Logic One

Jouko Väänänen

University of Helsinki
University of Amsterdam

Available also as video[1]

[1] mathstat.helsinki.fi/logic/people/jouko.vaananen/

Contents

List of Figures

Chapter 1

Propositional logic

1.1 Introduction

Welcome to the *Logic One* course. During this course we learn the very basics of logic. The course is divided into short lectures, sets of solved problems and sets of problems for you to solve. Emphasis in this course is in learning some basic methods that can be used to solve problems involving concepts from logic.

In this lecture we start the first of the two main topics of this course, the topic of propositional logic. Propositional logic is a method for understanding how we use the words "and", "or", "not", etc in everyday language and in science. During the course we find systematic ways to check whether an argument involving these so called connectives is correct or not.

It is important to understand that most sentences in everyday language and in science have in addition to the above mentioned connectives also other logical concepts, so when we only focus on "and", "or", "not", etc, we make a huge simplification. However, this is a useful simplification! Later during this course we learn how to deal with more complex expressions.

In some sense logic is one of the fundamental building blocks of science, something that mankind realized centuries ago. Nowadays it is also true that a lot of the electronic environment we live in, such as internet, computers, mobile phones, etc are based on logic. In a sense, logic is a common language between humans and computers, and perhaps the only common language. Therefore it is not surprising that logic has an important position not only in mathematics and philosophy, but also in computer science.

1.2 Propositional formulas

1.2.1 Introduction

Now we shall introduce the concept of a propositional formula. Intuitively, a propositional formula is an assertion, a sentence, which expresses how things are, something which can be thought to be true or false. Here are some examples:

- $x < 10$
- $x < 10 \rightarrow x^2 < 100$
- $(x = 10 \wedge y = 12) \wedge (z = 4 \vee z = 5)$
- It is raining or snowing.

The first three examples are from the arithmetic of natural numbers, and they are true or false depending on what x, y and z are. The last example is from natural language and it is true or false depending on—well—the weather.

Not all sequences of words are propositional formulas. Take for example

- $x + 10$
- $sin(x)$
- I promise that he comes.
- Stop that!

The first two are mathematical formulas. They have some value, depending on the value of x, but they do not have a meaning as something that is true or false.

The third one, "I promise that he comes" is a kind of speech act, but it is not a true or false sentence, although "I promised that he comes" would be. The last example, "Stop that!" is also a speech act, not a true or false sentence.

In this course it is not very important to have a sharp ear to what is a propositional formula and what is not. In fact, we will give now a mathematical definition of propositional formulas and then everything is clear.

1.2.2 Atomic formulas

We now give a mathematical definition of propositional formulas. The advantage of a mathematical definition is that we can then use mathematical methods for the study of such formulas. The phrase "mathematical definition" means here more or less the same as a "precise definition". When some language is defined in this kind of precise way it becomes what is called a *formal language*. Nobody speaks a formal language, except perhaps a computer when it is "thinking". A formal language is a model of a real language.

The simplest propositional formulas are the mere *proposition symbols*

$$p_0, p_1, p_2, \ldots$$

We call these simplest formulas *atomic formulas*.

When we look at a mathematical sentence like

If $x < 10$ then $x > 5$, and if not $x < 10$ then $x > 15$.

the atomic formulas are $x < 10$, $x > 5$, and $x > 15$. We could replace them by proposition symbols p_0, p_1, and p_2. It doesn't matter which p_n we use as long as we are systematic. When we replace, say $x < 10$, by p_0 we lose some information because obviously $x < 10$ is more informative than mere p_0. This loss of information is *essential* in propositional logic. By giving up some information we achieve freedom which is useful, as we see later. So don't worry when you see information disappear when we pass from natural or mathematical language to the formal language of propositional logic.

In the sentence

If it rains then it blows,
and if it blows then it gets cloudy.

the atomic formulas are "it rains", "it blows" and "it gets cloudy". They are atomic because we cannot break them into smaller pieces by means provided by propositional logic. Again we can replace them by p_0, p_1 and p_2 and from the point of view of propositional logic the meaning of the sentence would not change.

Perhaps some resistance builds up in your mind: How can you take such nice sentences like "it rains", "it blows", "it gets cloudy" and replace them with those cold technical symbols p_0, p_1 and p_2, and still maintain that this tells us something about the original sentence. Welcome to the world of propositional logic! Even after the bold replacement we can see the *logical structure* of the original sentence. This is the point of logic. But let us carry on.

1.2.3 Propositional operations

The propositional operations are:

Negation	$\neg$	not
Conjunction	$\wedge$	and
Disjunction	$\vee$	or
Implication	$\rightarrow$	if ... then...
Equivalence	$\leftrightarrow$	if and only if

Parentheses (,) are used for clarity. These are the so called *connectives*. They are used to form more complex sentences from atomic one. Natural language abounds these little words and we have a pretty good understanding about their meaning. However, we shall in a moment define their meaning mathematically and thereby arrive at exact methods to investigate complex propositional formulas.

There are also other connectives, such as "but", "unless", etc. but we focus on the above most important ones. There are also operations that look like connective but are not, such as "possibly", "until", etc. We shall see later why these cannot be treated the same way as the above connectives.

In the example

If $x < 10$ then $x > 5$, and if not $x < 10$ then $x > 15$.

we can see two occurrences of implication, one of negation, and one of conjunction.

In the example

If it rains then it blows,
and if it blows then it gets cloudy.

we can see one occurrence of conjunction and two occurrences of implication.

1.2.4 Propositional operations (contd.)

We are ready to give an exact definition of propositional formulas. The propositional formulas are of the form

$$\begin{array}{c} p_n \\ \neg A \\ (A \wedge B) \\ (A \vee B) \\ (A \to B) \\ (A \leftrightarrow B) \end{array}$$

where A and B are again propositional formulas. See Figure 1.1. Note that we use parentheses in each case except for negation.

This is an inductive definition in the sense that we build propositional formulas by starting from $p_0, p_1, ...$ and then use successively $\neg, \wedge, \vee, \to$ and $\leftrightarrow$. Whenever we have obtained in this way some formulas A and B, we can form a new formula by applying negation to one of them, or applying conjunction, disjunction, implication or equivalence to the two. And there are no other propositional formulas—each one has to come up like this.

In a moment we look at examples of propositional formulas, but let us first look at the use of parentheses in propositional formulas.

1.2.5 Parentheses

The role of parentheses is to indicate priority and help avoid ambiguity. For example, $(A \wedge (B \vee C))$ is quite different from $((A \wedge B) \vee C)$, not only in appearance but also in meaning. The former is a conjunction while the latter is a disjunction. For this reason we have not allowed propositional formulas of the form $A \wedge B \vee C$. Also, $A \to B \to C$ would be utterly ambiguous. With proper use of parentheses this should be either $(A \to (B \to C))$ or $((A \to B) \to C)$. Other ambiguous formulas, results of wrongful omission of parantheses, are $A \to B \wedge C$ and $A \vee B \to C$.

However, some parentheses are unnecessary. The definition of propositional formulas produces more parentheses than is in practice necessary. Therefore we make the following convention about parentheses:

Parentheses are left out unless necessary for unambiguous reading. $A \wedge B \wedge C$ means either $((A \wedge B) \wedge C)$ or $(A \wedge (B \wedge C))$. Similarly for $A \vee B \vee C$.

The reason why it is usually not necessary to make a distinction between $((A \wedge B) \wedge C)$ and $(A \wedge (B \wedge C))$ is that, when we learn to assign meaning to propositional formulas, we see that these formulas have the same meaning, which is—intuitively speaking—that A, B and C are all three true.

If for some special reason it is necessary to emphasize whether we are talking about $((A \wedge B) \wedge C)$ or $(A \wedge (B \wedge C))$ we can do it by leaving the parentheses where they are.

We also often leave the outmost parentheses away, for simplicity.

1.2.6 Examples

Here are some examples of propositional formulas

- A disjunction: $p_0 \vee p_1$. This is just about the simplest propositional formula we can think of apart from the proposition symbols alone. Note that we have left the parentheses away, according to our convention. So officially this is the formula $(p_0 \vee p_1)$.

- An implication between a disjunction and an atomic formula: $(p_1 \vee p_2) \to p_3$. Intuitively this formula says that if either p_1 or p_2 is true, then so is p_3. But we have not defined yet the meaning of propositional formulas. Until we do so, we assign to this propositional formula just an intuitive meaning, coming from the everyday use of the words "or" and "if...then". When we do define the meaning of propositional formulas in mathematical terms, it turns out that the meaning reflects our everyday use of the words "or" and "if...then" pretty well, but there are also some surprises. But we are getting ahead of ourselves.

- A negation of a disjunction: $\neg(p_1 \vee p_2)$. Intuitive meaning of this formula is that it is **not** the case that

p_n	Proposition symbol
$\neg A$	Negation
$(A \wedge B)$	Conjunction
$(A \vee B)$	Disjunction
$(A \to B)$	Implication
$(A \leftrightarrow B)$	Equivalence

Figure 1.1: Propositional formulas

(pause) p_1 is true or p_2 is true. Note that in spoken language we do not use parentheses but indicate with intonation what we mean. If the formula was $\neg p_1 \vee p_2$, the intuitive meaning would be that it is not the case that p_1 is true (pause) **or** (pause) p_2 is true, something that could be said a bit clearer as: either it is not the case that p_1 is true **or** else p_2 is true.

- A conjunction of an atomic formula and a disjunction of two atomic formulas: $p_0 \wedge (p_1 \vee p_2)$. When we read this formula we also read the parentheses. Also, when we say the intuitive meaning, we may have to add explanatory words because intonation is not sufficient. So we can say: the intuitive meaning of this formula is that p_0 is true and, in addition, either p_1 or p_2 is true.

- A negation of an implication: $\neg(p_3 \to (p_2 \to p_1))$. The intuitive meaning of this is that it is not the case that if p_3 is true then, if p_2 is true, then p_1 is true. Let us not pretend that it is easy to discern what this really means. Both implication and negation are rather complicated operations, and when they are combined and iterated, one has to take paper and pencil to calculate the exact meaning.

- A disjunction of four formulas: $(p_0 \wedge p_2 \wedge p_5) \vee (\neg p_1 \wedge \neg p_3 \wedge \neg p_4) \vee (\neg p_1 \wedge \neg p_4 \wedge p_5) \vee (p_0 \wedge \neg p_3 \wedge p_4)$. This kind of disjunctions, even quite long ones, are common in industrial uses of logic. This could be, for example, a condition under which a car engine has to switch itself off. There are a number of situations, described by the disjuncts, in which the car engine has to switch itself off. If just one of the disjuncts happens to be the case, the engine stops. The meanings of the proposition symbols would be propositions of the type: thermometer indicates overheating, air-intake is blocked, parking break is pulled, etc.

- A conjunction of four formulas: $(p_0 \vee \neg p_1 \vee p_2) \wedge (p_1 \vee \neg p_5 \vee \neg p_6) \wedge (\neg p_1 \vee \neg p_4 \vee p_5) \wedge (p_0 \vee p_3 \vee p_4)$. This type of conjunctions, again possibly quite long, are typical in industrial applications, just as the above disjunctions. An example could be a safety condition for a train to keep moving. There are a number of conditions each one of which has to be satisfied for the train to be allowed to keep moving. The meanings of the proposition symbols would be propositions of the type: a proceed signal is green, a proceed signal is yellow, a proceed signal is red, cab signal is green, crossing warning signal is red, the speed is over 80, etc.

1.2.7 Proposition symbols explained

Let us now focus on the simplest propositional formulas, the proposition symbols, and discuss in detail, what are they for and what can they do for us.

Recall that the proposition symbols are denoted $p_0, p_1, \ldots$. They denote basic states of affairs which can be true or false such as:

- It is raining.
- The lamp is lit.

