, by the identity law.

Since for each truth value of $q$ the propositions $p \wedge (p \vee q)$ and $p$ are equivalent, so they are equivalent.

Для того, чтобы доказать эквивалентность двух высказываний можно использовать и ранее доказанные эквивалентности вместе с перебором всевозможных вариантов.

Докажем закон дистрибутивности: $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$.

Пусть $p = 1$. Рассмотрим сначала левую, а потом правую части.
Тогда $p \wedge (q \vee r) \Leftrightarrow 1 \wedge (q \vee r)$, потому что мы подставили 1 вместо $p$;
$1 \wedge (q \vee r) \Leftrightarrow q \vee r$, в силу закона сокращения 1: $(1 \wedge P) \Leftrightarrow P$.
Кроме того, $(p \wedge q) \vee (p \wedge r) \Leftrightarrow (1 \wedge q) \vee (1 \wedge r)$, потому что $p$ есть 1;
$(1 \wedge q) \vee (1 \wedge r) \Leftrightarrow q \vee r$, в силу закона сокращения 1.
Левая и правая части эквивалентны $q \vee r$, следовательно, эквивалентны друг другу.

Пусть $p = 0$. Рассмотрим сначала левую, а потом правую части.
Тогда $p \wedge (q \vee r) \Leftrightarrow 0 \wedge (q \vee r)$, потому что мы подставили 0 вместо $p$;
$0 \wedge (q \vee r) \Leftrightarrow 0$, в силу законов $(0 \wedge P) \Leftrightarrow (P \wedge 0) \Leftrightarrow 0$.
Кроме того, $(p \wedge q) \vee (p \wedge r) \Leftrightarrow (0 \wedge q) \vee (0 \wedge r)$, потому что $p$ есть 0;
$(0 \wedge q) \vee (0 \wedge r) \Leftrightarrow 0 \vee 0$, в силу закона преобладания 0;
$0 \vee 0 \Leftrightarrow 0$ в силу определения дизъюнкции.
Левая и правая части эквивалентны 0, следовательно, эквивалентны друг другу.

### 1.1.19 Constructing a proposition for a given truth table

Now we can construct the truth table for any given proposition.

Consider the opposite question: given a truth table how can we write a proposition?

For example, what is $F(p, q, r)$, which corresponds to the following truth table.

| $p$ | $q$ | $r$ | $F(p,q,r)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |

A standard mathematical trick is that if we cannot solve a problem we divide this problem into several smaller problems, which are easier to solve.

In our example the truth table of $F$ contains four 1's (and four 0's). We divide the problem of finding $F$ into four smaller problems: we are going to find four propositions $F_1$, $F_2$, $F_3$, and $F_4$, such that each of them contains exactly one 1.

In order to explain this we show the truth tables for the propositions $F_1$, $F_2$, $F_3$, and $F_4$.

| $p$ | $q$ | $r$ | $F(p,q,r)$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

What is $F_1$? We know that a conjunction is true if and only if each conjunct is true.

So, $F_1 = p \wedge q \wedge r$.

Under what conditions $F_2$ is true? The proposition $F_2$ is true when $p$ and $q$ are true, but $r$ is false. So, we apply negation to $r$ in order to make it to be true.

Hence, $F_2 = p \wedge q \wedge \neg r$. Similarly, $F_3 = \neg p \wedge q \wedge r$ and $F_4 = \neg p \wedge \neg q \wedge r$.

Now we can obtain each 1 separately. It is necessary to combine all of these 1's.

We know that disjunction is true if at least one disjunct is true.

Thus, $F = (p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge r)$.

Как по таблице истинности построить высказывание?

Выше приведена таблица истинности для неизвестного высказывания $F$. Поскольку сразу не понтяно, как искать $F$, мы разобьём нашу задачу на более простые подзадачи: мы будем искать $F_1$, $F_2$, $F_3$, и $F_4$, у которых лишь один раз встречается значение 1.

Очевидно, что $F_1 = p \wedge q \wedge r$.

Как построить $F_2$? Смотрим, что $F_2$ истинно, когда $p$ и $q$ истинны, но $r$ ложно. Но если $r$ ложно, то $\neg r$ истинно. Значит, $F_2 = p \wedge q \wedge \neg r$, а $F_3 = \neg p \wedge q \wedge r$, $F_4 = \neg p \wedge \neg q \wedge r$.

Применяем дизъюнкцию, чтобы объединить все единицы, и получаем $F$.

### 1.1.20 Constructing a proposition, the second way

Now we consider another way of constructing a proposition.

In the first way we have separated 1's. Here we separate 0's.

Look to the next truth tables. If $G_i$ contains just one 0, where $i = 1, 2, 3, 4$.

| $p$ | $q$ | $r$ | $F(p,q,r)$ | $G_1$ | $G_2$ | $G_3$ | $G_4$ |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

We start from $G_4$. We know that disjunction is false iff all disjuncts are false. Then

$$G_4 = p \vee q \vee r$$

The proposition $G_3$ is false when $q$ is true, so $G_3 = p \vee \neg q \vee r$.

Similarly, $G_2 = \neg p \vee q \vee r$ and $G_1 = \neg p \vee q \vee \neg r$.

Now we combine all 0's: we know that conjunction is false if at least one conjunct is false.

$$F = G_1 \wedge G_2 \wedge G_3 \wedge G_4 = (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee r)$$

Рссмотрим следующую таблицу истинности:

| $p$ | $q$ | $r$ | $H(p,q,r)$ | $H_1$ | $H_2$ | $H_3$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 |

Мы хотим написать высказывание $H$.

Как и в первом методе, мы делим задачу на следующие подзадачи: нам надо сначала написать три высказывания $H_1$, $H_2$ и $H_3$, у каждого из которых лишь в одном случае истинностное значение 0, а потом мы объединим эти нули при помощи конъюнкции.

Проще всего написать $H_3$, потому что только дизъюнкция ложно только в одном случае, когда все пропозициональные символы ложны.

Итого, $H_3 = p \vee q \vee r$.

При написание $H_2$ следует иметь в виду, что оно ложно, когда $q$ и $r$ истинны, значит, перед ними надо поставить отрицание: $H_2 = p \vee \neg q \vee \neg r$.

Точно так же пишем $H_1$. Оно равно $H_1 = \neg p \vee \neg q \vee \neg r$.

Теперь объединяем их вместе и получаем

$$H = H_1 \wedge H_2 \wedge H_3 = (\neg p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee q \vee r)$$

### 1.1.21 Disjunctive and conjunctive normal forms

In this chapter we give a formal definition for two methods of constructing proposotions by given truth tables, described in the previous chapter.

**Definition 1.1.15** A *literal* is either a propositional variable, or its negation.

**Definition 1.1.16** An *elementary conjunction* is a conjunctions of literals.

An *elementary disjunction* is a disjunctions of literals.

A *clause* is either an elementary conjunction, or an elementary disjunction.

So, $p \wedge q \wedge \neg r$ is an elementary conjunction, while $\neg(p \wedge q)$ is not.

**Definition 1.1.17** A proposition is said to be in a *disjunctive normal form*, DNF for short, if it is a disjunction of elementary conjunctions

A proposition is said to be in a *conjunctive normal form*, CNF for short, if it is a conjunction of elementary disjunctions

Examples of DNF are $(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r)$ and $(p \wedge q \wedge r) \vee (p \wedge \neg r) \vee \neg p$.

Examples of CNF are $(\neg p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee q \vee r)$ and $(\neg p \vee \neg r) \wedge (\neg q \vee \neg r) \wedge r$.

**Definition 1.1.18** A DNF formula is in *full disjunctive normal form* if each of its variables appears exactly once in every clause.

A CNF formula is in *full conjunctive normal form* if each of its variables appears exactly once in every clause.

The first example of DNF is in full DNF, but the second one is not.

The first example of CNF is in full CNF, but the second one is not.

**Определение 1.1.16** Под *элементарной конънкцией* мы понимаем конъюнкцию пропозициональных букв (элементарных высказываний) и их отрицаний.

Под *элементарной дизънкцией* мы понимаем дизъюнкцию пропозициональных букв (или, что то же самое, пропозициональных переменных) и их отрицаний.

**Определение 1.1.17** Говорят, что высказывание находится в *дизъюнктивной нормальной форме* (ДНФ), если оно является дизъюнкцией элементарных конъюнкций.

Говорят, что высказывание находится в *конъюнктивной нормальной форме* (КНФ), если оно является конъюнкцией элементарных дизъюнкций.

**Определение 1.1.18** Дизъюнктивная нормальная форма называется *соверешнной*, если каждая элементарная конъюнкция содержит каждую пропозициональную букву ровно один раз.

Конъюнктивная нормальная форма называется *соверешнной*, если каждая элементарная дизъюнкция содержит каждую пропозициональную букву ровно один раз.

Совершенная ДНФ: $(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r)$.

Совершенная КНФ: $(\neg p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee q \vee r)$.

Несовершенная ДНФ: $(p \wedge q \wedge r) \vee (p \wedge \neg r) \vee \neg p$.

Несовершенная КНФ: $(\neg p \vee \neg r) \wedge (\neg q \vee \neg r) \wedge r$.

## 1.1.22 Algorithms of constructing DNF and CNF

Writing a DNF:

— we choose rows where the unknown proposition is equal to 1;

— for each chosen row we write the parentheses of the form $({}^{\square}p_1 \wedge {}^{\square}p_2 \wedge \ldots \wedge {}^{\square}p_n)$, where we write $p_1$ if $p_1 = 1$ in this row;
$\neg p_1$ if $p_1 = 0$ in this row;

— we write $\vee$ between each pair of parentheses.

| | $p$ | $q$ | $r$ | $F(p,q,r)$ | |
|---|---|---|---|---|---|
| | 1 | 1 | 1 | 0 | |
| ✓ | 1 | 1 | 0 | 1 | $(p \wedge q \wedge \neg r)$ |
| | 1 | 0 | 1 | 0 | |
| | 1 | 0 | 0 | 0 | |
| ✓ | 0 | 1 | 1 | 1 | $(\neg p \wedge q \wedge r)$ |
| | 0 | 1 | 0 | 0 | |
| ✓ | 0 | 0 | 1 | 1 | $(\neg p \wedge \neg q \wedge r)$ |
| | 0 | 0 | 0 | 1 | |

So, we obtain that $F$ is
$(p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge r)$

Writing a CNF:

— we choose rows where the unknown proposition is equal to 0;

— for each chosen row we write the parentheses of the form $({}^{\square}p_1 \vee {}^{\square}p_2 \vee \ldots \vee {}^{\square}p_n)$, where we write $p_1$ if $p_1 = 0$ in this row;
$\neg p_1$ if $p_1 = 1$ in this row;

— we write $\wedge$ between each pair of parentheses.

| | $p$ | $q$ | $r$ | $F(p,q,r)$ | |
|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | |
| | 1 | 1 | 0 | 1 | |
| ✓ | 1 | 0 | 1 | 0 | $(\neg p \vee q \vee \neg r)$ |
| | 1 | 0 | 0 | 1 | |
| | 0 | 1 | 1 | 1 | |
| ✓ | 0 | 1 | 0 | 0 | $(p \vee \neg q \vee r)$ |
| ✓ | 0 | 0 | 1 | 0 | $(p \vee q \vee \neg r)$ |
| | 0 | 0 | 0 | 1 | |

So, we obtain that $F$ is
$(\neg p \vee q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r)$

Написание ДНФ:

— выбираем те строки, где значение искомого высказывания равно 1;

— для каждой выбранной строки мы пишем скобки вида $({}^{\square}p_1 \wedge {}^{\square}p_2 \wedge \ldots \wedge {}^{\square}p_n)$, где мы пишем $p_1$, если $p_1 = 1$ в этой строке;
$\neg p_1$, если $p_1 = 0$ в этой строке;

— пишем $\vee$ между скобками.

Написание КНФ:

— выбираем те строки, где значение искомого высказывания равно 0;

— для каждой выбранной строки мы пишем скобки вида $({}^{\square}p_1 \vee {}^{\square}p_2 \vee \ldots \vee {}^{\square}p_n)$, где мы пишем $p_1$, если $p_1 = 0$ в этой строке;
$\neg p_1$, если $p_1 = 1$ в этой строке;

— пишем $\wedge$ между скобками.

### 1.1.23 Minimization of DNF: Introduction

Assume that a truth table contains 10 propositional variables. Then the number of rows is $2^{10} = 1024$. If a given truth table contains, say, 500 ones, the full DNF consists of 500 elementary conjunctions, each elementary disjunction contains 10 propositional letters, 9 conjunctions, some number of negations and 2 parentheses. So, the full DNF consists of more than 10,000 symbols.

That is why the question 'how can we minimize a DNF (or a CNF)' is very important.

Предположим, что таблица истинности содержит 16 пропозициональных переменных. Тогда число строк в ней равно $2^{16} = 65536$. Если в этой таблице истинности 10000 единиц, то совершенная ДНФ состоит из 10000 элементарных конъюнкций, каждая элементарная конъюнкция содержит 16 пропозициональных букв, 15 конъюнкций, некоторое количество отрицаний и 2 скобки. Таким образом, длина СДНФ составляет как минимум 330.000 символов.

Поэтому вопрос „как минимизировать ДНФ (или КНФ)?” является очень важным.

In order to minimize a DNF we do a series of sequential steps of applications of the following two logical laws:

$$(F \wedge r) \vee (F \wedge \neg r) \Leftrightarrow F \text{ and } F \vee (F \wedge G) \Leftrightarrow F$$

Для минимизации ДНФ мы будем последовательно применять два вышеуказанных закона.

Let us check these laws (проверим эти законы):

$$\begin{aligned}
(F \wedge r) \vee (F \wedge \neg r) &\Leftrightarrow F \wedge (r \vee \neg r), && \text{because of the distributive law;}\\
F \wedge (r \vee \neg r) &\Leftrightarrow F \wedge 1, && \text{because of the law } r \vee \neg r \Leftrightarrow 1;\\
F \wedge 1 &\Leftrightarrow F, && \text{because of the identity law } (F \wedge 1) \Leftrightarrow F.
\end{aligned}$$

$$\begin{aligned}
F \vee (F \wedge G) &\Leftrightarrow (F \wedge 1) \vee (F \wedge G), && \text{because of the identity law } (F \wedge 1) \Leftrightarrow F;\\
(F \wedge 1) \vee (F \wedge G) &\Leftrightarrow F \wedge (1 \vee G), && \text{because of the distributive law;}\\
F \wedge (1 \vee G) &\Leftrightarrow F \wedge 1, && \text{because of the domination law } (1 \vee G) \Leftrightarrow 1;\\
F \wedge 1 &\Leftrightarrow F, && \text{because of the identity law } (F \wedge 1) \Leftrightarrow F.
\end{aligned}$$

Often, we use another form of the law $(F \wedge r) \vee (F \wedge \neg r) \Leftrightarrow F$:

$$(F_1 \wedge r \wedge F_2) \vee (F_1 \wedge \neg r \wedge F_2) \Leftrightarrow F_1 \wedge F_2$$

Indeed, $(F_1 \wedge r \wedge F_2) \vee (F_1 \wedge \neg r \wedge F_2) \Leftrightarrow$

$$\begin{aligned}
&\Leftrightarrow (F_1 \wedge F_2 \wedge r) \vee (F_1 \wedge F_2 \wedge \neg r), && \text{because of the commutative law;}\\
&\Leftrightarrow ((F_1 \wedge F_2) \wedge r) \vee ((F_1 \wedge F_2) \wedge \neg r), && \text{because of the associative law;}\\
&\Leftrightarrow F_1 \wedge F_2, && \text{as we proved above.}
\end{aligned}$$

For instance (например):

$$(\underbrace{p \wedge \neg q \wedge \neg r}_{F_1} \wedge s \wedge \underbrace{t \wedge \neg u}_{F_2}) \vee (\underbrace{p \wedge \neg q \wedge \neg r}_{F_1} \wedge \neg s \wedge \underbrace{t \wedge \neg u}_{F_2}) \Leftrightarrow (\underbrace{p \wedge \neg q \wedge \neg r}_{F_1} \wedge \underbrace{t \wedge \neg u}_{F_2})$$

#### Glossary

Sequential — последовательный

Indeed — действительно

Because of — из-за, по причине, вследствие

### 1.1.24 Minimization of DNF: an example

Consider the following full DNF (рассмотрим следующую СДНФ):

$$(p \wedge q \wedge r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge \neg r)$$

Here we use another notation for the negation: we write $\bar{p}$ for $\neg p$, because for DNF we obtain a shorter notation. We rewrite the above proposition:

$$(p \wedge q \wedge r) \vee (p \wedge \bar{q} \wedge r) \vee (p \wedge q \wedge \bar{r}) \vee (\bar{p} \wedge q \wedge \bar{r})$$

Выше мы стали использовать другое обозначение для отрицания: пишем $\bar{p}$ вместо $\neg p$.

We find all pairs of the form (найдём все пары вида): $(F_1 \wedge r \wedge F_2) \vee (F_1 \wedge \bar{r} \wedge F_2)$:

$$(p \wedge q \wedge r) \vee (p \wedge \bar{q} \wedge r), \quad (p \wedge q \wedge r) \vee (p \wedge q \wedge \bar{r}), \text{ and } (p \wedge q \wedge \bar{r}) \vee (\bar{p} \wedge q \wedge \bar{r})$$

These pairs give (эти пары дают): $(p \wedge r)$, $(p \wedge q)$, and $(q \wedge \bar{r})$[11].

We obtain that (мы получаем, что)

$$(p \wedge q \wedge r) \vee (p \wedge \bar{q} \wedge r) \vee (p \wedge q \wedge \bar{r}) \vee (\bar{p} \wedge q \wedge \bar{r}) \Leftrightarrow (p \wedge r) \vee (p \wedge q) \vee (q \wedge \bar{r})$$

but it is not the final version of the answer (но это не конечный результат).

Note that (заметим, что) $(p \wedge r)$ implies $(p \wedge q \wedge r)$ and $(p \wedge \bar{q} \wedge r)$;
$(p \wedge q)$ implies $(p \wedge q \wedge r)$ and $(p \wedge q \wedge \bar{r})$;
$(q \wedge \bar{r})$ implies $(p \wedge q \wedge \bar{r})$ and $(\bar{p} \wedge q \wedge \bar{r})$.

We see that two propositions $(p \wedge r)$ and $(q \wedge \bar{r})$ imply all minterms[12] from the given full DNF, that is why the proposition $(p \wedge q)$ can be omitted.

Мы видим, что два высказывания $(p \wedge r)$ и $(q \wedge \bar{r})$ влекут все элементарные конъюнкции из данной СДНФ; вот почему высказывание $(p \wedge q)$ может быть опущено.

Finally, we obtain (в итоге мы получаем):

$$(p \wedge q \wedge r) \vee (p \wedge \bar{q} \wedge r) \vee (p \wedge q \wedge \bar{r}) \vee (\bar{p} \wedge q \wedge \bar{r}) \Leftrightarrow (p \wedge r) \vee (q \wedge \bar{r})$$

There is another explanation of the fact that $(p \wedge r) \vee (p \wedge q) \vee (q \wedge \bar{r})$ is equivalent to $(p \wedge r) \vee (q \wedge \bar{r})$ (есть и другое объяснение вышеуказанной эквивалентности).

Note that $p \wedge q \Leftrightarrow (p \wedge q) \wedge 1 \Leftrightarrow (p \wedge q) \wedge (r \vee \bar{r}) \Leftrightarrow (p \wedge q \wedge r) \vee (p \wedge q \wedge \bar{r})$.

Now we replace $(p \wedge q)$ in $(p \wedge r) \vee (p \wedge q) \vee (q \wedge \bar{r})$ with $(p \wedge q \wedge r) \vee (p \wedge q \wedge \bar{r})$ and twice apply the absorption law.

$$\begin{array}{cc}
(p \wedge r) \vee (p \wedge q) \vee (q \wedge \bar{r}) & \Leftrightarrow \\
(p \wedge r) \vee (p \wedge q \wedge r) \vee (p \wedge q \wedge \bar{r}) \vee (q \wedge \bar{r}) & \Leftrightarrow \\
[(p \wedge r) \vee ((p \wedge q) \wedge r)] \vee [((p \wedge q) \wedge \bar{r}) \vee (q \wedge \bar{r})] & \Leftrightarrow \\
(p \wedge r) \vee (q \wedge \bar{r}) &
\end{array}$$

**Glossary**

| | |
|---|---|
| Then — тогда | Imply — влечёт |
| Twice — дважды | Replace — заменяем |

[11] These propositions are called *prime implicants* (эти выражения называют *простыми импликантами*).

[12] Minterm is another name of elementary conjunctions (минтерм — элементарная конъюнкция).

### 1.1.25 Minimization of DNF. Quine's method

Quine's method consists of two steps:
1) Finding prime implicants (the result of application the law $(F \wedge p) \vee (F \wedge \bar{p}) \Leftrightarrow F$).
2) Prime implicant chart (reducing unnecessary implicants).

Метод Квайна состоит из двух шагов:
1) поиск простых импликантов (результатов применения вышеуказанного закона);
2) таблица простых импликантов (сокращение ненужных импликантов).

Consider a full DNF (рассмотрим СДНФ):

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \bar{r}) \vee (p \wedge \bar{q} \wedge r) \vee (p \wedge \bar{q} \wedge \bar{r}) \vee (\bar{p} \wedge \bar{q} \wedge \bar{r})$$

We find all pairs of the form (найдём все пары вида) $(F \wedge p) \vee (F \wedge \bar{p})$:

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \bar{r}) \vee (p \wedge \bar{q} \wedge r) \vee (p \wedge \bar{q} \wedge \bar{r}) \vee (\bar{p} \wedge \bar{q} \wedge \bar{r})$$

We calculate all implicants (найдём все импликанты):

$$(p \wedge r), \quad (p \wedge q), \quad (p \wedge \bar{r}), \quad (p \wedge \bar{q}), \quad (\bar{q} \wedge \bar{r})$$

The first and the third implicants give $p$, as well as the second and the forth ones. So, the minimized DNF is $p \vee (\bar{q} \wedge r)$.