- $4 < 10$
- $x < 10$
- The door is closed.
- The train is moving.
- The switch is on.
- I am in Rome.

In all the above examples one cannot further analyze the sentence in terms of propositional operations. Simply put: the words "not", "and", "or", "if...then", and "if and only if" do not occur in them.

The act of giving states of affairs, such as the above, a name such as p_n, is very much like denoting the speed of an object by v and mass by m, or denoting the length of a trip by a and time spent by t. The only difference is that the proposition symbols $p_0, p_1, ...$ do not denote magnitudes, such as speed, mass, etc, but states of affairs, whether something holds or not. This is the difference between logic and mathematics in general. In logic the fundamental concept is that of a proposition holding or not holding, being true of false. in mathematics the fundamental concept is that of a magnitude, something that can be measured, typically by real numbers. In the mid 19th century George Boole made the momentous invention that propositional logic can actually be seen as the mathematics of the numbers 1 (for "true") and 0 (for "false").

1.2.8 Negation explained

The negation of A, denoted $\neg A$, is intuitively simply the denial of A. Here are some examples:

- $\neg$ It rains: It is not the case that it rains; it does not rain.
- $\neg\ 4 > 10$: 4 is not greater than 10.
- $\neg$ The door is closed: The door is not closed.

In natural language negation is not as simple as in the formal language of propositional logic. In propositional logic, if we want to deny a sentence, we just put a negation sign in front of the sentence. Not so in natural language. In natural language you have to find the proper place for the negation. It would perhaps be understandable, if we said "It is not the case that the door is closed", but "The door is not closed" is how it is said.

Note that the negation of "$\neg$ The door is closed" is "$\neg\neg$ The door is closed". If we denote "The door is closed" by p_0, then "$\neg$ The door is closed" is $\neg p_0$ and "$\neg\neg$ The door is closed" is $\neg\neg p_0$. We may have the temptation to give "$\neg\neg$ The door is closed" the intuitive meaning "The door is closed", because we may think that the negations cancel each other out. This is how it is in natural language. If I say "The door is not closed" and you say that actually that is not the case, we both realize that the door is indeed closed. However, we are making here already logical conclusions. If we abstain from logical conclusions, $\neg\neg p_0$ is what it is—a proposition symbol with two negations in front of it. Only when we define meaning of propositional formulas we may conclude that $\neg\neg p_0$ and p_0 have the same meaning. This is an important point, because we could also choose to define the meaning of negation somewhat differently—as is done in so-called *intuitionistic* logic—and then $\neg\neg p_0$ and p_0 would *not* have the same meaning.

1.2.9 Conjunction explained

The conjunction of A and B, denoted $A \wedge B$, says intuitively that both A and B hold. The sentences A and B are called the *conjuncts*. So a conjunction $A \wedge B$ is the conjunction of two conjuncts. Here are some typical conjunctions:

- It rains and it blows.
- $4 < 10$ and $7 < 3$.
- The door is closed and the train is moving.

Conjunction is the easiest of our logical operations. It is really unproblematic, and we quickly move to disjunction, which provides formidable challenges.

1.2.10 Disjunction explained

The disjunction of A and B, denoted $A \vee B$, says intuitively that one of A and B holds or perhaps both hold. The sentences A and B are called the *disjuncts* of the disjunction.

- I am in Rome or I have lost my way.
- $4 < x$ or $y < 3$.
- The train is at a station or the train is moving.

There are many subtleties involved in the disjunction, as we will see later. Let us at this point only note that disjunction can, in principle, be used in two different meanings: The first is the *inclusive* meaning which allows both disjuncts to be true, as in the intended meaning of the sentence "You can sit next to an English-speaker or you can sit next to a logician." Well, conceivably an English-speaker can also be a logician. The other reading of disjunction is the *exclusive* meaning which does not allow both disjuncts to be true as in the intended meaning of the sentence "I can carry two small bags or else I can carry one large bag". We favor the inclusive reading when the choice is not clear. Accordingly, when we give a mathematical definition of the meaning of disjunction, it is going to be the inclusive meaning.

1.2.11 Implication explained

The implication from A to B, denoted $A \to B$, says intuitively that if A then B. Note that this says nothing about B in the case that A is false. A is called the antecedent, or hypothesis of $A \to B$. B is called the consequent, or conclusion of $A \to B$. When implication is used in mathematical or natural language it has the meaning that there is something in A which is sufficient for the truth of B, or that A *causes* B. But in propositional logic we take the very simple approach to implication that the truth of $A \to B$ is completely determined by the truth or falsity of A and the truth or falsity of B with no regard to whether there is some connection between the two. The fact that this does not quite fit our everyday use of "If...then" is the price we pay for using the formal language of propositional logic. There are more involved formal languages which avoid this but then one pays other prices.

Let us look at some examples:

- *If it rains then the streets are wet.* Note that this sentence says nothing about the wetness of streets when it does not rain. Maybe the streets are dry or maybe people are hosing their gardens and the streets are wet. The sentence in question makes no claims about those situations. All it says is that if it *happens* to be raining, then the streets are wet.
- *If* $x < 3$ *then* $x < 10$. This seems like a true statement about the integers. Of course, depending on x, it may be that $x < 10$ even though it is not the case that $x < 3$. The sentence makes no claims about such x that do not satisfy $x < 3$. Even though we know that in such cases it is still possible that $x < 10$, this sentence does not claim that. The sentence is only concerned with the cases that $x < 3$. All other situations are irrelevant from the point of view of this sentence.
- *If the train is moving then the door is closed.* This sentence calls for a more careful analysis:

Let p_0 be the atomic proposition

"The train is moving"

and p_1 the atomic proposition

"The door is closed".

Then $p_0 \to p_1$ says

"If the train is moving, then the door is closed".

Note that $p_0 \to p_1$ makes no claim about situations in which the train is not moving. We can imagine that the computer inside the train is following some indicators all the time, or let's say once every 3 seconds, making sure that certain safety constraints are met, one of which is $p_0 \to p_1$. The computer checks every 3 seconds whether the train is moving. If the train is not moving, the constraint $p_0 \to p_1$ gives no reason for action and the computer moves to the next constraint. This happens every 3 seconds, until the train starts moving. Then the computer detects the movement and rushes to make sure the door—or rather the doors–are closed. The computer simply makes sure that every 3 seconds the implication $p_0 \to p_1$ is true. For the computer the meaning of $p_0 \to p_1$ is: If p_0 is false, move on, but if p_0 is true, then make sure also p_1 is true. This is the genuine meaning of implication in propositional logic.

1.2.12 Equivalence explained

The equivalence of A and B, denoted $A \leftrightarrow B$, says intuitively that A holds if and only if B holds. Here are some examples of its uses.

- The lamp is lit if and only if the switch is on.
- $x < 10$ if and only if $x + 5 < 15$.
- The door is locked if and only if the train is moving.

In each case the intuitive meaning of $A \leftrightarrow B$ is that either both A and B are true or else they are both false. Equivalence is a relatively unproblematic logical operation, nothing like disjunction and implication.

1.2.13 Main connective

The main connective is a useful technical notion related to formulas. Locating the main connective is quite easy.

The main connective of

$\neg A$	is negation	$\neg$
$A \wedge B$	is conjunction	$\wedge$
$A \vee B$	is disjunction	$\vee$
$A \rightarrow B$	is implication	$\rightarrow$
$A \leftrightarrow B$	is equivalence	$\leftrightarrow$

For example, the main connective of $(A \vee \neg B) \rightarrow (C \vee D)$ is implication.

Logical analysis of a formula usually starts by identifying its main connective. Therefore the first thing a logician does upon seeing a formula is to locate its main connective, and depending on this connective, the logician decides on further action. Our rules concerning the use of parentheses guarantee that every formula has a unique main connective, that is, it does not happen that one tries to find the main connective and once it is conjunction, but another time with the same formula it is disjunction.

1.2.14 Subformula

Another useful technical notion related to formulas is that of a *subformula*. Intuitively, the subformulas of a given formula are the smaller formulas that the given formula is built from. We accept the formula itself as its own subformula.

The subformulas of

p_n	are	p_n itself only
$\neg A$	are	$\neg A$ and the subformulas of A
$A \wedge B$	are	$A \wedge B$ and the subformulas of A and B
$A \vee B$	are	$A \vee B$ and the subformulas of A and B
$A \rightarrow B$	are	$A \rightarrow B$ and the subformulas of A and B
$A \leftrightarrow B$	are	$A \leftrightarrow B$ and the subformulas of A and B

For example, the subformulas of

$$(p_0 \vee \neg p_1) \rightarrow (p_2 \wedge p_1)$$

are: $(p_0 \vee \neg p_1) \rightarrow (p_2 \wedge p_1)$, $p_0 \vee \neg p_1$, $p_2 \wedge p_1$, p_0, $\neg p_1$, p_2, and p_1.

Special among subformulas are *immediate* subformulas. They are just as easy to find as the main connective, discussed above.

The immediate subformulas of

p_n	are	none
$\neg A$	is	A
$A \wedge B$	are	A, B
$A \vee B$	are	A, B
$A \rightarrow B$	are	A, B
$A \leftrightarrow B$	are	A, B

Example: The immediate subformulas of $(p_0 \vee \neg p_1) \rightarrow (p_2 \wedge p_1)$ are $(p_0 \vee \neg p_1)$ and $(p_2 \wedge p_1)$.

Thus a propositional formula is always formed by connecting the immediate subformulas by the main connective. This sound so easy that one may ask is this knowledge of any use. Yes it is, a whole lot of use. All fundamental methods in logic are based on analyzing formulas in terms of their immediate subformulas. We learn several of them:

- Evaluating truth
- Natural deduction
- Semantic tree

So it is worth being comfortable with the concepts of main connective and immediate subformula. This ends the first lecture on propositional logic. In the next lecture we shall discuss truth tables.

1.2.15 Solved problems

Let's try to solve some simple problems related to the concepts that we have learnt.

Problem 1 *Write the sentence*

"If it rains, then I will not go out without an umbrella." as a propositional formula, and indicate its main connective.

Solution: The atomic parts are "it rains", and "I will go out without an umbrella". These are atomic, in the context of propositional logic, because we cannot build them from smaller parts by means of "not", "and", "or", "if...then", and "...if and only if..."

The sentence is of the form $p_0 \to \neg p_1$. Its main connective is implication.

We see here clearly that propositional logic simplifies natural language sentences substantially. But in propositional logic we only want to get the *logical structure* of the sentence, and even so, only the structure that pertains to the structure that *propositional logic* is capable of handling. □

Problem 2 *Write the sentence*

"If $y' = ax + by$, then $y'' = ax' + by'$" as a propositional formula, and indicate its main connective.

Solution: The atomic parts are $y' = ax + by$, and $y'' = ax' + by'$. These atomic parts clearly have their internal structure as equations but this structure is not one that propositional logic focuses on.

The sentence is of the form $p_0 \to p_1$. Its main connective is implication.□

Problem 3 *Write the sentence*

"If the door is not closed or the train is at a station, the green light is on." as a propositional formula and indicate its main connective and immediate subformulas.

Solution: This is a good example of the use of implication in natural language. The atomic parts of the sentence are

p_0 : the door is closed

p_1 : the train is at a station

p_2 : the green light is on

The sentence is of the form $(\neg p_0 \vee p_1) \to p_2$. Its main connective is implication and its immediate subformulas are $(\neg p_0 \vee p_1)$ and p_2. □

Problem 4 *Write the sentence*

"Either A buys Z and Z does not buy U, or else B buys both Z and U." as a propositional formula, and indicate its main connective and immediate subformulas.