Первый и третий импликанты дают $p$, так же как и второй и четвёртый. Таким образом, минимизированная ДНФ — $p \vee (\bar{q} \wedge r)$.

Consider another example (рассмотрим другой пример):

$$(p \wedge q \wedge r \wedge s) \vee (p \wedge q \wedge r \wedge \bar{s}) \vee (p \wedge \bar{q} \wedge r \wedge \bar{s}) \vee (\bar{p} \wedge \bar{q} \wedge r \wedge \bar{s}) \vee (\bar{p} \wedge \bar{q} \wedge \bar{r} \wedge s) \vee (\bar{p} \wedge \bar{q} \wedge \bar{r} \wedge \bar{s})$$

We find all prime implicants and obtain (находим все простые импликанты и получаем):

$$(p \wedge q \wedge r) \vee (q \wedge r \wedge \bar{s}) \vee (\bar{p} \wedge r \wedge \bar{s}) \vee (\bar{p} \wedge \bar{q} \wedge \bar{s}) \vee (\bar{p} \wedge \bar{q} \wedge \bar{r})$$

Enumerate all minterms in the given full DNF and construct the prime implicant chart (перенумеруем все элементарные конъюнкции в данной СДНФ и построим таблицу простых импликантов):

| Prime implicants | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|
| $(p \wedge q \wedge r)$ | × | × | | | | | ✓ |
| $(q \wedge r \wedge \bar{s})$ | | × | × | | | | |
| $(\bar{p} \wedge r \wedge \bar{s})$ | | | × | × | | | ✓ |
| $(\bar{p} \wedge \bar{q} \wedge \bar{s})$ | | | | × | | × | |
| $(\bar{p} \wedge \bar{q} \wedge \bar{r})$ | | | | | × | × | ✓ |

We see that the prime implicants $(p \wedge q \wedge r)$, $(\bar{p} \wedge r \wedge \bar{s})$, and $(\bar{p} \wedge \bar{q} \wedge \bar{r})$, denoted in the above table by ✓, covers all minterms. So, the minimized DNF is

$$(p \wedge q \wedge r) \vee (\bar{p} \wedge r \wedge \bar{s}) \vee (\bar{p} \wedge \bar{q} \wedge \bar{r})$$

Мы видим, что простые импликанты $(p \wedge q \wedge r)$, $(\bar{p} \wedge r \wedge \bar{s})$ и $(\bar{p} \wedge \bar{q} \wedge \bar{r})$, выделенные в вышеприведённой таблице знаком ✓, покрывают все необходимые элементарные конъюнкции. Поэтому минимизированная ДНФ получается, как приведено выше.

### 1.1.26 Minimization of DNF. Quine–McCluskey method

We have seen that looking for minterms of the form $(F \wedge r)$ and $(F \wedge \bar{r})$ is rather laborious. Edward McCluskey suggested some modification of the Quine method.

Мы видели, что поиск минтермов вида $(F \wedge r)$ и $(F \wedge \bar{r})$ достаточно трудоёмок. Эдвард Мак-Класки предложил модификацию метода Квайна.

Consider a disjunction of two minterms, say, $(p \wedge \bar{q} \wedge r) \vee (p \wedge \bar{q} \wedge \bar{r})$. These minterms correspond to the rows $\boxed{1 \mid 0 \mid 1}$ and $\boxed{1 \mid 0 \mid 0}$ of the truth table.

Рассмотрим дизъюнкцию двух минтермов, скажем, $(p \wedge \bar{q} \wedge r) \vee (p \wedge \bar{q} \wedge \bar{r})$. Они соответствуют строкам $\boxed{1 \mid 0 \mid 1}$ и $\boxed{1 \mid 0 \mid 0}$ данной таблицы истинности.

We can see that the number of 1's in these rows differs by one. That is why we rearrange the rows in truth tables in the following ways: first, we put the row with all 1's; second, we put the rows with all but one 1, or with exactly one 0; and so one. Also we omit rows in which the proposition is false.

Можно заметить, что число единиц в этих строках отличается на одну. Вот почему мы меняем порядок строк следующим способом: сперва мы пишем строку, составленную только из единиц; потом — строку, где все, кроме одного, значения — 1, или где ровно один 0; и так далее. Также мы опускаем строки, в которых высказывание ложно.

Consider the following truth table (рассмотрим следующую таблицу истинности).

| $p$ | $q$ | $r$ | $F(p,q,r)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

| $p$ | $q$ | $r$ | $F(p,q,r)$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 |

We obtain four prime implicants (получили четыре простых импликанта):

$$(p \wedge q) \vee (q \wedge r) \vee (p \wedge \bar{r}) \vee (\bar{q} \wedge \bar{r})$$

We construct the prime implicant chart (построим таблицу простых импликантов):

| Prime implicants | $p \wedge q \wedge r$ | $p \wedge q \wedge \bar{r}$ | $\bar{p} \wedge q \wedge r$ | $p \wedge \bar{q} \wedge \bar{r}$ | $\bar{p} \wedge \bar{q} \wedge \bar{r}$ | | |
|---|---|---|---|---|---|---|---|
| $p \wedge q$ | × | × | | | | ✓ or | |
| $q \wedge r$ | × | | × | | | ✓ | ✓ |
| $p \wedge \bar{r}$ | | × | | × | | | ✓ |
| $\bar{q} \wedge \bar{r}$ | | | | × | × | ✓ | ✓ |

Minimized DNF is either $(p \wedge q) \vee (q \wedge r) \vee (\bar{q} \wedge \bar{r})$ or $(q \wedge r) \vee (p \wedge \bar{r}) \vee (\bar{q} \wedge \bar{r})$.

There is a graphical representation, which is functionally identical to Quine–McCluskey method and is called Karnaugh mapping, but it is out of the scope of this textbook.

Существует графическое представление, которое функционально идентично методу Квайна–МакКласки и называется „карты Карно", но оно выходит за рамки этого учебника.

### 1.1.27 Checking whether a proposition is a tautology

Now we know two ways of checking whether a proposition is a tautology: constructing its truth tables and applying logical laws in order to reduce a given proposition to 1.

Sometimes it is not necessary to construct a whole truth table. If we know that in some rows the answer is 1, we can omit calculating the truth value in these rows.

For instance, consider a proposition of the form $P \to Q$, where both $P$ and $Q$ are compound proposition.

We know that implication is false only if the hypothesis is true and the conclusion is false. All other cases give 1, so we can omit them.

Indeed, if $Q = 1$, then $P \to Q \Leftrightarrow P \to 1 \Leftrightarrow \neg P \vee 1 \Leftrightarrow 1$,
if $P = 0$, then $0 \to P \Leftrightarrow \neg 0 \vee Q \Leftrightarrow 1 \vee Q \Leftrightarrow 1$.

Let us consider a simple example: $p \to (q \to p)$.
The hypothesis here is $p$, and the conclusion is $(q \to p)$.

$$\underbrace{p}_{\text{hypothesis}} \to \underbrace{(q \to p)}_{\text{conclusion}}$$

We assume that $p = 1$. Then we consider the conclusion $q \to 1$, which is equivalent to 1.
Hence, if the hypothesis $p$ is 1, then the conclusion $q \to p$ is also 1.
By definition of implication $1 \to 1$ gives 1. Thus $p \to (q \to p)$ is a tautology.

One more time. If $p = 1$ then $p \to (q \to p) \Leftrightarrow 1 \to (q \to 1) \Leftrightarrow 1 \to 1 \Leftrightarrow 1$,
If $p = 0$ then $0 \to (q \to 0) \Leftrightarrow \neg 0 \vee (q \to 0) \Leftrightarrow 1 \vee (q \to 0) \Leftrightarrow 1$.

Consider one more example: $p \to (p \to q)$.

$$\underbrace{p}_{\text{hypothesis}} \to \underbrace{(p \to q)}_{\text{conclusion}}$$

So we assume that $p = 1$. Then we consider the conclusion, which is $1 \to q$.
We know that $1 \to 0$ is 0, so if we put $q = 0$ we obtain that $1 \to q$ is 0. So,

$$(1 \to (1 \to 0)) = (1 \to 0) = 0$$

Hence, if $p = 1$ and $q = 0$ we obtain that $p \to (p \to q)$ is false. Thus it is not a tautology.

Мы уже знаем два способа проверки того, является ли данное высказывание тавтологией: построение таблицы истинности и использования логических законов, при помощи которых мы можем, если это тавтология, упростить данное высказывание до 1.

Иногда нет необходимости строить всю таблицу истинности. Скажем, если мы знаем, что в данной строке истинностное значение равно 1, то можно эту строку и не вычислять.

Поскольку $P$ и $Q$ — составные высказывания, не всегда понятно, какие именно строки дают значение 1, но этого и не требуется. Суть данного упрощения заключается в том, что мы пытаемся сделать исходное высказывание ложным и в ходе этого испытания отбрасываем те случаи, которые заведомо дают 1.

Например, мы знаем что дизъюнкция $P \vee Q$ может быть ложной только в одном случае, когда и $P$, и $Q$ ложны. Поэтому остальные случаи можно и не рассматривать.

Рассмотрим $p \vee (p \to q)$. Пусть $p = 0$. Тогда $0 \vee (0 \to q) \Leftrightarrow 0 \to q \Leftrightarrow 1$.

Следовательно, $p \vee (p \to q)$ — тавтология.

### 1.1.28 Graphical checking whether a proposition is a tautology

All of these reasoning can be made in a simple way as it is shown on the next pictures.

Все такие рассуждения удобно изображать, как показано ниже.

Consider (рассмотрим)

$$(p \to (q \to r)) \to ((p \to q) \to (p \to r))$$

If it is 0 then $(p \to q) \to (p \to r)$ is 0. (if = если, then = то)

If $(p \to q) \to (p \to r)$ is 0, then $(p \to r)$ is false.

If $(p \to r)$ is false, then $p = 1$ and $q = 0$.

We can draw this as it shown below. (мы можем изобразить это, как показано ниже)

$$\underbrace{(p \to (q \to r))}_{?} \to \underbrace{(\underbrace{(p \to q)}_{1} \to \underbrace{(\underbrace{p}_{1} \to \underbrace{r}_{0})}_{0})}_{0}$$

We have found that $p = 1$ and $p \to q$ is 1. It is possible only in one case: $q = 1$.

Мы нашли, что $p = 1$ и $p \to q$ истинно. Это возможно только в одном случае: $q = 1$.

$$\underbrace{(\underbrace{p}_{1} \to \underbrace{(\underbrace{q}_{1} \to \underbrace{r}_{0})}_{?})}_{?} \to \underbrace{(\underbrace{(\underbrace{p}_{1} \to \underbrace{q}_{1})}_{1} \to \underbrace{(\underbrace{p}_{1} \to \underbrace{r}_{0})}_{0})}_{0}$$

So, $q \to r$ is 0 as well as $p \to (q \to r)$. Finally, we obtain

Таким образом, $q \to r$ ложно, так же как и $p \to (q \to r)$. В итоге получаем

$$\underbrace{(\underbrace{p}_{1} \to \underbrace{(\underbrace{q}_{1} \to \underbrace{r}_{0})}_{0})}_{0} \to \underbrace{(\underbrace{(\underbrace{p}_{1} \to \underbrace{q}_{1})}_{1} \to \underbrace{(\underbrace{p}_{1} \to \underbrace{r}_{0})}_{0})}_{0}$$

So, if the conclusion $(p \to q) \to (p \to r)$ is false, then the hypothesis $(p \to (q \to r))$ is false, this implies that the proposition is true.

If the conclusion if true, then in any case the proposition is true.

Hence, $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$ is a tautology.

Значит, если заключение $(p \to q) \to (p \to r)$ ложно, то и посылка $(p \to (q \to r))$ ложна, но это влечёт, что данное высказывание истинно.

Если заключение истинно, то в любом случае высказывание истинно.

Таким образом, $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$ является тавтологией.

**Glossary**

Hence, consequently — следовательно

Thus — таким образом, так что

Because, since, as[13] — потому что

[13] We often use *as* and *since* when we want to focus more on the result than the reason. *As* and *since* are more formal than *because*. We often use *as* and *since* clauses at the beginning of the sentence. We use a comma after the *as*- or *since*- clause.

### 1.1.29 Functional completeness

We know 6 logical operations: 1 unary operation (negation) and five binary operations (conjunction, disjunction, exclusive 'Or', implication and biconditional).

Is it sufficient? Is it possible to create a new logical operation in $n$ variables $p_1, \ldots, p_n$ which is not expressible by means of these operations?

In fact, this set of logical operations is sufficient to express any proposition. Moreover, we can remove some of these logical operations.

Which one? We shall get to know this, but first we need the following definition.

**Definition 1.1.19** A system $\mathcal{F}$ of logical operations is said to be *functionally complete*, if for any compound proposition $P$ there exists a proposition $Q$ such that:

— $Q$ is equivalent to $P$;

— each logical operation, which occurs in $Q$, belongs to $\mathcal{F}$.

**Theorem 1.1.20** *The system of logical operations* $\{\neg, \wedge, \vee\}$ *is functionally complete.*

*Proof.* Let $P$ be an arbitrary compound proposition.

— For $P$ we construct its truth tables.

— For this truth tables we construct either DNF, or CNF.

— These DNF and CNF are equivalent to $P$, because they have the same truth tables.

— Both DNF and CNF involves only three logical operations: $\neg$, $\wedge$, and $\vee$.

Thus, $\{\neg, \wedge, \vee\}$ is functionally complete. □

One remark to the proof of Theorem 1.1.20.

There exists one truth table, for which we cannot construct DNF. It is the truth table of a contradiction. Recall that for writing DNF we choose rows with the truth tables 1. But the truth table of a contradiction does not contain any 1. We write a contradiction as $p \wedge \neg p$.

Also there exists one truth table, for which we cannot construct CNF. It is the truth table of a tautology. Recall that for writing CNF we choose rows with the truth tables 0. But the truth table of a tautology does not contain any 0. We write a tautology as $p \vee \neg p$.

На данный момент мы знаем 6 логических операций. Достаточно ли этого, чтобы написать любое высказывание, какое бы мы только ни пожелали?

Ответ на этот вопрос положительный, более того, некоторые операции мы можем даже выбросить. Какие именно? Узнаем чуть позже.

**Определение 1.1.19** Система $\mathcal{F}$ логических операций называется *функционально полной*, если для любого составного высказывания $P$ существует такое высказывание $Q$, что

— $Q$ эквивалентно $P$;

— каждая логическая операция, которая встречается в $Q$, лежит в $\mathcal{F}$.

**Theorem 1.1.20** *Система логических операций* $\{\neg, \wedge, \vee\}$ *функционально полна.*

*Proof.* Пусть $P$ — произвольное составное высказывание.

— Построим для $P$ его таблицу истинности.

— Для этой таблицы истинности построим ДНФ или КНФ.

— Эти ДНФ и КНФ эквивалентны $P$, потому что их таблицы истинности одинаковы.

— И ДНФ, и КНФ содержат только $\neg$, $\wedge$, and $\vee$.

Таким образом, $\{\neg, \wedge, \vee\}$ функционально полна. □

### 1.1.30 Other functionally complete systems

We already know that $\neg$, $\wedge$, and $\vee$ form a functionally complete system.

In this system we can omit one of $\wedge$ and $\vee$, preserving functional completeness.

**Theorem 1.1.21** *The system* $\{\neg, \wedge\}$ *is functionally complete as well as the system* $\{\neg, \vee\}$.

*Proof.* Let $P$ be a compound proposition. Since $\{\neg, \wedge, \vee\}$ is functionally complete, there exists a proposition $Q$, which is equivalent to $P$ and involves only $\neg$, $\wedge$, and $\vee$.

Below we apply the double negation law to the $p \vee q$ and then we apply de Morgan's law to the inner negation:

$$p \vee q \Leftrightarrow \neg(\neg(p \vee q)) \Leftrightarrow \neg(\neg p \wedge \neg q)$$

Now we replace each occurrence of $\vee$ in $Q$ according to this rule: $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$ and obtain a proposition, which is equivalent to $Q$, and then to $P$, and which involves only $\neg$ and $\wedge$.

Thus, $\neg$ and $\wedge$ form a functionally complete system.

We do similar reasoning for $\neg$ and $\vee$.

Below we apply the double negation law to the $p \wedge q$ and then de Morgan's law:

$$p \wedge q \Leftrightarrow \neg(\neg(p \wedge q)) \Leftrightarrow \neg(\neg p \vee \neg q)$$

Now we replace each occurrence of $\wedge$ in $Q$ according to this rule: $p \wedge q \Leftrightarrow \neg(\neg p \vee \neg q)$ and obtain a proposition, which is equivalent $P$, and which involves only $\neg$ and $\vee$.

Thus, $\neg$ and $\vee$ form a functionally complete system. $\square$

Consider some examples. Below we use DNF for $p \leftrightarrow q$ and then $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$.

$$p \leftrightarrow q \Leftrightarrow (p \wedge q) \vee (\neg p \wedge \neg q) \Leftrightarrow \neg(\neg(p \wedge q) \wedge \neg(\neg p \wedge \neg q))$$

We see that the proofs that both $\{\neg, \wedge\}$ and $\{\neg, \vee\}$ are functionally complete are similar. So, we can find a scheme how to find new functionally complete systems of logical operations.

**Theorem 1.1.22** *Assume that* $\mathcal{F}$ *is a functionally complete systems and each logical operation from* $\mathcal{F}$ *is expressible by means of another system* $\mathcal{G}$*. Then* $\mathcal{G}$ *is functionally complete.*

*Proof.* Let $P$ be a compound proposition.

Since $\mathcal{F}$ is functionally complete, there exists a proposition $Q$, which is equivalent to $P$ and involves only those logical operations, which belongs to $\mathcal{F}$.

Now we replace each operation in $Q$ by operations in $\mathcal{G}$, because by our assumption it is possible, and obtain a proposition $R$, which is equivalent to $Q$, and consequently to $P$, and which contains only those operations, which are in $\mathcal{G}$. Thus $\mathcal{G}$ is functionally complete. $\square$

Выше мы доказали, что системы $\{\neg, \wedge\}$ и $\{\neg, \vee\}$ являются функционально полными.

Доказательство достаточно простое.

1) По высказыванию $P$ строим ему эквивалентное высказывание $Q$, которое содержит только $\neg$, $\wedge$ и $\vee$. Это возможно так как $\{\neg, \wedge, \vee\}$ — функционально полная система.

2) В высказывании $Q$ заменяем все дизъюнкции по правилу $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$.

3) Получили высказывание, эквивалентное и $Q$, и $P$, и содержащее только $\{\neg, \wedge\}$.

Следовательно, система $\{\neg, \wedge\}$ функционально полна.

Функциональная полнота системы $\{\neg, \vee\}$ доказывается похожим образом.

Более общо, аналогичным методом можно доказать, что если $\mathcal{F}$ — функционально полная система и что каждая логическая операция из $\mathcal{F}$ выразима при помощи логических опреаций из $\mathcal{G}$, то и система $\mathcal{G}$ функционально полная (теорема 1.1.22).

### 1.1.31 More functionally complete systems

**Theorem 1.1.23** *The system $\{\neg, \to\}$ is functionally complete.*

*Proof.* By Theorem 1.1.21 the system $\{\neg, \vee\}$ is functionally complete.
Since $p \vee q \Leftrightarrow \neg p \to q$, so $\{\neg, \to\}$ is functionally complete, because of Theorem 1.1.22. □

We consider some examples. (Рассмотрим некоторые примеры)

$$p \vee q \Leftrightarrow \neg p \vee q$$

$$p \wedge q \Leftrightarrow \neg(\neg p \vee \neg q) \Leftrightarrow \neg(p \to \neg q)$$

$$p \oplus q \Leftrightarrow (p \wedge \neg q) \vee (\neg p \wedge q) \Leftrightarrow \neg(\neg p \vee q) \vee \neg(p \vee \neg q) \Leftrightarrow (p \to q) \to \neg(q \to p)$$

$$p \leftrightarrow q \Leftrightarrow (p \to q) \wedge (q \to p) \Leftrightarrow \neg((p \to q) \to \neg(q \to p))$$

Recall, that conjunction is also called the logical multiplication, because $1 \times 1 = 1$ and $0 \times p = 0$ for any $p$.

If we replace the symbol $\wedge$ with $\times$ or $\cdot$, and the symbol $\oplus$ with $+$ we obtain the following[14]

**Theorem 1.1.24** *The system $\{+, \times, 1\}$ is functionally complete.*

*Proof.* By Theorem 1.1.21 the system $\{\neg, \wedge\}$ is functionally complete.
We know that $p \wedge q \Leftrightarrow p \times q = p \cdot q = pq$ (here we use different notations of $\times$).
Let us consider $p + 1$. By definition of exclusive 'Or'

| $p$ | $p+1$ | $\neg p$ |
|---|---|---|
| 1 | $1+1=0$ | 0 |
| 0 | $0+1=1$ | 1 |

Then $\neg p \Leftrightarrow p + 1$. Thus $\{+, \times, 1\}$ is functionally complete by Theorem 1.1.22. □

Here we have two interesting laws:

$$p + p \Leftrightarrow p \oplus p \Leftrightarrow 0, \text{ because } 1 \oplus 1 = 0 \text{ and } 0 \oplus 0 = 0,$$

$$p \times p \Leftrightarrow p \wedge p \Leftrightarrow p, \text{ because of the idempotent law.}$$

Some examples.

$$p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q) \Leftrightarrow (\neg p \wedge \neg q) + 1 \Leftrightarrow (p+1)(q+1) + 1 \Leftrightarrow pq + p + q$$

$$\begin{aligned} p \leftrightarrow q \Leftrightarrow &\quad (\neg p \wedge \neg q) \vee (p \wedge q) \Leftrightarrow ((p+1)(q+1)) \vee pq \Leftrightarrow \\ \Leftrightarrow &\quad (p+1)(q+1)pq + (p+1)(q+1) + pq \Leftrightarrow \\ \Leftrightarrow &\quad p^2q^2 + p^2q + pq^2 + pq + pq + p + q + 1 + pq \Leftrightarrow \\ \Leftrightarrow &\quad \underbrace{pq + pq}_{0} + \underbrace{pq + pq}_{0} + \underbrace{pq + p + q + 1 + pq}_{p+q+1} = p + q + 1 \end{aligned}$$

Such kind of expression are called *Zhegalkin's polynomials* (*многочлены Жегалкина*).