Solution: Solution. The atomic parts are

p_0 : A buys Z

p_1 : Z buys U

p_2 : B buys Z

p_3 : B buys U

The sentence is of the form $(p_0 \wedge \neg p_1) \vee (p_2 \wedge p_3)$. Its main connective is disjunction and its immediate subformulas are $(p_0 \wedge \neg p_1)$ and $(p_2 \wedge p_3)$.

Here we have used "either ..., or else ... both ..." in natural language to indicate where the parentheses should go, because the mere

" A buys Z and Z does not buy U or B buys Z and B buys U."

would be ambiguous, or at least could be ambiguous.

□

1.2.16 Problems

Problem 5 *Which of the following are propositional formulas?*

1. *If the lamp is on, the room is lit.*
2. *The room is dark, if the lamp is broken.*
3. *I declare you man and wife.*
4. *The priest declared them man and wife.*
5. *Let x be a real number.*
6. *x is a real number.*

Problem 6 *Which of the following are propositional formulas?*

1. *Unless the lamp is on, the room is dark.*
2. *The room is dark, only if the lamp is broken.*
3. *I declared you man and wife.*
4. *The priest, the man, the wife.*
5. *g is the derivative of f.*
6. *Suppose g is the derivative of f.*

Problem 7 *Which of the following are unambiguous propositional formulas?*

1. $(p_2 \leftrightarrow \neg p_1) \rightarrow (p_2 \rightarrow p_3)$
2. $p_0 \wedge \neg p_1 \vee (p_2 \wedge p_1)$
3. $\neg(p_0 \rightarrow (p_1 \rightarrow (p_2 \rightarrow p_3)))$
4. $\neg\neg p_0 \rightarrow p_0$

Problem 8 *Which of the following are unambiguous propositional formulas?*

1. $p_2 \leftrightarrow p_1 \leftrightarrow p_2$
2. $p_0 \vee \neg p_1 \vee (p_2 \rightarrow p_0)$
3. $(p_0 \wedge p_1) \rightarrow p_0 \wedge (p_0 \wedge p_1) \rightarrow p_1$
4. $(p_0 \wedge \neg p_1 \wedge p_2) \vee (p_0 \wedge p_1 \wedge \neg p_2)$

Problem 9 *The main connective of*
$(p_2 \leftrightarrow \neg p_1) \rightarrow (p_2 \rightarrow p_3)$
is

1. *Negation*
2. *Conjunction*
3. *Disjunction*
4. *Implication*
5. *Equivalence?*

Problem 10 *The main connective of*
$\neg((p_2 \vee \neg p_1) \wedge (p_2 \vee p_3))$ *is*

1. *Negation*
2. *Conjunction*
3. *Disjunction*
4. *Implication*
5. *Equivalence?*

Problem 11 *The main connective of*
$(p_2 \wedge \neg p_1) \vee (p_2 \wedge p_3)$ *is*

1. *Negation*
2. *Conjunction*
3. *Disjunction*
4. *Implication*
5. *Equivalence?*

Problem 12 *The main connective of*
$((p_2 \rightarrow \neg p_1) \vee ((p_2 \vee p6) \wedge p_3)) \rightarrow p_3$ *is*

1. *Negation*
2. *Conjunction*
3. *Disjunction*
4. *Implication*
5. *Equivalence?*

Problem 13

- *Write the sentence*

  "If mn is even, then m is even or n is even."

  as a propositional formula and indicate the main connective and immediate subformulas.

- *Write the sentence*

  "a is a prime if and only if $a2 + 6a + 1$ is a prime."

  as a propositional formula and indicate the main connective and immediate subformulas.

Problem 14

- *Write the sentence*

  "The gate was not open and the lights were not on."

  as a propositional formula and indicate the main connective and immediate subformulas.

- *Write the sentence*

  "If either A buys both Z and U or else A buys V, then B does not sell W."

  as a propositional formula and indicate the main connective and immediate subformulas.

Problem 15

- *Write the sentence*

  "If the door is open and the train is not moving, the green light is on."

  as a propositional formula and indicate the main connective and immediate subformulas.

- *Write the sentence*

  "If the file can be accessed then either John can access it or Bob can access it but not both."

  as a propositional formula and indicate the main connective and immediate subformulas.

Problem 16

- *Write the sentence*

  "We go to the beach or to the park, unless it rains."

  as a propositional formula and indicate the main connective and immediate subformulas.

- *Write the sentence*

  "The train moves only if doors are closed and the security lamp is green."

  as a propositional formula and indicate the main connective and immediate subformulas.

Problem 17 *Write the sentence*

"You are not only able to leave comments on this photo, but you can also upload your own photos in our photo section and participate on our blogs."
as a propositional formula and indicate the main connective and immediate subformulas.

Hint: Do not worry if you cannot replicate in propositional logic all the stylistic aspects of the sentence. Just capture the essential structure.

Problem 18 *Write the sentence*

"Thick snow covered everything, but houses could be clearly seen, and here and there one could see a car."
as a propositional formula and indicate the main connective and immediate subformulas.

Hint: Do not worry if you cannot replicate in propositional logic all the stylistic aspects of the sentence. Just capture the essential structure.

1.3 Truth table

1.3.1 Truth values

Atomic propositional formulas, that is, the mere proposition symbols, denote states of affairs, such as

- It is raining.
- The train is moving.

Such states of affairs hold or not—are true or false—depending on the circumstances. One day it rains, another it does not. It may rain in Helsinki but not in Warsaw. The train may be moving or not. We call truth and falsity *truth values* and denote them by the numbers 1 and 0. The more complex propositional formulas $\neg A$, $A \wedge B$, $A \vee B$, etc denote likewise states of affairs, only more complicated ones. Intuitively they are also true or false, depending on the circumstances, so they, too, have a truth value, 1 or 0.

1.3.2 Truth values contd.

Propositional formulas of the simplest form $p_0, p_1, ...$ can have truth value 1 or 0 according to our choice. But if we give them truth values, then the truth values of formulas built from them such as $p_0 \vee p_1$ and $p_0 \rightarrow p_1$ will be completely determined by the rules that we give. So giving truth values to $p_0, p_1, ...$ is really the crucial thing, it decides everything else. A choice of truth values for the proposition symbols is called a *valuation*.

If we use propositional logic to investigate everyday phenomena around us, we may take propositional symbols to denote such states of affairs as "It is raining" or "The train is moving". Then a natural valuation would give "It is raining" the truth value 1 if it indeed is raining and 0 otherwise, and to "The train is moving" the truth value 1 if the train is indeed moving and 0 otherwise. However, the point of propositional logic is *not* to investigate phenomena around ourselves *as they are*, but rather *as they could be*. It is raining but it could also be the case that it is not. The train is moving but it could be also standing. In fact, in propositional logic we study propositional formulas under any valuation what so ever. This freedom to change the truth values of proposition symbols in every which way is the essence of propositional logic. It is what distinguishes logic from empirical science. In empirical science one builds on observed facts, which means, in a sense, that the valuation of the atomic propositions is fixed. Mind you, this is not quite true of quantum physics. Also, in scientific tests it may be essential to vary atomic propositions freely, for example, if we drop balls of the same size from the Leaning Tower of Pisa, we may want to vary the weight of the balls and the height from where the drop is made in order to see how the time of descent depends on the weight and the height.

1.3.3 Valuation

Valuations assign truth values 1 (true) or 0 (false) to proposition symbols. Thus a valuation is a function with proposition symbols as its domain and values in the set $\{0, 1\}$. We use v to denote a valuation function. Then $v(p_0), v(p_1), \dots$ are the values of $p_0, p_1, \dots$ in this valuation. In symbols

$$v(p_n) \in \{0, 1\}.$$

Valuations are the building blocks of *truth tables*, which we will next introduce. The truth tables constitute a systematic tool for analyzing complicated propositional formulas. They can be avoided, but they do help a lot.

1.3.4 Valuation Contd.

Valuations extend to all formulas by means of truth tables. This extension is heavily based on the analysis of a propositional formula in terms of its main connective and its immediate subformulas.

The truth value $v(A)$ of an arbitrary propositional formula A can be easily computed in terms of the truth values of the immediate subformulas of A and the truth table of the main connective.

1.3.5 Truth table

The *truth table* of a propositional formula is a table of truth values of the formula and its sub formulas under all possible valuations. Before we can build even the first truth table, we have to determine how truth values behave in the logical connectives:

- Conjunction
- Disjunction
- Negation
- Implication
- Equivalence

Truth tables reflect our intuition of the meaning of each connective.

1.3.6 Conjunction

Intuitively, a conjunction $A \wedge B$ is true if and only if both conjuncts A and B are true. This intuition is easily written into the following truth table:

A	B	$A \wedge B$
1	1	1
1	0	0
0	1	0
0	0	0

It is good to remember that the truth table of conjunction has exactly one row on which the conjunction has truth value 1.

In a truth table like this we allow A and B to be the same formula, but we want the table to describe the most general case and therefore we take into account the possibility that A and B can both get the value 1 or 0, independently of each other.

1.3.7 Disjunction

Intuitively, a disjunction $A \vee B$ is true if and only if one of the disjuncts A and B is true. What if both are true? We make the decision that we call such disjunctions true, as well. Thus when we now give meaning to the disjunction sign $\vee$, we give it the *inclusive* meaning, that is, both can be true in a true disjunction.

A	B	$A \vee B$
1	1	1
1	0	1
0	1	1
0	0	0

It is good to remember that the truth table of disjunction has exactly one row on which the disjunction has truth value 0, namely when both disjuncts have value 0.

1.3.8 Negation

Negation is a very simple case in this context. A negation $\neg A$ is true if and only if A is false. The negation is a kind of truth value switcher. It switches 1 to 0 and 0 to 1.

A	$\neg A$
1	0
0	1

1.3.9 Implication

We now have to decide what the truth value of $A \to B$ is, if we already know the truth values of A and B. Intuitively $A \to B$ should be true if B "follows" from A. However, with two truth values it is not really possible to express the "follows"-concept. We make the decision that $A \to B$ is true if B is true but also if A is false.

A	B	$A \to B$
1	1	1
1	0	0
0	1	1
0	0	1

It is good to remember that the truth table of implication has exactly one row on which the implication has truth value 0, namely when the hypothesis has value 1 but the conclusion has value 0.

There are more involved systems of propositional logic, which give a meaning to implication which is closer to the intuition of "follows". The choice we have made works surprisingly well. On the other hand, it renders such unintuitive implications as "If the Moon is cheese, then $2 + 2 = 5$" and "If $0 \neq 0$, then $2 + 2 = 4$" true.

1.3.10 Interpretation of implication

Let us look at a real life example of implication. Let p be the basic proposition "Jukka lives in Helsinki". Let q be the basic proposition "Jukka lives in Finland". Since Helsinki is in Finland, the formula $p \to q$ is true, whether Jukka lives in Helsinki or not. This truth is based on Helsinki being in Finland, and is unaffected if Jukka in fact does not live in Helsinki. The only thing that would shatter the truth of $p \to q$ is if Jukka lived in Helsinki but not in Finland, in which case we would have to review our geography.

1.3.11 Interpretation of implication contd.

One way to understand the truth table of implication is the following: Think of A and B as subsets of $\{0\}$. There are just two subsets: $\emptyset$ and $\{0\}$. Let us identify these two subsets with 0 and 1.

We can now think of $A \to B$ as " A is contained in B"

		A	B	$A \to B$
$\{0\} \subseteq \{0\}$	so 1st row is 1	1	1	1
$\{0\} \not\subseteq \emptyset$	so 2nd row is 0	1	0	0
$\emptyset \subseteq \{0\}$	so 3rd row is 1	0	1	1
$\emptyset \subseteq \emptyset$	so 4th row is 1.	0	0	1

1.3.12 Equivalence

Perhaps surprisingly, equivalence is easier to understand than implication. We give the equivalence $A \leftrightarrow B$ the truth value 1 exactly in the case that A and B have the *same* truth value.