For the proposition $p \vee q \vee r$ we obtain a more complicated law:

$$\begin{aligned} p \vee q \vee r \Leftrightarrow &\neg(\neg p \wedge \neg q \wedge \neg r) \Leftrightarrow (\neg p \wedge \neg q \wedge \neg r) + 1 \Leftrightarrow \\ &\Leftrightarrow \quad (p+1)(q+1)(r+1) + 1 \Leftrightarrow pqr + pq + pr + qr + p + q + r \end{aligned}$$

[14] Если мы заменим символ $\wedge$ на $\times$ или $\cdot$, а символ $\oplus$ на $+$, то получим следующее.

### 1.1.32 Constructing Zhegalkin polynomial for a given truth table

We already know for to write DNF and CNF for a given truth table.

The question is how to write Zhegalkin polynomial for a given truth table.

Assume that the DNF for a given truth table is

$$(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$$

We know that $\neg p \Leftrightarrow (p+1)$ and $p \wedge q \Leftrightarrow pq$. Laws for disjunction are more complicated:

$$\begin{aligned} p \vee q &\Leftrightarrow pq + p + q \\ p \vee q \vee r &\Leftrightarrow pqr + pq + pr + qr + p + q + r \\ p \vee q \vee r \vee s &\Leftrightarrow pqrs + pqr + prs + pqs + qrs + \\ &\qquad +pq + pr + pr + ps + qr + qs + rs + p + q + r + s \end{aligned}$$

So, using these laws is quite fatigue, but there is one trick here.

Recall the truth tables of disjunction and exclusive 'Or':

| $p$ | $q$ | $p \vee q$ | $p \oplus q$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

As we can see the difference is just in the first row.

So, if it is not possible that two compound propositions $P$ and $Q$ are true at the same time, then $P \vee Q$ looks like $P \oplus Q$.

Full DNF has this property: if one elementary conjunction is true, then all other elementary conjunctions are false. So,

$$\begin{gathered} (p\ \wedge\ q\ \wedge\ r)\ \vee\ (p\ \wedge\ q\ \wedge\ \neg r)\ \vee\ (p\ \wedge\ \neg q\ \wedge\ r)\ \vee\ (\neg p\ \wedge\ \neg q\ \wedge\ \neg r) \Leftrightarrow \\ p \cdot q \cdot r + p \cdot q \cdot (r+1) + p \cdot (q+1)(r+1) + (p+1)(q+1)(r+1) \end{gathered}$$

Now we open parentheses and apply the law that $P + P \Leftrightarrow 0$.

$$\begin{gathered} p \cdot q \cdot r + p \cdot q \cdot (r+1) + p \cdot (q+1)(r+1) + (p+1)(q+1)(r+1) = \\ = \underbrace{p \cdot q \cdot r + p \cdot q \cdot r}_{0} + p \cdot q + \underbrace{p \cdot (q+1)(r+1) + p \cdot (q+1)(r+1)}_{0} + (q+1)(r+1) = \\ = p \cdot q + q \cdot r + q + r + 1 \end{gathered}$$

**Caution:** this method is applicable only for a full DNF!

Как построить многочлен Жегалкина по данной таблицы истинности?

Если законы преобразования отрицания и конъюнкции достаточно просты, то законы преобразования дизъюнкции весьма громоздки и тяжело применимы.

Однако существует один трюк, который позволяет значительно упростить задачу: в совершенной ДНФ можно заменить дизъюнкцию на исключающее „Или", поскольку $P \vee Q$ и $P \oplus Q$ отличаются только если и $P$, и $Q$ истинны, что невозможно в совершенной ДНФ: если одна скобка истинна, то все остальные ложны.

### 1.1.33 Nand and Nor

We know that the systems $\{\neg, \wedge\}$, $\{\neg, \vee\}$, and $\{\neg, \to\}$ are functionally complete.

These systems consists of two logical operations.

So, it is a quite natural question whether there exists one logical operation, which forms a functional complete system.

Here we suggest two such logical operations: Nand and Nor.

Nand, or Sheffer stroke is a binary logical operation which is denoted as $p \mid q$ and defined by its truth table (see below).

Nor, or Peirce arrow is a binary logical operation which is denoted as $p \downarrow q$ and defined by its truth table (see below).

Nand:

| $p$ | $q$ | $p \mid q$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

Nor:

| $p$ | $q$ | $p \downarrow q$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

**Theorem 1.1.25** *Sheffer stroke forms a functionally complete system.*
*Peirce arrow forms a functionally complete system.*

*Proof.* It is quite easy to see that $p \mid q \Leftrightarrow \neg(p \wedge q)$ and $p \downarrow q \Leftrightarrow \neg(p \vee q)$. Then

$$\neg p \Leftrightarrow \neg(p \wedge p) \Leftrightarrow p \mid p$$

$$\neg p \Leftrightarrow \neg(p \vee p) \Leftrightarrow p \downarrow p$$

That is why Sheffer stroke is also called Nand, that stands for 'Not And'.
That is why Peirce arrow is also called Nor, that stands for 'Not Or'.
Also, we can see that

$$p \wedge q \Leftrightarrow \neg\neg(p \wedge q) \Leftrightarrow \neg(\neg(p \wedge q)) \Leftrightarrow \neg(p \mid q) \Leftrightarrow (p \mid q) \mid (p \mid q)$$

$$p \vee q \Leftrightarrow \neg\neg(p \vee q) \Leftrightarrow \neg(\neg(p \vee q)) \Leftrightarrow \neg(p \downarrow q) \Leftrightarrow (p \downarrow q) \downarrow (p \downarrow q)$$

Since $\{\neg, \wedge\}$ is functionally complete, so is $\{\mid\}$ by Theorem 1.1.22.
Since $\{\neg, \vee\}$ is functionally complete, so is $\{\downarrow\}$ by Theorem 1.1.22 □

Поскольку ранее изученные функционально полные системы состоят как минимум из двух логических операций, можно задать совершенно естественный вопрос, а можно ли найти одну операцию, которая образует функционально полную систему.

Ответ на этот вопрос положительный. Выше мы привели две такие операции: штрих Шеффера $p \mid q$ и стрелка Пирса $p \downarrow q$, таблицы истинности которых приведены выше.

Поскольку при помощи $\mid$ можно выразить и $\neg$, и $\wedge$, штрих Шеффера образует функционально полную систему, так как система $\{\neg, \wedge\}$ — функционально полна.

Поскольку при помощи $\downarrow$ можно выразить и $\neg$, и $\vee$, стрелка Пирса образует функционально полную систему, так как система $\{\neg, \vee\}$ — функционально полна.

**Glossary**

| | |
|---|---|
| Stands for — обозначает | Nor, Peirce arrow — стрелка Пирса |
| Consists of — состоит из | Nand, Sheffer stroke — штрих Шеффера |

### 1.1.34 Applications of functional completeness

Here we show some examples of replacing some logical operations by other operations.

$$p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q) \Leftrightarrow \neg((p \mid p) \wedge (q \mid q)) \Leftrightarrow \ (p \mid p) \mid (q \mid q)$$
$$p \wedge q \Leftrightarrow \neg(\neg p \vee \neg q) \Leftrightarrow \neg((p \mid p) \vee (q \mid q)) \Leftrightarrow (p \downarrow p) \downarrow (q \downarrow q)$$

Let us consider $p \oplus q$. We are going to express exclusive 'Or' using only Sheffer stroke.

First of all we have to rewrite $p \oplus q$ using $\neg$, $\wedge$, and $\vee$. We can do this in two ways: we write either DNF, or CNF. After that we use logical laws for expressing $\neg$, $\wedge$, and $\vee$ by $\mid$.

$$\begin{aligned}
p \oplus q &\Leftrightarrow (\neg p \vee \neg q) \wedge (p \vee q) && \text{we have written CNF;}\\
&\Leftrightarrow \neg(p \wedge q) \wedge \neg(\neg p \wedge \neg q)) && \text{we have applied de Morgan's law;}\\
&\Leftrightarrow (p \mid q) \wedge ((\neg p) \mid (\neg q)) && \text{because } \neg(P \wedge Q) \Leftrightarrow P \mid Q;\\
&\Leftrightarrow (p \mid q) \wedge ((p \mid p) \mid (q \mid q)) && \text{because } \neg P \Leftrightarrow P \mid P;\\
&\Leftrightarrow ((p \mid q) \mid ((p \mid p) \mid (q \mid q))) \mid ((p \mid q) \mid ((p \mid p) \mid (q \mid q)))\\
& && \text{because } P \wedge Q \Leftrightarrow (P \mid Q) \mid (P \mid Q).
\end{aligned}$$

$$\begin{aligned}
p \oplus q &\Leftrightarrow (p \wedge \neg q) \vee (\neg p \wedge q) && \text{we have written DNF;}\\
&\Leftrightarrow \neg(\neg(p \wedge \neg q) \wedge \neg(\neg p \wedge q)) && \text{we have applied de Morgan's law;}\\
&\Leftrightarrow \neg((p \mid (\neg q)) \wedge ((\neg p) \mid q)) && \text{because } \neg(P \wedge Q) \Leftrightarrow P \mid Q;\\
&\Leftrightarrow (p \mid (\neg q)) \mid ((\neg p) \mid q) && \text{because } \neg(P \wedge Q) \Leftrightarrow P \mid Q;\\
&\Leftrightarrow (p \mid (q \mid q)) \mid ((p \mid p) \mid q) && \text{because } \neg P \Leftrightarrow P \mid P.
\end{aligned}$$

We have obtained two different answers, which, of course, are equivalent, but the second one is shorter.

Now we express the same proposition $p \oplus q$ using $\neg$ and $\rightarrow$.

$$\begin{aligned}
p \oplus q \Leftrightarrow &\quad (p \wedge \neg q) \vee (\neg p \wedge q) && \text{we have written DNF;}\\
&\Leftrightarrow \neg(\neg p \vee q) \vee \neg(p \vee \neg q) && \text{we have twice applied de Morgan's law;}\\
&\Leftrightarrow \neg(p \rightarrow q) \vee \neg(p \vee \neg q) && \text{because } \neg P \vee Q \Leftrightarrow P \rightarrow Q;\\
&\Leftrightarrow \neg(p \rightarrow q) \vee \neg(\neg q \vee p) && \text{because } \quad P \vee Q \Leftrightarrow Q \vee P;\\
&\Leftrightarrow \neg(p \rightarrow q) \vee \neg(q \rightarrow p) && \text{because } \neg P \vee Q \Leftrightarrow P \rightarrow Q;\\
&\Leftrightarrow \quad (p \rightarrow q) \rightarrow \neg(q \rightarrow p) && \text{because } \neg P \vee Q \Leftrightarrow P \rightarrow Q.
\end{aligned}$$

One more example: we express $(p \wedge (p \vee (r \oplus q))) \vee q$ using $\neg$ and $\rightarrow$.

By the absorption law $(P \wedge (P \vee Q)) \Leftrightarrow P$ we obtain $(p \wedge (p \vee (r \oplus q))) \vee q \Leftrightarrow p \vee q$, which is equivalent to $\neg p \rightarrow q$. Sometimes it is better to simplify, and only then to express.

**Glossary**

| | | | |
|---|---|---|---|
| We are going to express — | мы собираемся выразить | Rewrite — | переписать |
| We have applied — | мы только что применили | Replace — | заменить |

### 1.1.35 Systems which are not functionally complete

There are some systems which are not functionally complete.

Consider $\{\wedge, \vee, \rightarrow\}$. What do we know about these logical operations?

We know that $1 \wedge 1 = 1 \vee 1 = 1 \rightarrow 1 = 1$. Such king of functions are called 1-*preserving*.

So, any compound proposition involving only these operations is true provided that each propositional variable is true. But $\neg p$ does not satisfy this property.

Hence, the system $\{\wedge, \vee, \rightarrow\}$ is not functionally complete.

Consider $\{\wedge, \vee, \oplus\}$. What do we know about these logical operations?

We know that $0 \wedge 0 = 0 \vee 0 = 0 \oplus 0 = 0$. Such king of functions are called 0-*preserving*.

So, any compound proposition involving only these operations is false provided that each propositional variable is false. But $\neg p$ does not satisfy this property.

Hence, the system $\{\wedge, \vee, \rightarrow\}$ is not functionally complete.

By Post completeness theorem there exist five classes which are not functionally complete.

— $T_0$ is the class of all 0-preserving functions, such as $f(0, \ldots, 0) = 0$;

— $T_1$ is the class of all 1-preserving functions, such as $f(1, \ldots, 1) = 1$;

— $S$ is the class of all self-dual functions, such as $f(x_1, \ldots, x_n) = \neg f(\neg x_1, \ldots, \neg x_n)$;

— $M$ is the class of all monotonic functions, such as: if $(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n)$ then $f(x_1, \ldots, x_n) \leq f(y_1, \ldots, y_n)$, where $(x_1, \ldots, x_n) \leq (y_1, \ldots, y_n)$ if $x_i \leq y_i$ for each $i$;

— $L$ is the class of all linear functions, which can be presented as: $f(x_1, \ldots, x_n) = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n$, where $a_i = 0$ or $a_i = 1$.

**Theorem 1.1.26 (Post's Functional Completeness Theorem)** *A system $\mathcal{F}$ of logical operations is functionally complete if and only if for each of the five defined classes $T_0$, $T_1$, $S$, $M$, $L$, there is a member of $\mathcal{F}$ which does not belong to that class.*

Не все системы логических операций являются функционально полными.

Рассмотрим $\{0, 1, \wedge, \vee\}$. Мы знаем, что $0 \vee 0 = 0$. Но если увеличить хотя бы одну переменную с 0 до 1, то увеличится и значение — мы получим 1. Дальнейшее увеличение второй переменной не влияет на результат. Такая функция называется *монотонной*[15].

Легко понять, что и константы 0 и 1, и конъюнкция с дизъюнкцией являются монотонными. В курсе математического анализа доказывают, что монотонная функция от монотонных функций является монотонной. Таким образом, при помощи функций из системы $\{0, 1, \wedge, \vee\}$ мы можем получить только монотонные функции. Но $\neg p$ не является монотонной. Следовательно, система $\{0, 1, \wedge, \vee\}$ не является функционально полной.

**Теорема 1.1.26 (Теорема Поста о функциональной полноте)** *Система $\mathcal{F}$ логических операций является функционально полной тогда и только тогда, когда для каждого из пяти классов $T_0$, $T_1$, $S$, $M$ и $L$ существует такая операция из $\mathcal{F}$, которая не лежит в этом классе. Здесь*

*— $T_0$ — класс всех сохраняющих 0 операций;*

*— $T_1$ — класс всех сохраняющих 1 операций;*

*— $S$ — класс всех самодвойственных функций;*

*— $M$ — класс всех монотонных функций;*

*— $L$ — класс всех линейных функций;*

[15]В математическом анализе такую функцию назвали бы слабо возрастающей, или неубывающей, поскольку в математичеком анализе к монотонным функциям относятся и убывающие функции. Однако здесь под терминов „монотонный" подразумевают только „неубывающий".

## 1.2 Naive Set Theory

### 1.2.1 Basic definition

Flowers in a vase. What do you see: a bouquet or several flowers? Is a bouquet one object, or a collection of objects, of flowers? The answer depends only on us. From one point of view a bouquet is a collection of flowers, but we look at a nice bouquet, not flowers. Considering a collection of objects as a whole we call this whole a *set* and the objects *elements*.

A set is one of the most important notions in mathematics. We know the sets of natural numbers and integers. But what is a set? The first thing we can say is the following.

**Definition 1.2.1** A *set* is a collection of elements.

This definition seems to be very simple until we ask what is a collection of elements. We can say that a *collection of elements* is a ... set. Something wrong is here. We are trying to give definition of a set using notion of a set. Unfortunately, this is the best we can do.

There is no definition of a set, because it is a fundamental notion in mathematics, we can just give some explanation what a set looks like.

Georg Cantor, one of the founders of set theory, gave the following definition of a set:

**Definition 1.2.2** A *set* is a gathering together into a whole of definite, distinct objects of our perception or of our thought—which are called elements of the set.

Цветы в вазе. Что мы видим: букет или несколько цветов? Является ли букет одним объектом или набором объектов, цветов? Ответ зависит только от нас. С одной точки зрения букет — это набор цветов, с другой — мы смотрим на букет, а не на цветы. Рассматривая совокупность объектов как целое, мы называем это целое *множеством*, а объекты — элементами.

Множество — одно из самых важных понятий в математике. Мы знаем множества натуральных и целых чисел. Но что такое множество? Первое, что мы можем сказать, это следующее.

**Определение 1.2.1** *Множество* — это совокупность элементов.

Это определение кажется очень простым, но лишь до тех пор, пока мы не спросим, а что такое „совокупность элементов"? Мы можем сказать, что *совокупность элементов* — это... множество. Что-то здесь не так. Мы пытаемся дать определение множества используя понятие множества. К сожалению, это всё, что мы можем сделать.

Не существует определения множества, поскольку это фундаментальное понятие в математике, мы можем только дать некоторое объяснение тому, как множество выглядет.

Георг Кантор, один из основателей теории множеств, дал следующее определение множества:

**Определение 1.2.2** *Множество* есть многое, мыслимое нами как единое.

**Glossary**

| | | |
|---|---|---|
| An element — элемент | A set — множество | A whole — целое |
| A collection — набор, коллекция | | Thought — мысль |

### 1.2.2 Sets, elements, and notations

Any set can be defined by listing its elements, which we write inside braces (curly brackets).

We write $A = \{1, 3, 7\}$, meaning that the set $A$ consists of three elements: 1, 3, and 7.

We write $3 \in A$ and $4 \notin A$, meaning that the element 3 belongs to the set $A$ and that the element 4 does not belong to the set $A$.

If a set is finite, we really can list all its elements. Unfortunately, we do not know yet which set is finite. Perhaps, it is better to write that if we can list all elements of a set, then this set is finite.

Множество может быть определено списков всех своих элементов, которые мы записываем в фигурных скобках.

Пишем, что $A = \{a, b, c\}$, подразумевая, что $A$ состоит из трёх элементов: $a$, $b$ и $c$.

Пишем $3 \in A$ и $4 \notin A$, подразумевая, что элемент 3 принадлежит множеству $A$ и что элемент 4 не принадлежит этому множеству.

Если множество конечно, то, действительно, можно перечислить все его элементы. К сожалению, мы до сих пор не знаем, какое множество можно назвать конечным. Возможно, лучше сказать, что если мы можем перечислить все элементы множества, то это множество конечно.

In order to shorten a note we can use suspension points (dots) "…":

Для того, чтобы укоротить запись, мы можем использовать многоточие:

$$\{1, 2, 3, \ldots, 10\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Without dots we cannot list all elements of an infinite set, for instance, $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$. Note, that dots give just a hint how to continue listing of elements, it is not a mathematical operation.

Без многоточия было бы сложно перечислить все элементы бесконечного множества. Заметим, что многоточие дает лишь намёк на то, как продолжить список элементов, но оно не является математической операцией.

If we are going to list some (not all) elements of a set $A$, we use set builder notation:

$$\{x \in A : x \text{ satisfies a given property}\} \text{ or } \{x \in A \mid x \text{ satisfies a given property}\}$$

where there is a variable on the left of the separator (":", "|"), and a rule on the right of it.

For example,

$$\begin{aligned}
\{x \in \mathbb{Z} : x^2 \leq 9\} &= \{-3, -2, -1, 0, 1, 2, 3\} \\
\{x \in \mathbb{N} : x < 10\} &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\
\{x \in \mathbb{N} : x \text{ is prime}\} &= \{2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots\} \\
\{x \in \mathbb{Z} : x \text{ is a full square}\} &= \{0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, \ldots\}
\end{aligned}$$

Если же мы хотим перечислить лишь некоторые (не все) элементы множества $A$, мы используем следующие обозначения:

$$\{x \in A : x \text{ удовлетворяет заданному свойству}\}$$

где есть переменная слева от разделителя (":", "|"), и некоторое правило справа от него.

**Glossary**

Suspension points (dots) — многоточие
Braces (curly brackets) — фигурные скобки
To list — перечислить
A prime — простое число

### 1.2.3 A subset

**Definition 1.2.3** A set $B$ is a *subset* of a set $A$, if each element of the set $B$ is also an element of the set $A$. We denote it by $B \subseteq A$.

A set $A$ is a *superset* of $B$, if $B$ is a subset of $A$.

Example (пример): $\{1,3,7\} \subseteq \{0,1,2,3,4,5,6,7,8,9\}$.

**Определение 1.2.3** Множество $B$ является *подмножеством* множества $A$, если каждый элемент множества $B$ является также элементом множества $A$.

Мы обозначаем это при помощи записи $B \subseteq A$.

Множество $A$ является *надмножеством* для $B$, если $B$ — подмножество для $A$.

The sentence 'each element of the set $B$ is also an element of the set $A$' is quite long. Implication allows us to shorten this sentence in the following way:

$$B \subseteq A \text{ iff (for all elements } x)(x \in B \to x \in A)$$

Предложение „каждый элемент множества $B$ является также элементом множества $A$" достаточно длинное. Импликация позволяет нам укоротить его.

This note is shorter, nevertheless there exists one more way of shortening this notation. The phrases 'for all elements $x$' and 'for any elements' are quite often used in mathematics. That is why mathematicians introduced special symbols for expressing this phrase. We shall use the symbol $\forall$, which was introduced by German mathematician Gerhard Gentzen in 1935. The symbol $\forall$ is a rotated letter "A", which stands for "for all", "for any": $\forall$ll, $\forall$ny.

So, we can write:

$$B \subseteq A \text{ iff } \forall x(x \in B \to x \in A)$$

Поскольку выражения вида „для любого элемента $x$" и „для каждого элемента" встречаются часто, математики придумывали специальные обозначения для записи таких выражений. Мы будем использовать знак, предложенным Г. Генценом в 1935 году. Символ „$\forall$" представляет собой перевернутую букву „A", которая обозначает „для любого", „для каждого", из английского $\forall$ll, $\forall$ny. .