A	B	$A \leftrightarrow B$
1	1	1
1	0	0
0	1	0
0	0	1

See Figures 1.2 and 1.3 for a summary of the valuations and truth tables of propositional formulas.

$v(p_n)$	=	0 or 1
$v(\neg A)$	=	0 if $v(A) = 1$, otherwise 1
$v(A \wedge B)$	=	0 if $v(A) = 0$ or $v(B) = 0$, otherwise 1
$v(A \vee B)$	=	1 if $v(A) = 1$ or $v(B) = 1$, otherwise 0
$v(A \to B)$	=	0 if $v(A) = 1$ and $v(B) = 0$, otherwise 1
$v(A \leftrightarrow B)$	=	1 if ($v(A) = 1$ and $v(B) = 1$),
		or ($v(A) = 0$ and $v(B) = 0$), otherwise 0

Figure 1.2: Valuation of propositional formulas

A	B	$\neg A$	$(A \wedge B)$	$(A \vee B)$	$(A \to B)$	$(A \leftrightarrow B)$
1	1	0	1	1	1	1
1	0	0	0	1	0	0
0	1	1	0	1	1	0
0	0	1	0	0	1	1

Figure 1.3: Truth tables of propositional formulas combined

1.3.13 Example

Let us now look at a full truth table. We consider the propositional formula

$$\neg(p_0 \vee p_1) \leftrightarrow (\neg p_0 \wedge p_1).$$

We write the proposition symbols that occur in the formula to the left most columns and form a row for each possible combination of truth values of the proposition symbols. To the right we form a column under each occurrence of a proposition symbol in the formula as well as a column under each connective.

Every connective in a formula is the main connective of some subformula. The truth value that we write under that connective is the truth value of that subformula.

First we fill in the proposition symbols. We just copy them from the columns on the left:

p_0	p_1	$\neg$	$(p_0$	$\vee$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\wedge$	$p_1)$
1	1		1		1			1		1
1	0		1		0			1		0
0	1		0		1			0		1
0	0		0		0			0		0

Then we fill the column of negation. That is easy—just swap zeros and ones:

p_0	p_1	$\neg$	$(p_0$	$\vee$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\wedge$	$p_1)$
1	1		1		1		0	1		1
1	0		1		0		0	1		0
0	1		0		1		1	0		1
0	0		0		0		1	0		0

Now we can fill in the conjunction. Note that the conjuncts of this conjunction are $\neg p_0$ and p_1. So we have to look at the columns corresponding to $\neg p_0$ and p_1.

p_0	p_1	$\neg$	$(p_0$	$\vee$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\wedge$	$p_1)$
1	1		1		1		0	1	0	1
1	0		1		0		0	1	0	0
0	1		0		1		1	0	1	1
0	0		0		0		1	0	0	0

Now the disjunction:

p_0	p_1	$\neg$	$(p_0$	$\vee$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\wedge$	$p_1)$
1	1		1	1	1		0	1	0	1
1	0		1	1	0		0	1	0	0
0	1		0	1	1		1	0	1	1
0	0		0	0	0		1	0	0	0

Next we deal with the negation on the left. Note that we cannot compute the column of the equivalence-symbol before we take care of the negation, because $\neg(p_0 \vee p_1)$ is an immediate subformula of the big formula.

p_0	p_1	$\neg$	$(p_0$	$\vee$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\wedge$	$p_1)$
1	1	0	1	1	1		0	1	0	1
1	0	0	1	1	0		0	1	0	0
0	1	0	0	1	1		1	0	1	1
0	0	1	0	0	0		1	0	0	0

Finally the equivalence:

p_0	p_1	$\neg$	$(p_0$	$\vee$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\wedge$	$p_1)$
1	1	0	1	1	1	1	0	1	0	1
1	0	0	1	1	0	1	0	1	0	0
0	1	0	0	1	1	0	1	0	1	1
0	0	1	0	0	0	0	1	0	0	0

The truth table of the given formula is ready. We can see that the formula gets value 1 if p_0 has value 1 and otherwise it gets value 0. This can be a very useful piece of information. All that complicated sentence is trying to say is that p_0 is true.

1.3.14 Example

Here is a bigger truth table. This is built just like the previous one, column by column, starting from the proposition symbols, goin up to bigger and bigger subformulas, always inserting the truth value of the subformula beneath its main connective:

p_0	p_1	p_2	$((p_0$	$\to$	$p_1)$	$\wedge$	$(p_1$	$\to$	$p_2))$	$\to$	$(p_0$	$\to$	$p_2))$
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	0	0	1	1	0	0
1	0	1	1	0	0	0	0	1	1	1	1	1	1
1	0	0	1	0	0	0	0	1	0	1	1	0	0
0	1	1	0	1	1	1	1	1	1	1	0	1	1
0	1	0	0	1	1	0	1	0	0	1	0	1	0
0	0	1	0	1	0	1	0	1	1	1	0	1	1
0	0	0	0	1	0	1	0	1	0	1	0	1	0

At first look it may seem that the propositional formula in question is incomprehensible, as such a long formula might very well be. As it happens, this formula seems to state a kind of transitivity of implication: If p_0 implies p_1, and p_1 implies p_2, then p_0 implies p_2. Be it as it may, our task here is to fill in the truth table quite mechanically with no concern to any exterior meaning. But we can make the interesting observation that the formula in question gets the truth value 1 whatever the valuation is. We have a special name for such formulas ("tautology") and we will pay special attention to them in a while.

1.3.15 Inefficiency of truth tables

Truth tables become eventually too large. If we have n proposition symbols, we are going to have 2^n rows in the truth table. Thus the truth table grows exponentially in the number of proposition symbols. This limits the applicability of the truth table method in practical applications. During this course we learn two other methods to analyze propositional formulas. Those other methods will be a bit more complicated but at the same time quite a bit more useful.

1.3.16 Solved problems

Problem 19 *Suppose* $v(p_0) = 1$ *and* $v(p_1) = 0$. *Calculate* $v(\neg(p_0 \vee p_1))$*: The truth values of* p_0 *and* p_1 *are given. So we can compute the truth value of their disjunction. Then we compute the truth value of the negation of the disjunction, and we are done.*

$\neg$	$(p_0$	$\vee$	$p_1)$
0	1	1	0

The answer is 0.

Problem 20 *Suppose* $v(p_0) = 1$ *and* $v(p_1) = 0$. *Calculate* $v(\neg p_0 \to p_1)$*: The truth values of* p_0 *and* p_1 *are given. We can immediately calculate the truth value of* $\neg p_0$. *Then we compute the truth value of the implication* $\neg p_0 \to p_1$, *and we are done.*

$\neg$	p_0	$\to$	p_1
0	1	1	0

The answer is 1.

Problem 21 *Suppose* $v(p_0) = 1$, $v(p_1) = 0$ *and* $v(p_2) = 1$. *Calculate* $v(p_0 \wedge (p_1 \vee p_2))$*: The truth values of* p_0, p_1 *and* p_2 *are given. We can immediately calculate the truth value of the disjunction. Then we compute the truth value of the conjunction, and we are done.*

p_0	$\wedge$	$(p_1$	$\vee$	$p_2)$
1	1	0	1	1

The answer is 1.

Problem 22 *Draw the truth table of* $p_0 \leftrightarrow \neg p_1$. *There are two proposition symbols so we have four rows in the truth table. We first fill in the columns of the proposition symbols by simply copying from the left. Then we fill in the column of the negation, and finally of the equivalence.*

p_0	p_1	p_0	$\leftrightarrow$	$\neg$	p_1
1	1	1	0	0	1
1	0	1	1	1	0
0	1	0	1	0	1
0	0	0	0	1	0

Problem 23 *Draw the truth table of* $\neg p_0 \vee p_1$. *There are two proposition symbols so we have four rows in the truth table. We first fill in the columns of the proposition symbols by simply copying from the left. Then we fill in the column of the negation, and finally the column of the disjunction.*

p_0	p_1	$\neg$	p_0	$\vee$	p_1
1	1	0	1	1	1
1	0	0	1	0	0
0	1	1	0	1	1
0	0	1	0	1	0

Problem 24 *Draw the truth table of* $\neg(p_0 \wedge p_1) \leftrightarrow (\neg p_0 \vee p_1)$. *There are two proposition symbols so we have four rows in the truth table. We first fill in the columns of the proposition symbols by simply copying from the left. Then we fill in the column of the conjunction. Next the negation on the left. Then the negation on the right. Then the remaining disjunction. And finally the equivalence. Note that the order in which we proceed allows some variation. For example, after we have filled in the proposition symbol columns we could fill in the negation on the right, rather than that of the conjunction. However, one has to be careful with the order. A wrong order usually yields a wrong answer.*

p_0	p_1	$\neg$	$(p_0$	$\wedge$	$p_1)$	$\leftrightarrow$	$(\neg$	p_0	$\vee$	$p_1)$
1	1	0	1	1	1	0	0	1	1	1
1	0	1	1	0	0	0	0	1	0	0
0	1	1	0	0	1	1	1	0	1	1
0	0	1	0	0	0	1	1	0	1	0

1.3.17 Problems

Problem 25 *Suppose* $v(p_0) = 1$, $v(p_1) = 1$ *and* $v(p_2) = 0$ *Compute* $v(p_0 \to (p_1 \to p_2))$.

Problem 26 *Suppose* $v(p_3) = 1$, $v(p_6) = 0$ *and* $v(p_9) = 0$ *Compute* $v((p_3 \to p_9) \lor (p_6 \to p_9))$.

Problem 27 *Suppose* $v(p_0) = 1$, $v(p_1) = 0$ *and* $v(p_2) = 0$ *Compute* $v((p_0 \lor p_1) \leftrightarrow (p_2 \land \neg p_0))$.

Problem 28 *Suppose* $v(p_0) = 1$, $v(p_1) = 0$ *and* $v(p_2) = 0$ *Compute* $v(\neg(\neg p_0 \to \neg p_1) \lor \neg(\neg p_1 \to \neg p_2))$.

Problem 29 *Fill in the missing values:*

p_0	p_1	$\neg$	$(p_0$	$\lor$	$p_1)$	$\land$	$(\neg$	p_0	$\land$	$\neg$	$p_1)$
1	1	0	1	1	1	0	0	1	0	0	1
1	0	0	1	1	0	0	0	1	0	1	0
0	1		0	1	1		1	0		0	1
0	0		0		0			0			0

Problem 30 *Fill in the missing values:*

p_0	p_1	$\neg$	$(p_0$	$\land$	$p_1)$	$\lor$	$(\neg$	p_0	$\to$	$\neg$	$p_1)$
1	1	0	1	1	1	1	0	1	1	0	1
1	0	1	1	0	0		0	1		1	0
0	1	1	0	0	1	1	1	0		0	1
0	0	1	0	0	0			0			0

Problem 31 *Prove*

1. $v(\neg A) = 1 - v(A)$
2. $v(A \land B) = v(A) \cdot v(B)$
3. $v(A \lor B) = v(A) + v(B) - v(A) \cdot v(B)$
4. $v(A \to B) = 1 - v(A) + v(A) \cdot v(B)$
5. $v(A \leftrightarrow B) = 1 - v(A) - v(B) + 2 \cdot v(A) \cdot v(B)$

Problem 32 *Exclusive disjunction is the connective " A or B but not both". Here are some examples of exclusive disjunction in everyday language If today is Friday, I am in Rome or I am in Madrid. You get your luggage back or the airline pays a full compensation. Draw the truth table of exclusive disjunction.*

Problem 33 *Let* M *be the set of all valuations. Let for each propositional formula* A: $[A] = \{v \in M : v(A) = 1\}$. *Show*

1. $[A \land B] = [A] \cap [B]$
2. $[A \lor B] = [A] \cup [B]$
3. $[\neg A] = M - [A]$
4. $[A \to B] = M$ *iff* $[A] \subseteq [B]$
5. $[A \leftrightarrow B] = M$ *iff* $[A] = [B]$

Problem 34 *Let* M *be the set of all valuations of* $p_0, \ldots, p_{n-1}$. *Consider propositional formulas* A *in proposition symbols* $p_0, \ldots, p_{n-1}$. *Let* $\#(A)$ *be the number of* v *in* M *such that* $v(A) = 1$. *Let* $p(A) = \#(A)/2^n$. *We call* $p(A)$ *the probability of* A. *Show*

1. $p(A \lor B) + p(A \land B) = p(A) + p(B)$
2. $p(\neg A) = 1 - p(A)$

Problem 35 *Recall that* $p(A)$ *the probability of* A.