Clearly, $\{1,2\}$ is not a subset of $\{1,3\}$, because $2 \notin \{1,3\}$. We denote it $\{1,2\} \not\subseteq \{1,3\}$. What does it mean, that $A$ is not a subset of $B$? We put negation to definition of $A \subseteq B$:

$$B \not\subseteq A \text{ iff (it is not true that for all elements } x)(x \in B \to x \in A)$$

Here we have to understand that if not all elements satisfy some property then there exists at least one element which does not satisfies this property. So,

$$\begin{aligned}
B \not\subseteq A \quad & \text{iff} \quad \text{(there exists } x) \text{ it is not true that } (x \in B \to x \in A) \\
& \text{iff} \quad \text{(there exists } x)\ \neg(x \in B \to x \in A) \\
& \text{iff} \quad \text{(there exists } x)\ \neg(\neg(x \in B) \vee x \in A) \\
& \text{iff} \quad \text{(there exists } x)\ (\neg\neg(x \in B) \wedge \neg(x \in A)) \\
& \text{iff} \quad \text{(there exists } x)\ (x \in B \wedge x \notin A)
\end{aligned}$$

Добавляя отрицание к определению подмножества, понимаем, что множество $B$ не является подмножеством множества $A$, если не все элементы множества $B$ лежат в $A$, что равносильно утверждению, что в $B$ найдется такой элемент, которого нет в $A$.

### 1.2.4 Equality of sets

**Definition 1.2.4** Two sets are *equal* if they consists of the same elements.
Два множества *равны*, если они состоят из одних и тех же элементов.
We denote this by (мы обозначаем это): $A = B$.

This definition is short but it does not explain how we can check that two sets are equal. So, we use an equivalent form of this definition: sets $A$ and $B$ are equal if each element of $A$ is in $B$ and each element of $B$ is in $A$, which we can write as

$$A = B \text{ iff } A \subseteq B \land B \subseteq A$$

Хотя определение равенства множеств короткое, оно не объясняет, как проверить, что два множества равны. Поэтому мы используем эквивалентное определение: два множества равны, если все элементы первого множества лежат во втором, и наоборот.

Examples.

Consider $\{1,2\}$ and $\{2,1\}$. We see that $1 \in \{2,1\}$ and $2 \in \{2,1\}$, so $\{1,2\} \subseteq \{2,1\}$. Similarly, $2 \in \{1,2\}$ and $1 \in \{1,2\}$, so $\{2,1\} \subseteq \{1,2\}$. Then $\{1,2\} = \{2,1\}$.

Conclusion: a set does not depend on ordering of its elements.
Вывод: множество не зависит от порядка своих элементов.

Consider $\{1,1,1,2,2\}$ and $\{2,1\}$. We see $1 \in \{2,1\}$, $1 \in \{2,1\}$, $1 \in \{2,1\}$, and $2 \in \{2,1\}$, $2 \in \{2,1\}$. Then $\{1,1,1,2,2\} \subseteq \{2,1\}$. Since $\{2,1\} \subseteq \{1,1,1,2,2\}$, so $\{1,1,1,2,2\} = \{2,1\}$.

Conclusion: a set does not depend on repetition of its elements.
Вывод: множество не зависит от повторения своих элементов.

Since a set can be an element of another set, for instance, $\{1,2\} \in \{3,\{1,2\}\}$, comparing two sets is not always easy. Let us count how many elements are there in the following set:

$$\Big\{ \{1,2,\{1,2\}\}, \{2,\{1,2,1\},1,2\}, \{\{2,1\},2,1,2\} \Big\}$$

First of all, we find repetitions of elements in sets (ищем повторы элементов):

$$\Big\{ \{1,2,\{1,2\}\}, \{\underline{2},\{\underline{1},2,\underline{1}\},1,\underline{2}\}, \{\{2,1\},\underline{2},1,\underline{2}\} \Big\}$$

and remove them (убираем их):

$$\Big\{ \{1,2,\{1,2\}\}, \{2,\{1,2\},1\}, \{\{2,1\},2,1\} \Big\}$$

We change ordering of elements (меняем порядок элементов):

$$\Big\{ \{1,2,\{1,2\}\}, \{1,2,\{1,2\}\}, \{1,2,\{1,2\}\} \Big\}$$

and again remove repetitions (и снова убираем повторы).

Finally we obtain the set (в итоге получаем множество)

$$\Big\{ \{1,2,\{1,2\}\} \Big\},$$

consisting of one element (состоящее из одного элемента) $\{1,2,\{1,2\}\}$.

### 1.2.5 Russell's paradox

In Section 1.2.1 we have given definition of a set, as a collection of elements satisfying some property. Unfortunately, this definition is wrong as we show below.

We call a set *normal*, if it is not an element of itself, otherwise we call this set *abnormal*. For instance, $\{a\} \notin \{a\}$, so any one-element set is normal. On the other hand, if we take the set of all sets, this set is abnormal, because it is a set and we have taken it as an element.

Let $A$ be the set of all normal sets, that is which are not elements of themselves.
Let us answer the question whether $A$ is normal.

Assume that $A$ is normal, that is not a member of itself. Since $A$ is normal and we have put all normal sets into $A$, we have put $A$ into $A$, it means that $A \in A$ and $A$ is not normal.
Now we assume that $A$ is abnormal, that is $A \in A$. Since $A$ is an element of $A$ and each element of $A$ is normal by definition of $A$, so $A$ is normal.
Hence, $A$ is neither normal nor abnormal. This contradiction is Russell's paradox.

How can we solve this problem? We say that this collection of all normal sets is not a set.

It means that there exist objects which look like a set but not a set.
We obtain a new problem: how to distinguish sets from objects which just look like a set.
There is another definition of a set, but nobody knows if it is really better.
Set theory, a part of mathematical logic, studies this extremely complicated question.

Definition 1.2.1 is wrong, but since other definitions are too complicated, we shall use it.

Мы дали определение множества, как совокупности элементов, удовлетворяющих заданному свойству. К сожалению, как мы покажем ниже, это определение неправильно.
Пусть $A = \{x \mid x \notin x\}$, то есть множество $A$ состоит из всех множеств, которые не содержат себя в качестве собственного элемента. Вопрос: $A \in A$?
Если $A \notin A$, то $A$ удовлетворяет условию $x \notin x$, а поскольку мы взяли все множества, удовлетворяющие этому условию, получаем, что мы взяли и $A$, значит $A \in A$.
Если же $A \in A$, то поскольку $A$ является элементом $A$ и все элементы $A$ удовлетворяют условию $x \notin x$, то $A \notin A$. Получили противоречие, называемое парадокс Рассела.

Какой выход из этой ситуации — сказать, что $\{x \mid x \notin x\}$ не является множеством. Но оно выглядет как множество! Значит, есть объекты, которые похожи на множество, но множеством не являются. Как научиться отличать одно от другого? Этим очень сложным вопросом занимается теория множеств — часть математической логики.

Б. Рассел дал и более наглядное изложение этого парадокса.
Предположим, что совет старейшин одной маленькой деревушки решил, что все мужчины в деревне должны ходить чисто выбритыми. При этом в деревне живет один мужчина, который работает брадобреем. Вот ему и приказали брить всех мужчин, которые не бреются сами. Если же кто-то бреется сам, то брадобрею брить его не полагается, поскольку зачем брить бритого. Вопрос: кто должен брить брадобрея?
Если брадобрей бреется сам, то он бреет человека, который бреется сам, но брадобрею запрещено брить людей, которые бреются сами. Значит, он не может брить себя.
Но, с другой стороны, брадобрей должен брить всех людей, кто не бреется сам, значит, поскольку он не бреется сам, то должен брить и себя.
Выходит, что если брадобрей не бреет себя, то он должен брить себя, а если бреет себя, то не должен брить себя — парадокс.

### 1.2.6 Operations over sets: the union and the intersection

Sets can be combined in order to obtain a new set. For example if we take together the set of all boys and the set of all girls we obtain the set of all children.

**Definition 1.2.5** The *union* of two sets $A$ and $B$, denoted by $A \cup B$, is the set that consists of those elements that are in $A$ or in $B$, or in both:

$$\begin{aligned} A \cup B &= \{\, x \mid (x \in A) \vee (x \in B)\} \\ x \in A \cup B &\iff (x \in A) \vee (x \in B) \end{aligned}$$

**Определение 1.2.5** *Объединением* двух множеств $A$ и $B$ является множество, обозначаемое $A \cup B$ и состоящее из тех элементов, что есть в $A$ или в $B$, или и там, и там.

Let $A = \{a, b, c\}$ and $B = \{c, d\}$. Then $A \cup B = \{a, b, c, d\}$.

There is a graphical representation of operations over sets, which are called *Venn Diagram*:

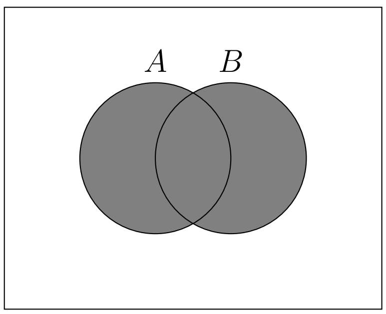


Figure 1.11: The union of two sets

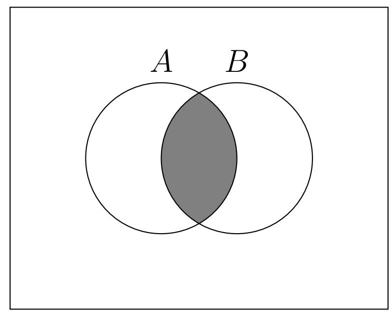


Figure 1.12: The intersection of two sets

**Definition 1.2.6** The *intersection* of two sets $A$ and $B$, denoted by $A \cap B$, is the set that consists of those elements that are in both: in $A$ and in $B$:

$$\begin{aligned} A \cap B &= \{\, x \mid (x \in A) \wedge (x \in B)\} \\ x \in A \cap B &\iff (x \in A) \wedge (x \in B) \end{aligned}$$

**Определение 1.2.6** *Пересечением* двух множеств $A$ и $B$ является множество, обозначаемое $A \cap B$ и состоящее из тех элементов, что есть и в $A$, и в $B$.

Let $A = \{a, b, c\}$ and $B = \{c, d\}$. Then $A \cap B = \{c\}$.

**Lemma 1.2.7** $A \cup B = A \iff B \subseteq A$.

*Proof.* ($\Rightarrow$) Let $A \cup B = A$. Assume, $b \in B$. Since $p \vee 1 \Leftrightarrow 1$, so $(b \in A) \vee (b \in B)$ is true. Then $b \in A \cup B$. By the assumption $A \cup B = A$, so $b \in A$.

Thus, each element of $B$ is in $A$, that is why $B \subseteq A$.

($\Leftarrow$) Let $B \subseteq A$. Assume that $a \in A \cup B$. Then $(a \in A) \vee (a \in B)$ is true.

If $a \in B$ is true, then $a \in A$ is true, because $B \subseteq A$.

If $a \in B$ is false, then $a \in A$ is true, because otherwise $(a \in A) \vee (a \in B)$ is false.