1. *All else being equal, which of the following is more probable? (In other words: which has a greater probability):*
   - $p_0 \land p_1$
   - $p_0 \to p_1$
2. *All else being equal, which of the following is more probable? (In other words: which has a greater probability):*
   - $(p_0 \land \neg p_1) \lor (p_1 \land \neg p_2)$
   - $p_0 \lor (p_1 \land p_2)$

Problem 36 *Suppose* X *is a set of valuations. We say that*

1. X *is of type* p_i *if* $v(p_i) = 1$ *for all* v *in* X.
2. X *is of type* $\neg p_i$ *if* $v(p_i) = 0$ *for all* v *in* X.
3. X *is of type* $A \land B$ *if it is of type* A *and of type* B.
4. X *is of type* $A \lor B$ *if* $X = Y \cup Z$, *where* Y *is of type* A *and* Z *is of type* B.
5. X *is of type* $\neg(A \land B)$ *if it is of type* $\neg A \lor \neg B$.
6. X *is of type* $\neg(A \lor B)$ *if it is of type* $\neg A \land \neg B$.
7. X *is of type* $A \to B$ *if it is of type* $\neg A \lor B$.
8. X *is of type* $\neg(A \to B)$ *if it is of type* $A \land \neg B$.

9. *X is of type $A \leftrightarrow B$ if it is of type $(A \to B) \wedge (B \to A)$.*

10. *X is of type $\neg(A \leftrightarrow B)$ if it is of type $(A \wedge \neg B) \vee (\neg A \wedge B)$.*

Show that X is of type A iff $v(A) = 1$ for all v in X.

1.4 Using truth tables

1.4.1 Tautology

In everyday language a person utters a tautology if he or she says something which is true but only because of its form, such as "It rains or it doesn't rain". Ordinarily uttering a tautology sounds a bit silly. It is like saying something which has no content. On the other hand, in a conversation a person may say something which at first seems clever but in closer scrutiny turns out to be a tautology, such as "If Caesar built a bridge over the Rhine, then Caesar went to Gaul or Caesar built a bridge over the Rhine". Despite this negative reputation of tautologies in social contexts, it is of utmost importance to be able to distinguish between a tautology and a non-tautology. For large formulas this is so difficult that finding a good algorithm for it is a famous open problem (called the $P = NP$-problem[1]). Tautologies can help us squeeze information from known facts without acquiring new data. In simple cases like the above Caesar example, this may seem superfluous, but when the formulas are complex it is not so easy any more.

A propositional formula is a **tautology** if its truth value is 1 under any valuation. This can be checked with a truth table.

Example 1.1 *The following are examples of tautologies:*

$$(p_0 \vee p_1) \leftrightarrow (p_1 \vee p_0)$$

$$A \vee \neg A$$

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

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

$$(A \to B) \leftrightarrow (\neg A \vee B)$$

In each case one can draw the truth table to see that the formula is a tautology.

[1] `http://en.wikipedia.org/wiki/P_versus_NP_problem`

1.4.2 Satisfiability

A formula is **satisfiable** if its truth value is 1 under some valuation. This can be checked with a truth table.

Uttering something which is satisfiable is like saying something which could in principle be true. For example, if I say "It rains but it is not snowing", I may be right or wrong about the current weather but surely it could be the case that it rained and at the same time did not snow. Likewise, the sentence "The train is moving and the door is open" seems by all means satisfiable, for we can imagine a situation, perhaps undesirable, in which a train is moving and someone has left the door open. In propositional logic we can decide the satisfiability of a formula simply by examining the truth table of the formula.

Example 1.2 *The following are examples of satisfiable formulas:*

$$(p_0 \to p_1) \wedge \neg p_0 \wedge \neg p_1$$

$$(p_0 \vee (p_1 \vee p_2)) \wedge \neg p_0 \wedge \neg p_2$$

$$\neg(p_0 \vee p_1) \leftrightarrow (\neg p_0 \wedge p_1)$$

In each case one can draw the truth table to see that the formula is satisfiable.

1.4.3 Contingency

A formula is **contingent** if its truth value is 1 under some valuation and 0 under another valuation. This can be checked with a truth table.

In everyday language a contingency is something that may be the case but most likely is not the case. An airplane may lose one engine and still fly safely, but losing two engines may have catastrophic consequences. For such a contingency the pilots probably have special emergency routines to follow. In propositional logic a continent formula is one that is satisfiable—i.e. may be true—but also the negation is satisfiable—i.e. the formula may also be false, with no concern as to which possibility is more likely.

Example 1.3

$$(p_0 \to p_1) \wedge \neg p_0 \wedge \neg p_1$$

$$\neg(p_0 \to (p_0 \to p_1))$$

$$\neg(p_0 \vee p_1) \leftrightarrow (\neg p_0 \wedge p_1)$$

In each case one can draw the truth table to see that the formula is contingent.

1.4.4 Refutability

A formula is **refutable** if its truth value is 0 under some valuation. This can be checked with a truth table.

In everyday language a refutable statement is one which may happen to be true but can certainly be false, too, such as "If it rains, then it blows from the west". A teacher may yell at a student: "Either you are late, or you have left your books home!" But the student may reply: "Two days ago I was on time and I had all my books with me". So what the teacher had said was refutable. When someone makes a statement claiming that it is a universal truth, he or she may prompt a person present to point out that the statement can be refuted and therefore cannot be a universal truth. In propositional logic refutability of a formula has a technical meaning. It just means that there is at least one valuation (row in the truth table) which renders the formula false.

Example 1.4 *Here are some refutable propositional formulas:*

$$(p_0 \to p_1) \wedge \neg p_0 \wedge \neg p_1$$

$$\neg(p_0 \to (p_0 \to p_1))$$

$$\neg(p_0 \vee p_1) \leftrightarrow (\neg p_0 \wedge p_1)$$

In each case one can draw the truth table to see that the formula is refutable.

1.4.5 Contradiction

In everyday language a person utters a contradiction if he or she says something which is false merely because of its form, such as "It rains and it doesn't rain". The contradiction may be more hidden, as in "It is not that it rains or it blows, but it certainly rains." In a conversation a person may make a statement which others find controversial, and the statement may be under fire from all directions, but if the statement turns out to be contradictory, or in contradiction with something else that the person has just said, the statement is not taken seriously any more.

A propositional formula is a **contradiction**, or contradictory, if its truth value is 0 under any valuation. This can be checked with a truth table.

Example 1.5 *Here are some examples of contradictions:*

$$(p_0 \vee p_1) \wedge \neg p_1 \wedge \neg p_0$$

$$A \wedge \neg A$$

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

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

In each case one can draw the truth table to see that the formula is contradictory.

1.4.6 The categories of propositional formulas

Every propositional formula is either tautological, contradictory or contingent. This is simply because in the truth table of a propositional formula we have under the main connective—indicating the truth value of the formula itself—either a column of ones, a column of zeros or a column that has both ones and zeros. In the first case—all ones—we have a tautology, in the second case—all zeros—we have a contradiction, and in the third case—both ones and zeros—we have a contingency. There are no other possibilities.

Moreover, it follows immediately that every satisfiable formula is either a tautology or a contingency, and every refutable formula is either a contradiction or a contingency.

When a logician looks at a propositional formula, he or she immediately tries to figure out whether the formula is a tautology, a contradiction or a contingency. These are the basic qualities of propositional formulas, and knowing the quality of the formula gives a hint which way to proceed next with the formula.

1.4.7 Million dollar question

Now comes a hot question: Given a formula, can you decide in polynomial time whether it is a tautology, a contradiction or a contingency? **Polynomial time** means here the following: There is a number k such that you need to perform only n^k basic operations if the input formula has n symbols. One million dollars has been promised by the Clay Mathematical Institute in Toronto[2]. This question is also known as the $P = NP$-problem[3].

1.4.8 Inefficiency of Truth Tables

Why not use truth tables to decide whether a given propositional formula is a tautology, a contradiction or a contingency? This is perfectly possible, but it is not polynomial time. Truth tables become eventually too large. With n propositional symbols we have 2^n rows in the truth table. So truth tables grow exponentially in the number of proposition symbols. With 100 proposition symbols, which is quite feasible in industrial applications, we have more than 10^{30} rows in the truth table. Tricky!

1.4.9 Logical equivalence and logical consequence

Two propositional formulas A and B are called *(logically) equivalent* if $A \leftrightarrow B$ is a tautology. In other words, the formulas have the same truth value in every valuation. Equivalence of formulas is used in everyday language and in science all the time, often without paying much attention to it. For some well-known equivalences, see Figure 1.4.

A propositional formula B is a *logical consequence* of the propositional formula A if $A \rightarrow B$ is a tautology. In other words, in every valuation where A gets value 1 also B gets value 1. Just like logical equivalence, logical consequence is used in everyday language and in science all the time, often without paying much attention to it. For some simple logical consequences, see Figure **??**.

[2] http://www.claymath.org/millennium/

[3] http://en.wikipedia.org/wiki/P_versus_NP_problem

1.4.10 Solved problems

Problem 37 *Use the truth table method to decide whether the formula $\neg p_0 \vee \neg\neg p_0$ is a tautology, a contingency or a contradiction.*

The truth table has just two rows, so this is going to be easy. We first fill in the columns corresponding to the proposition symbol p_0. Then the three negations. And finally the disjunction:

p_0	$\neg$	p_0	$\vee$	$\neg$	$\neg$	p_0
1	0	1	1	1	0	1
0	1	0	1	0	1	0

It is a tautology.

Problem 38 *Use the truth table method to decide whether the formula $\neg p_0 \rightarrow p_1$ is a tautology, a contingency or a contradiction.*

We have four rows in the truth table. First we fill in the columns corresponding to the proposition symbols. Then the negation. And finally the implication:

p_0	p_1	$\neg$	p_0	$\rightarrow$	p_1
1	1	0	1	1	1
1	0	0	1	1	0
0	1	1	0	1	1
0	0	1	0	0	0

It is contingent.

Problem 39 *Use the truth table method to decide whether the formula $\neg(p_0 \rightarrow p_1) \wedge \neg p_0$ is a tautology, a contingency or a contradiction.*

Again we have four rows. First we fill in the columns corresponding to the proposition symbols. Then the implication. Next the two negations. And finally the conjunction:

p_0	p_1	$\neg$	$(p_0$	$\rightarrow$	$p_1)$	$\wedge$	$\neg$	p_0
1	1	0	1	1	1	0	0	1
1	0	1	1	0	0	0	0	1
0	1	0	0	1	1	0	1	0
0	0	0	0	1	0	0	1	0

It is a contradiction.