On any case $a \in A$ is true, so $A \cup B \subseteq A$. Obviously, $A \subseteq A \cup B$. Then $A = A \cup B$. $\square$

### 1.2.7 Operations over sets: differences

Here we consider two more operations over sets: the difference (or, set-difference) and the symmetric difference of two sets.

**Definition 1.2.8** The *difference* of two sets $A$ and $B$, denoted by $A \setminus B$, or that is the same $A - B$, is the set that consists of those elements that are in $A$, but not in $B$:

$$\begin{aligned} A \setminus B &= \{ x \mid (x \in A) \wedge (x \notin B) \} \\ x \in A \setminus B &\iff (x \in A) \wedge \neg(x \in B) \end{aligned}$$

**Определение 1.2.8** *Разностью* двух множеств $A$ и $B$ является множество, обозначаемое $A \setminus B$ или $A - B$ и состоящее из тех элементов, что есть в $A$, но нет в $B$.

Let $A = \{a, b, c\}$ and $B = \{c, d\}$. Then $A \setminus B = \{a, b\}$.

Venn diagrams of these operations are the following:

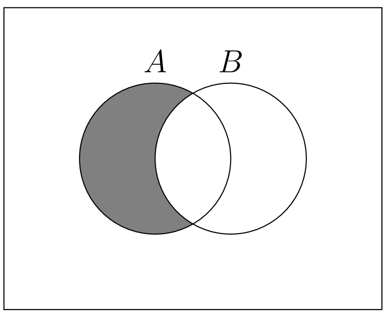


Figure 1.13: The difference of two sets

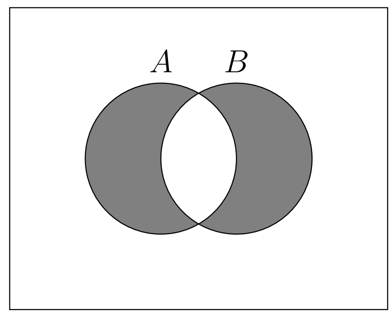


Figure 1.14: The symmetric difference

**Definition 1.2.9** The *symmetric difference* of two sets $A$ and $B$, denoted by $A \oplus B$, is the set that consists of those elements that are either in $A$, but not in $B$, or in $B$, but not in $A$:

$$\begin{aligned} A \oplus B &= \{ x \mid (x \in A) \oplus (x \in B) \} \\ x \in A \oplus B &\iff (x \in A) \oplus (x \in B) \end{aligned}$$

**Определение 1.2.9** *Симметрической разностью* двух множеств $A$ и $B$ является множество, обозначаемое $A \oplus B$ или $A \bigtriangleup B$ и состоящее из тех элементов, что есть в $A$, но нет в $B$, и есть в $B$, но нет в $A$.

Let $A = \{a, b, c\}$ and $B = \{c, d\}$. Then $A \oplus B = \{a, b, d\}$.

**Lemma 1.2.10** $A \cap B = A \iff A \subseteq B$.

*Proof.* ($\Rightarrow$) Let $A \cap B = A$. Assume $a \in A$. Then $a \in A \cap B$ is true, because $A \cap B = A$. Then $a \in A \wedge a \in B$ is true. By definition of conjunction $a \in B$ is true. Thus, $A \subseteq B$.

($\Leftarrow$) Let $A \subseteq B$. Assume $a \in A$. Then $a \in B$, because $A \subseteq B$. So, $a \in A \wedge a \in B$ is true. Then $a \in A \cap B$ is true. Hence $A \subseteq A \cap B$.

Let $a \in A \cap B$. Then $(a \in A) \wedge (a \in B)$ is true.
By definition of conjunction $a \in B$ is true. Hence, $A \cap B \subseteq A$.

Both $A \subseteq A \cap B$ and $A \cap B \subseteq A$ implies that $A \cap B = A$. $\square$

### 1.2.8 An empty set, a universal set and a universe

Given a set $A$ and an element $a$ the expression $a \in A$ is a proposition, it is either true or false. Varying a set and an element we can change the truth value of $a \in A$. Recall, that there are two special propositions: a contradiction and a tautology.

Here we introduce two special sets $\emptyset$ and $U$, which correspond to these propositions in that sense, that $a \in \emptyset$ is false and $a \in U$ is true for any element $a$.

The first set is called an *empty set*, it contains no element. We denote this set by $\emptyset$:

$$\emptyset = \{\}, \quad a \in \emptyset \Leftrightarrow false$$

Если даны множество $A$ и элемент $a$, то выражение $a \in A$ является высказыванием, оно либо истинно, либо ложно. Изменяя множество и элемент, мы меняем истинностное значение выражения $a \in A$. Вспомним, что есть два особых высказывания: противоречие и тавтология.

Здесь мы введём два особых множества $\emptyset$ и $U$, которые соответствуют этим высказываниям в том смысле, что $a \in \emptyset$ ложно, а $a \in U$ истинно для любого элемента $a$.

Первое множество называется *пустое множество*, оно не содержит элементов. Его обозначают символом $\emptyset$.

Defining a set, which corresponds to a tautology, is more complicated. If $a \in U$ is true for any element $a$, any set, considered as an element, is an element of $U$. In particular, $U \in U$, as well as any subset of $U$ is an element of $U$.

This leads us to Russell's Paradox, since we can consider the following subset of $U$:

$$\{a \in U \mid a \text{ is a set and } a \notin a\}$$

Определение множества, которое соответствует тавтологии, более сложное. Если $a \in U$ истинно для любого элемента $a$, то любое множество, рассматриваемое как элемент, является элементом множества $U$. В частности, $U \in U$, так же как и любое подмножество множества $U$ является его элементом. Это приводит нас к парадоксу Рассела, поскольку мы можем рассмотреть вышеуказанное подмножества множества $U$.

That is why we cannot introduce a universal set, the set which contains everything. We introduce a relatively universal set, which we call a *universe*. A universe is a set, which consists of all elements we need for a particular problem.

For instance, when we study arithmetic the universe is the set of all natural numbers, when we study calculus on a line the universe is the set of all real numbers.

Thus, content of a universe depends only on us. But if a universe $U$ is fixed, we consider the proposition $a \in U$ to be identically true.

Вот почему мы не можем ввести универсальное множество, множество, которое содержит всё. Мы вводим относительно универсальное множество, которое называем *универсум*. Универсум — это множество, которое содержит все элементы, необходимые для данной конкретной задачи.

Например, когда мы изучаем арифметику, то универсум — это множество всех натуральных чисел, а когда изучаем математический анализ на прямой, то универсум — это множество всех действительных чисел.

Таким образом, содержание универсума зависит только от нас. Но после того, как мы определились с унивесумом $U$, мы рассматриваем высказывание $a \in U$ как тождественно истинное.

### 1.2.9 Operations over sets: the complement

Given a universe $U$ we can define the complement of a set $A$ (which is a subset of $U$).

**Definition 1.2.11** The *complement* of a set $A$, denoted by $\overline{A}$, is the set that consists of those elements of $U$ that are not in $A$:

$$\begin{aligned} \overline{A} &= \{ x \in U \mid (x \notin A) \} \\ x \in \overline{A} &\iff \neg(x \in A) \end{aligned}$$

*Дополнением* множества $A$ является множество, обозначаемое $\overline{A}$ и состоящее из тех элементов универсума $U$, которых нет в $A$.

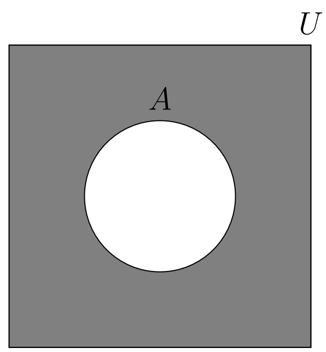


Figure 1.15: The complement of a set

The operation of taking the complement is expressible by means of the difference:
Операцию взятия дополнения можно выразить при помощи разности:

$$\overline{A} = U \setminus A$$

Indeed (действительно):

$$a \in \overline{A} \Leftrightarrow 1 \wedge (a \in \overline{A}) \Leftrightarrow (a \in U) \wedge (a \in \overline{A}) \Leftrightarrow a \in U \setminus A$$

Thus we know 5 operations over sets
(таким образом, мы знаем 5 операций над множествами):

The union: $A \cup B = \{a \in U \mid (a \in A) \vee (a \in B)\}$

The intersection: $A \cap B = \{a \in U \mid (a \in A) \wedge (a \in B)\}$

The set-difference: $A \setminus B = \{a \in U \mid (a \in A) \wedge \neg(a \in B)\}$

The symmetric difference: $A \oplus B = \{a \in U \mid (a \in A) \oplus (a \in B)\}$

The complement: $\overline{A} = \{a \in U \mid \neg(a \in A)\}$

**Glossary**

The union — объединение
The intersection — пересечение
The complement — дополнение
The difference — разность
Symmetric — симметрическая

### 1.2.10 Set identities

Here we give the most important set identities (наиболее важные тождества).

| Name | Identity |
|---|---|
| Identity laws<br>Законы сокращения $\emptyset$ и $U$ | $A \cap U = A$<br>$A \cup \emptyset = A$ |
| Domination laws<br>Законы преобладания $\emptyset$ и $U$ | $A \cap \emptyset = \emptyset$<br>$A \cup U = U$ |
| Idempotent laws<br>Законы идемпотентности | $A \cap A = A$<br>$A \cup A = A$ |
| Double negation law<br>Закон двойного отрицания | $\overline{\overline{A}} = A$ |
| Commutative laws<br>Законы коммутативности | $A \cap B = B \cap A$<br>$A \cup B = B \cup A$ |
| Associative laws<br>Законы ассоциативности | $(A \cap B) \cap C = A \cap (B \cap C)$<br>$(A \cup B) \cup C = A \cup (B \cup C)$ |
| Distributive laws<br>Законы дистрибутивности | $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$<br>$A \cup (B \cap C) = (A \cup B) \cap (A \cup B)$ |
| De Morgan's laws<br>Законы де Моргана | $\overline{A \cap B} = \overline{A} \cup \overline{B}$<br>$\overline{A \cup B} = \overline{A} \cap \overline{B}$ |
| Absorption laws<br>Законы поглощения | $A \cap (A \cup C) = A$<br>$A \cup (A \cap C) = A$ |
| Complement laws<br>Законы для дополнения | $A \cap \overline{A} = \emptyset$<br>$A \cup \overline{A} = U$ |

Please, compare these identities with the logical laws in Section 1.1.17.
Сравните, пожалуйста, эти тождества с логическими законами в разделе 1.1.17.

All of these identities can be proved in the same way:
— we transform set identities to equivalence of propositions by definitions of operations;
— we apply an appropriate logical law;
— we transform an obtained equivalence of propositions back to set identities.

Все эти тождества могут быть доказаны одним и тем же способом:
— преобразуем тождества в эквивалентность высказываний по определению операций;
— применяем подходящий логический закон;
— преобразуем полученную эквивалентность обратно в тождество.

For instance, let us prove the commutative law $A \cap B = B \cap A$.

$$\begin{aligned} a \in A \cap B &\Leftrightarrow (a \in A) \wedge (a \in B) && \text{by definition of } \cap; \\ &\Leftrightarrow (a \in B) \wedge (a \in A) && \text{by the commutative law;} \\ &\Leftrightarrow a \in B \cap A && \text{by definition of } \cap. \end{aligned}$$