1.4.11 Problems

Problem 40 *Use the truth table method to solve the following problems:*

Formula	Equivalent formula	Name of the equivalence
$\neg(A \land B)$	$\neg A \lor \neg B$	De Morgan law
$\neg(A \lor B)$	$\neg A \land \neg B$	De Morgan law
$\neg\neg A$	A	Law of double negation
$A \rightarrow B$	$\neg A \lor B$	
$A \leftrightarrow B$	$(A \rightarrow B) \land (B \rightarrow A)$	
$A \lor (B \lor C)$	$(A \lor B) \lor C$	Associativity law of disjunction
$A \land (B \land C)$	$(A \land B) \land C$	Associativity law of conjunction
$A \land (B \lor C)$	$(A \land B) \lor (A \land C)$	Distributivity law
$A \lor (B \land C)$	$(A \lor B) \land (A \lor C)$	Distributivity law
$A \land A$	A	Law of idempotence of conjunction
$A \lor A$	A	Law of idempotence of disjunction

Figure 1.4: Logical equivalences.

Formula	Logical consequence
$A \land B$	A
$A \land B$	B
A	$A \lor B$
B	$A \lor B$
B	$A \rightarrow B$
$\neg A$	$A \rightarrow B$

Figure 1.5: Logical consequences.

1. *Decide whether the formula* $p_0 \to \neg p_0$ *is a tautology, a contingency, or a contradiction.*
2. *Decide whether the formula* $p_0 \vee \neg(p_0 \wedge p_1)$ *is a tautology, a contingency, or a contradiction.*

Problem 41 *Use the truth table method to solve the following problems:*

1. *Decide whether* $p_0 \to p_1$ *is equivalent to* $\neg(p_1 \to p_0)$ *or not.*
2. *Decide whether* $\neg p_0 \vee p_1$ *is equivalent to* $\neg(p_0 \wedge p_1)$ *or not.*

Problem 42 *Use the truth table method to prove the equivalence of*

1. $\neg(A \wedge B)$ *and* $\neg A \vee \neg B$
2. $\neg\neg A$ *and* A
3. $A \wedge A$ *and* A
4. $A \vee A$ *and* A

Problem 43 *Use the truth table method to prove the equivalence of*

1. $A \to B$ *and* $\neg A \vee B$
2. $A \leftrightarrow B$ *and* $(A \to B) \wedge (B \to A)$

Problem 44 *Use the truth table method to prove the equivalence of*

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

Problem 45 *Use the truth table method to prove the equivalence of*

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

Problem 46 *Use the truth table method to prove the equivalence of*

1. $A \to (B \to C)$ *and* $B \to (A \to C)$
2. $A \to (B \to C)$ *and* $(A \wedge B) \to C$

Problem 47 *Show that a conjunction of implications* $p_i \to p_{i^*}$ *is not a tautology if and only if there is an* i *such that* i *is not* i^*.

Problem 48 *Define for formulas built up using just* $\neg$, $\wedge$ *and* $\vee$*:*

1. $(p_i)^+ = p_i, (p_i)^- = \neg p_i$
2. $(\neg A)^+ = A^-, (\neg A)^- = A^+$
3. $(A \wedge B)^+ = A^+ \wedge B^+, (A \wedge B)^- = A^- \vee B^-$
4. $(A \vee B)^+ = A^+ \vee B^+, (A \vee B)^- = A^- \wedge B^-$

A^+ *is called the Negation Normal Form of* A. *It has negations only in front of proposition symbols.*

1. *Find* $(\neg((p_0 \wedge p_1) \vee p_2))^+$.
2. *Show that* A^+ *and* A *are equivalent.*
3. *Show that* A^- *and* $\neg A$ *are equivalent.*

Problem 49 *Define for formulas built up using just* $\neg$, $\wedge$ *and* $\vee$*, which are in negation normal form:*

1. $(p_i)^* = p_i$
2. $(\neg p_i)^* = \neg p_i$
3. $(A \wedge B)^* = A^* \vee B^*$
4. $(A \vee B)^* = A^* \wedge B^*$

The formula A^* *is the dual of* A. *It is obtained from* A *by switching conjunctions to disjunctions. What is the dual of* $(p_0 \vee \neg p_1) \wedge p_2$*? Show that if* A *is a tautology, so is* $\neg A^*$. *Show that if* A *and* B *are equivalent, then so are* A^* *and* B^*.

Problem 50 *Suppose* A *is a propositional formula in which the proposition symbols* $p_0, \ldots, p_{n-1}$ *occur. Let* A_i *be for each* $i = 0, \ldots, n-1$ *an arbitrary propositional formula. Let* A' *be the result of substituting everywhere in* A *the formula* A_i *for* p_i. *Show that if* A *is a tautology, then so is* A'.

Problem 51 *A topological interpretation is a topological space E and a function f from proposition symbols to open sets in E. We extend this to propositional formulas built up from $\wedge$, $\vee$ and $\neg$:*

1. $f(A \wedge B) = f(A) \cap f(B)$
2. $f(A \vee B) = f(A) \cup f(B)$
3. $f(\neg A) = Int(E - f(A))$*, the interior of the complement of $f(A)$ in the topological sense.*

Show that A is a tautology iff $f(A) = E$ in every topological interpretation where E is a discrete space. A is said to be a constructive tautology if $f(A) = E$ in every topological interpretation. Show that $\neg(A \wedge \neg A)$ is, but $A \vee \neg A$ is not a constructive tautology.

1.5 Truth functions

1.5.1 What are truth functions

Truth functions are generalizations of the familiar connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$. By focusing on the general concept of a truth function, rather than each individual connective separately, we can understand the nature of connectives better.

One of the main reasons to study truth functions is that *computers* are ultimately based on truth functions that are welded into microprosessors inside the computer. Also, truth functions have interesting mathematical properties, and truth functions turn out to have a close connection to propositional formulas.

1.5.2 Truth function

A truth function (also called a connective) is any function f from the set $\{0, 1\}^n$ to the set $\{0, 1\}$, for some n. A truth function of n variables is called n-ary. A 2-ary truth function is called binary. Truth functions can be identified with truth tables, because we can simply list all the values of a truth function. Note that there is a big difference to functions on infinite domains such as $\mathbb{N}$ and $\mathbb{R}$. A truth function has only finitely many arguments and we can simply make a list of all of them, together with the values that the truth function gets on those arguments. Such tables we call truth tables.

We have already defined the connectives

$$\neg, \wedge, \vee, \rightarrow, \leftrightarrow .$$

We identify these with the corresponding truth functions. So, a connective is a symbol with certain meaning but we can also think of it—indeed identify it with—a truth function. Negation is the truth function which maps 0 to 1 and 1 to 0. Conjunction is the binary truth function which maps $(1, 1)$ to 1, $(0, 1)$ to 0, $(1, 0)$ to 0, and $(0, 0)$ to 0. Similarly the other connectives.

1.5.3 More binary truth functions

Here are some examples of truth functions. In the first one

x	y	$f(x, y)$
1	1	1
1	0	1
0	1	1
0	0	1

the value of the function is constant 1. This may not be the most interesting truth function but it is a truth function all the same. In the second truth function

x	y	$f(x, y)$
1	1	0
1	0	1
0	1	1
0	0	0

the value of the function is 1 if the arguments x and y have a *different* value. The third truth function

x	y	$f(x, y)$
1	1	0
1	0	0
0	1	0
0	0	0

is again a constant function, this time constant 0. Again we leave aside the question how interesting this function is, because, from a mathematical point, all functions are equal. Our fourth truth function

x	y	$f(x, y)$
1	1	1
1	0	0
0	1	0
0	0	1

may look familiar: this is the truth table of the connective $\leftrightarrow$. Our familiar connectives pop up as truth functions but there are also others.

1.5.4 There are exactly 16 binary truth functions

Here is a table of all binary truth functions. The table starts on the left, after the column for the arguments x and y, with the rather boring constant 0 truth function. Then comes a function that is 1 only if both x and y are 0. After an odd one we can see the truth table of the negation of x. Then again an odd one and then the negation of y.

x	y				$\neg$		$\neg$			$\wedge$	$\leftrightarrow$		$\rightarrow$			$\vee$	
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

We skip over two functions and come to the truth table of conjunction, and immediately after that the truth table of equivalence. Then we have the identity function y, after which comes the important truth table of implication. Then we have the identity function x. After the truth table of $y \rightarrow x$ we find the truth table of disjunction, and finally the constant function 1.

The important connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ seem to appear somewhat nondescriptly in the table of all binary truth functions. Their important role in our everyday language does not jump on our face. In fact, many of the binary truth functions could have emerged in our language and overshadowed connectives such as disjunction and implication. Why this has not happened? When we proceed, we may—perhaps—get a clue what is so special about $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$.

If we made a similar table of all ternary (3-ary) truth functions, there would be much more cases simply because there are much more ternary functions than binary ones.

1.5.5 A ternary truth function

Here is an example of a ternary truth function.

x	y	z	$f(x,y,z)$
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	0

This truth function does not have any particular intuitive interpretation, but in a moment we will learn how to associate a propositional formula to each truth function. The propositional formula can then give a kind of intuitive picture of the truth function.

1.5.6 Sheffer stroke

Here is an important new connective, called *Sheffer stroke*:

A	B	$A \mid B$
1	1	0
1	0	1
0	1	1
0	0	1

This connective can be read in natural language as "not...and". When we learn about *definability* in a moment, we can see that Sheffer stroke has remarkable properties. It's importance is not in that people might use it in everyday language as "not ... and", but rather in its power to define every other connective, as we shall now see.

1.5.7 Definability of truth functions

Definability is an important concept in logic and occurs in many disguises. A truth function f is said to be *definable* in terms of truth functions $g_1, ..., g_m$ if f can be obtained by composition from $g_1, ..., g_m$. What this means in practice is the following:

Disjunction can be defined in terms of negation and conjunction:

$$A \vee B = \neg(\neg A \wedge \neg B).$$

Technically speaking, the connective $\vee$ can be obtained by composition from the connective $\neg$ and $\wedge$. Composition means roughly speaking the same as successive application, as in the above definition of $A \vee B$ we first apply negation to A and B, then conjunction to the result and then finally once again negation. Sheffer stroke $A|B$ can be defined in terms of negation and conjunction:

$$A|B = \neg(A \wedge B).$$

Negation and conjunction can be defined in terms of the Sheffer stroke:

$$\neg A = A|A$$

$$A \wedge B = (A|B)|(A|B)$$

So we see the interesting property of Sheffer stroke that negation and conjunction can be defined in terms of it. Since disjunction can be defined in terms of negation and conjunction we may conclude that disjunction too can be defined in terms of the Sheffer stroke. This interesting property of the Sheffer stroke is called *universality*.

1.5.8 Universal sets of connectives

A set T of truth functions is universal if every truth function can be defined in terms of functions in T. A function f is universal if the set $\{f\}$ is. It turns out—we will give an argument to this effect in a moment—that every truth function can be defined in terms of the Sheffer stroke, i.e. the Sheffer stroke is a universal connective. Microprocessors are built from "gates" that are essentially connectives. It suffices to manufacture Sheffer stroke (also called NAND) gates as all others can be built from them. The sets $\{\neg, \wedge\}, \{\neg, \vee\}, \{\neg, \rightarrow\}$ are also universal. All this needs, of course, to be proved. So let us move on.

1.5.9 Propositional formulas define truth functions

Suppose A is a propositional formula built from proposition symbols $p_1, ..., p_n$. A defines the following truth function:

$f_A(x_1, ..., x_n)$ = the truth value of A under the valuation that gives p_i the value x_i for $i = 1, ..., n$

For example,

$f_{\neg p_1}(x_1)$ is the truth value of $\neg p_1$ under the valuation $v(p_1) = x_1$. So $f_{\neg p_1}(x_1)$ is 1 if $x_1 = 0$ and 0 if $x_1 = 1$.

Also, $f_{p_1 \wedge p_2}(x_1, x_2)$ is the truth value of $p_1 \wedge p_2$ under the valuation $v(p_1) = x_1$, $v(p_2) = x_2$. So $f_{p_1 \wedge p_2}(x_1, x_2)$ is 1 if $x_1 = x_2 = 1$ and 0 otherwise.

Finally, $f_{\neg p_1 \vee p_2}(x_1, x_2)$ is the truth value of $\neg p_1 \vee p_2$ under the valuation $v(p_1) = x_1$, $v(p_2) = x_2$. So $f_{\neg p_1 \vee p_2}(x_1, x_2)$ is 0 if (as we can see after a little calculation) $x_1 = 1$ and $x_2 = 0$, and 1 otherwise.

Whatever propositional formula A is given, we can quite easily define the truth function f_A by simply building the truth table of A. This turns out to be a very useful method for creating truth functions. In fact so useful that every truth function is of the form f_A for some A.

1.5.10 Propositional formulas cover all truth functions

We shall now indicate—without giving a rigorous proof—that every truth function is of the form f_A for some propositional formula A.

Theorem: Every truth function is defined by some propositional formula.

Let us look at the truth table of the truth function f and focus on the rows where f gets value 1. We simply forget the rows where f gets value 0. We represent f as the "disjunction" of those rows.

To see how this is done we consider an example:

x	y	z	$f(x,y,z)$
1	1	1	1
1	1	0	0
1	0	1	1
1	0	0	0
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	0

There are 3 rows where f gets the value 1. Let us call these *1-rows*. Now we use a trick: we know that a disjunction of several formulas has truth value 1 exactly when at least one of the formulas has. So we look for a disjunction of three formulas—because there are three ones—such that each disjunct picks exactly one of the three 1-rows.

1.5.11 Capturing a truth function with a formula

We look for a propositional formula the truth table of which has 1 on exactly the same rows as f, i.e. on the 1-rows.

$$\begin{aligned} A \;=\; & (p_0 \wedge p_1 \wedge p_2) \vee \\ & (p_0 \wedge \neg p_1 \wedge p_2) \vee \\ & (\neg p_0 \wedge p_1 \wedge \neg p_2) \end{aligned}$$

This choice works wonderfully. Let us try: On the first 1-row of the truth table of f we have $x = y = z = 1$. This corresponds to the valuation $v(p_0) = v(p_1) = v(p_2) = 1$. With this valuation the first (top) disjunct $p_0 \wedge p_1 \wedge p_2$ and hence the whole disjunction A becomes true. What about the second 1-row. Here we have $x = z = 1$, $y = 0$. This corresponds to the valuation $v(p_0) = v(p_2) = 1$, $v(p_1) = 0$. And, surprise, surprise, the second (middle) disjunct is true with this valuation. Finally, the third 1-row likewise gives rise to a valuation which renders A true.

But, now comes the catch, what about the other rows, all those rows where f gets value 0. What happens to A on those rows. Let us assume we have a valuation which gives A the value 1. Then it gives value 1 to at least one disjunct, say the last one $\neg p_0 \wedge p_1 \wedge \neg p_2$. Hence $v(A) = 1$ for the valuation which satisfies $v(p_0) = v(p_2) = 0$, $v(p_1) = 1$. But this is the third 1-row of the truth table of f. Going through all the cases where $v(A)$ can be 1 one can be convinced that, indeed, $f = f_A$.

If there are no 1-rows, we let A be $p_0 \wedge \neg p_0$.

1.5.12 Applications

A side-result of the trick we just learnt for representing every truth function in the form f_A is that **every truth function can be defined by means of $\neg$, $\wedge$ and $\vee$.** Namely, these are the only connectives that occur in the A that we constructed. Thus $\{\neg, \wedge, \vee\}$ is a universal set of connectives, even just $\{\neg, \wedge\}$, because $\vee$ can be defined in terms of $\neg$ and $\wedge$. Moreover, every propositional formula A gives rise to a truth function f_A and then for this f_A we can find a propositional formula B by using just $\neg$, $\wedge$ and $\vee$ with $f_A = f_B$. So now A and B have the same truth table. We call such formulas *logically equivalent*. Thus, every propositional formula A can be expressed in a logically equivalent form

$$A_1 \vee ... \vee A_n,$$

where each A_i is of the form

$$B_1 \wedge ... \wedge B_m,$$

and each B_i is a proposition symbol or its negation. This is called a *disjunctive normal form* (denoted DNF) of A. It is not at all unique, the one and same A may have the same truth table with many different formulas that are all in disjunctive normal form.

For example, a mechanically built disjunctive normal form of $p_0 \rightarrow p_1$ is

$$(p_0 \wedge p_1) \vee (\neg p_0 \wedge p_1) \vee (\neg p_0 \wedge \neg p_1).$$

However, this is not at all the simplest way to represent $p_0 \rightarrow p_1$ in disjunctive normal form. Here is a much simpler one

$$\neg p_0 \vee p_1.$$

Finding a disjunctive normal form for a propositional formula is an important tool for a better understanding of the formula. For example, seeing $p_0 \rightarrow p_1$ in the form $\neg p_0 \vee p_1$ is often most useful, so this form is worth keeping in mind.

1.5.13 Solved problems

Problem 52 *Find the truth function defined by the propositional formula* $(p_0 \wedge \neg p_1) \rightarrow p_2$.

Solution: This is essentially the problem of finding the truth table of $(p_0 \wedge \neg p_1) \rightarrow p_2$. The truth function is a function of three variables, as there are three proposition symbols in the formula. Let us denote the value of p_0 by x_0, the value of p_1 by x_1, and the value of p_2 by x_2. So we need a function $f(x_0, x_1, x_2)$ of the three variables x_0, x_1 and x_2, so that if $x_0 = v(p_0)$, $x_1 = v(p_1)$, and $x_2 = v(p_2)$, then $f(x_0, x_1, x_2) = v((p_0 \wedge \neg p_1) \rightarrow p_2)$. We build the table of f step by step.

To find the value of $f(1, 1, 1)$ we compute what the truth value of $(p_0 \wedge \neg p_1) \rightarrow p_2$ is when p_0, p_1, p_2 are all true. It is clearly 1.

We now make an observation: $(p_0 \wedge \neg p_1) \to p_2$ is true when p_2 is true. So we can immediately fill three more values to the table of f.

The next observation is that $(p_0 \wedge \neg p_1) \to p_2$ is true also if p_0 is false, because then $p_0 \wedge \neg p_1$ is false and hence the implication true.

There are just two values left to compute. In the first case we need a value for $f(1,1,0)$. The value is 1, And finally $f(1,0,0)$ is 0. The table is ready.

x_0	x_1	x_2	$f(x_0,x_1,x_2)$
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1

□

Problem 53 *In each case below, find a propositional formula using $\neg$, $\wedge$ and $\vee$ which defines the given truth function.*

The first truth function:

x_0	x_1	$f(x_0,x_1)$
1	*1*	*1*
1	*0*	*1*
0	*1*	*1*
0	*0*	*1*

It is a column of ones. This truth function is a constant function, constant 1. So if we take the formula $p_0 \vee \neg p_0$, it is a perfect answer. This formula is a tautology, so its truth value is always 1. It does not matter that p_1 does not occur in the formula. But there is no problem adding it to the formula either, and we get the formula $(p_0 \vee \neg p_0) \vee p_1$, but this is superfluous.

The second truth function:

x_0	x_1	$f(x_0,x_1)$
1	*1*	*0*
1	*0*	*1*
0	*1*	*1*
0	*0*	*0*

This is a non-constant function and we apply the method of disjunctive normal forms to it. There are two one-rows, the second and the third. So we build a disjunction of two formulas. The first disjunct is $p_0 \wedge \neg p_1$, and the second disjunct is $\neg p_0 \wedge p_1$. So the formula is $(p_0 \wedge \neg p_1) \vee (\neg p_0 \wedge p_1)$. Now one can easily check that, indeed, this formula defines the given truth function f. This can be seen by simply going through the possible arguments x_0, x_1 and checking in each of the four cases that the two functions, the given function f, and the truth function defined by our formula, give the same answer.

The third truth function:

x_0	x_1	$f(x_0,x_1)$
1	*1*	*0*
1	*0*	*0*
0	*1*	*0*
0	*0*	*0*

This is again a constant function. There are no one-rows, so we could not use the disjunctive normal form method even if we wanted. Instead we use common sense. Any contradiction is false under any valuation. So we take, for example, $p_0 \wedge \neg p_0$. This formula defines a truth function which is constant 0.

The final truth function:

x_0	x_1	$f(x_0,x_1)$
1	*1*	*1*
1	*0*	*0*
0	*1*	*0*
0	*0*	*1*

Again we have two one-rows. We have to build a formula using only $\neg$, $\wedge$ and $\vee$, so we cannot take the formula $p_0 \leftrightarrow p_1$, although this formula clearly defines the given truth function. So we use the method of disjunctive normal forms. There are two one-rows, the first and the fourth. So we build a disjunction of two formulas. The first disjunct is $p_0 \wedge p_1$, and the second disjunct is $\neg p_0 \wedge \neg p_1$. So the formula is $(p_0 \wedge p_1) \vee (\neg p_0 \wedge \neg p_1)$. Now it remains to check that, indeed, this formula defines the given truth function. As above, this can be seen by simply going through the possible arguments x_0, x_1 and checking in each of the four cases that the two functions, the given function f, and the truth function defined by our formula, give the same answer.

x_0	x_1	x_2	$f(x_0, x_1, x_2)$
1	1	1	0
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	1
0	0	0	1

Figure 1.6: A ternary truth function

Problem 54 *Find a propositional formula using* $\neg, \wedge$ *and* $\vee$ *which defines the truth function of Figure 1.6:*

Solution:

This problem clearly calls for an application of the disjunctive normal form method. Let us see how it works. There are 6 one-rows and 2 zero-rows. It is too bad that the disjunctive normal form method focuses on the one-rows, and there are so many of them. Let us use a trick. By focusing on the zero-rows we can build a formula whose **negation** is the requested formula. So we start our defining formula with a negation and then write a formula which has truth value 1 on exactly the zero-rows. There are two zero rows, so we build—inside the negation—a disjunction of two formulas. The first is $p_0 \wedge p_1 \wedge p_2$ and the second is $\neg p_0 \wedge p_1 \wedge p_2$. The final formula is

$$\neg((p_0 \wedge p_1 \wedge p_2) \vee (\neg p_0 \wedge p_1 \wedge p_2)).$$

This formula works, but at the same time it looks suspiciously independent of the truth value of p_0, so let us try a simpler formula

$$\neg(p_1 \wedge p_2).$$

We need to check that the truth function defined by $\neg(p_1 \wedge p_2)$ is the function f. This is easy to do. We note that $f(x_0, x_1, x_2)$ is independent of x_0, just as the truth function defined by our formula is. For the 4 possible values of x_1, x_2 we note f and the truth function defined by our formula give the same value, so we are done.

□

Problem 55 *Show that the ternary truth function of Figure 1.6 is universal.*

Solution: Note that $f(x, x, x) = 1 - x$ and $f(x, x, y) = 1 - xy$. Thus

$$f(f(x, x, y), f(x, x, y), f(x, x, y)) = xy.$$

This shows that we can express negation and conjunction in terms of f. But we already know that $\{\neg, \wedge\}$ is a universal set of connectives. So we are done.

□

Problem 56 *Show that the set* $\{\wedge, \vee\}$ *of connectives is not universal.*

Solution: It is easy to see that any connective $f(x_1, ..., x_n)$, defined in terms of $\wedge$ and $\vee$, satisfies $f(1, ..., 1) = 1$. How? This property holds for $\wedge$ and $\vee$, and the property is preserved by composition of functions. Therefore it holds for every connective defined in terms of $\wedge$ and $\vee$. Thus it follows that negation cannot be defined in terms of $\wedge$ and $\vee$.

□

1.5.14 Problems

Problem 57 *In each case below, find a propositional formula using* $\neg$ *and* $\rightarrow$ *which defines the given truth function.*

x_0	x_1	f
1	1	1
1	0	1
0	1	1
0	0	1

x_0	x_1	f
1	1	1
1	0	0
0	1	0
0	0	1

x_0	x_1	f
1	1	0
1	0	0
0	1	0
0	0	0

x_0	x_1	f
1	1	0
1	0	1
0	1	1
0	0	1

Problem 58 *Show that* $\{\rightarrow\}$ *is not universal. Hint: Show first that any connective* $f(x_1, \ldots, x_n)$*, defined in terms of* $\rightarrow$*, satisfies* $f(1, \ldots, 1) = 1$.

Problem 59 *Show that* $\{\neg\}$ *is not universal. Hint: Think carefully first what kind of truth functions can be defined in terms of negation only. Maybe they are all of a rather simple form.*

Problem 60 *Show that* $\{\wedge, \leftrightarrow\}$ *is not universal.*

Problem 61 *Which of the following formulas are in disjunctive normal form:*

1. p_1
2. $p_0 \vee p_1$
3. $\neg p_0 \wedge p_1$
4. $p_0 \leftrightarrow p_2$
5. $(\neg p_0 \wedge p_1 \wedge p_2) \vee (p_0 \wedge \neg p_1 \wedge p_3) \vee (\neg p_0 \wedge p_2 \wedge p_3)$
6. $(\neg p_0 \vee p_1 \vee p_2) \wedge (p_0 \vee \neg p_1 \vee p_3) \wedge (\neg p_0 \vee p_2 \vee p_3)$
7. $(p_0 \to p_1) \wedge (\neg p_0 \to p_2)$

Problem 62 *Write the following formulas in an equivalent disjunctive normal form:*

1. $p_0 \to p_1$
2. $\neg(p_0 \to p_1)$
3. $(p_0 \vee p_1) \wedge (\neg p_0 \vee p_2)$
4. $(\neg p_0 \vee p_1 \vee p_2) \wedge (p_0 \vee \neg p_1 \vee p_3) \wedge (\neg p_0 \vee p_2 \vee p_3)$

Problem 63 *How many n-ary truth functions are there?*

Problem 64 *Recall the definition of the probability* $p(A)$ *of A in Problem 34. Show*

1. *A is a tautology iff* $p(A) = 1$
2. *A is a contradiction iff* $p(A) = 0$
3. *A is a contingency iff* $0 < p(A) < 1$.

1.6 Natural deduction

1.6.1 What is deduction?

A deduction—also called an inference, derivation or a proof—is a sequence of formulas formed by obeying certain fixed rules, so called *deduction rules*, or *rules of inference*, or *rules of proof*.

The rules reflect "correct thinking".

A deduction is a simplified model of proving theorems in mathematics or in any other science. It tries to capture the rules of correct inference. Deductions are also called (formal) proofs.

The study of deduction goes back to Aristotle, who gave the first set of rules of correct inference, so called *syllogisms*, such as the following:

Every man is mortal.
Socrates is a man.
Therefore, Socrates is mortal.

This does not only sound convincing but it is even convincing if "Socrates", "man" and "mortal" are changed to anything else, for example "Anna", "Finn" and "European":

Every Finn is European.
Anna is a Finn.
Therefore, Anna is European.

It is important to acknowledge right in the beginning of the discussion on deductions (or proofs) that in practice one seldom writes proofs as carefully as we do in this course. Maybe a computer does, but humans talking or writing with paper and pencil certainly don't. In practice proofs are so called *informal proofs*. With an informal proof a scientist becomes convinced—and can convince also fellow scientists—that some conclusion is a valid one. For example, a scientist may perform various experiments and then draw a conclusion. The conclusion is usually not just a statement that such and such results were obtained in the experiments. Rather, the scientist is supposed to make also some conclusions of the type that a hypothesis made in the research project has been confirmed or rejected by the experiments. The conclusions may be based on an ingenious combination of conclusions from known theories and the new experiments. A need for a proof of a different kind may be the need to prove that

- a bridge does not collapse,
- a dam does not burst,
- a nuclear reactor does not go to meltdown,
- an operating system does not crash,
- a rocket does not explode on take off,

- and so on.

In such cases the proof—if such even exists—may be so complicated that is has to be written in a manner that a computer can understand and check for correctness. This manner may very well be the kind of pedantic formal proof that we now learn. There are computer based proof assistants[4] that help humans write very exact formal proofs.

The point of introducing an exact concept of deduction is that on the one hand it throws light on the important process of making correct conclusions from given hypothesis, and on the other hand it subjects the concept of deduction to mathematical investigation.

1.6.2 Complexity of deduction

It is easy to check if a given deduction is correct, that is, whether it is written according to the deduction rules. Even a computer can check correctness of deductions. The difficulty is in finding the deduction. A big part of the creative work of a mathematician, or of a researcher of some other science, consists of trying to prove something that one believes must be true.

1.6.3 Natural deduction

Natural deduction is a particular system of writing deductions. As its name indicates, it attempts to imitate as far as possible human thinking.

In natural deduction we have some assumptions $B_1, ..., B_n$, and we want to derive a conclusion A from them. If we can, we write $\{B_1, \ldots, B_n\} \vdash A$. If no such deduction exists, we write $\{B_1, \ldots, B_n\} \nvdash A$.

Deductions are built from simpler deductions and assumptions by means of introduction and elimination rules.

1.6.4 A simple natural deduction

Here is a simple natural deduction:

$$\dfrac{\dfrac{A \quad B}{A \wedge B}\,\wedge\mathbf{I} \qquad \dfrac{A \quad B}{A \wedge B}\,\wedge\mathbf{I}}{(A \wedge B) \wedge (A \wedge B)}\,\wedge\mathbf{I}$$

[4]see e.g. http://coq.inria.fr/

Let us look at its components. On the top there are *assumptions* A and B. Each horizontal line represents an application of an inference rule. The name of the rule, in this case $\wedge I$, is next to the inference line. The conclusion $(A \wedge B) \wedge (A \wedge B)$ is at the bottom.

We can see that this inference (or deduction) has

$$\dfrac{A \quad B}{A \wedge B}\,\wedge\mathbf{I}$$

as a sub inference, even twice. This is the general pattern of deductions (i.e. inferences): they consist of pieces we put together and each piece is a smaller inference. The smallest inferences are like the one above in that they consist of just on inference line.

1.6.5 Rules for conjunction

The essence of a deduction is that it is built by combining certain rules. The rules governing conjunction, trying to capture how we understand the word "and", are as follows:

$\wedge$-Introduction Rule:

$$\dfrac{A \quad B}{A \wedge B}\,\wedge\mathbf{I}$$

$\wedge$-Elimination Rules:

$$\dfrac{A \wedge B}{A}\,\wedge\mathbf{E} \qquad \dfrac{A \wedge B}{B}\,\wedge\mathbf{E}$$

The idea of the the $\wedge$-Introduction Rule is very simple: If we know A and we know B, we know their conjunction $A \wedge B$. Similarly for the $\wedge$-Elimination Rule if we know $A \wedge B$, we know both A and we know B.

1.6.6 Deriving $(A \wedge B) \wedge C$ from $A \wedge (B \wedge C)$

Let us look at a deduction involving the rules governing conjunction. We derive $(A \wedge B) \wedge C$ from $A \wedge (B \wedge C)$.

Assuming $A \wedge (B \wedge C)$ we know in principle that A, B and C hold and can be derived from $A \wedge (B \wedge C)$ by the rule $\wedge E$.

To derive $(A \wedge B) \wedge C$ we first derive $A \wedge B$ and then combine this with C, which again can be derived from $A \wedge (B \wedge C)$ by the rule $\wedge E$, to get $(A \wedge B) \wedge C$.

$$\dfrac{\dfrac{\dfrac{A\wedge(B\wedge C)}{A}\,{\scriptstyle\wedge\mathbf{E}} \qquad \dfrac{\dfrac{A\wedge(B\wedge C)}{B\wedge C}\,{\scriptstyle\wedge\mathbf{E}}}{B}\,{\scriptstyle\wedge\mathbf{E}}}{A\wedge B}\,{\scriptstyle\wedge\mathbf{I}} \qquad \dfrac{\dfrac{A\wedge(B\wedge C)}{B\wedge C}\,{\scriptstyle\wedge\mathbf{E}}}{C}\,{\scriptstyle\wedge\mathbf{E}}}{(A\wedge B)\wedge C}\,{\scriptstyle\wedge\mathbf{I}}$$

Figure 1.7: Deriving $(A \wedge B) \wedge C$ from $A \wedge (B \wedge C)$

Now the deduction. First we derive A, then B, and so we have $A \wedge B$. Next we derive C, and finally put together what we have got.
For the entire deduction, see Figure 1.7.

1.6.7 An example of a proof by cases

Proof by cases is common in mathematics. Here is an example.

Suppose we want to show that $n^2 + 7n$ is even for all n. A possible proof proceeds as follows:

- We know that every number is even (of the form $2m$) or odd (of the form $2m + 1$).
- Let us take an arbitrary natural number n.
- Case 1: n is even i.e. $n = 2m$: $n^2 + 7n = 4m^2 + 14m = 2(2m^2 + 7m)$ even.
- Case 2: n is odd i.e. $n = 2m + 1$: $n^2 + 7n = (2m+1)^2+7(2m+1) = 4m^2+4m+1+14m+7 = 4m^2 + 18m + 8 = 2(2m^2 + 9m + 4)$ even.
- So in either case $n^2 + 7n$ is even.
- So $n^2 + 7n$ really is always even.

1.6.8 The structure of the proof

The structure of the above proof by cases is the following. We want to prove that $n^2 + 7n$ is even. We know that n is even or odd. Note that this is a disjunction. We also know that if n is even, then $n^2 + 7n$ is even, and also that if n is odd, then $n^2 + 7n$ is even. And that is all we have to do, we are convinced that $n^2 + 7n$ is even whether n is even or odd. Schematically we can think of this proof in this form:

$$\dfrac{\begin{array}{c} \\ \\ n \text{ is even or odd}\end{array} \quad \begin{array}{c} n \text{ is even} \\ \vdots \\ n^2+7n \text{ is even}\end{array} \quad \begin{array}{c} n \text{ is odd} \\ \vdots \\ n^2+7n \text{ is even}\end{array}}{n^2+7n \text{ is even}}$$

1.6.9 Proof by cases

If we analyze further what is going on in the above proof by cases we can think of it as an application of a rule of the form

$$\dfrac{\begin{array}{c} \\ \\ A\vee B\end{array} \quad \begin{array}{c} [A] \\ \vdots \\ C\end{array} \quad \begin{array}{c} [B] \\ \vdots \\ C\end{array}}{C}$$

We know $A \vee B$, and we can prove C if we assume A, but also if we assume B. So since $A \vee B$ is established, we are convinced of C.

1.6.10 Making assumptions

In proofs by cases we make temporary assumptions ("n even", "n odd"). In a natural deduction we can make temporary assumptions, postulate a formula which is not an assumption, as long as we are able to eliminate the assumption at some point. When a temporary assumption A is eliminated it is put into square brackets $[A]$. Eliminating a temporary assumption is not always necessary. It is a right, not a duty. However, a temporary assumption which is not eliminated remains an assumption of the deduction and usually ruins the goal of the deduction.

It is important to learn to distinguish assumptions and temporary assumptions from each other. Temporary assumptions arise typically when we know a disjunction and we want to make some conclusion. Then we temporarily assume first one disjunct and then the second, to see what happens. In the above proof that $n^2 + 7n$ is even we knew that n itself is even or odd but we did not know which. So we assumed first that n is even and then that n is odd.

1.6.11 Rules for Disjunction

We are ready now to lay down the rules that govern disjunction in natural deduction.

$\vee$-Introduction Rules:

$$\frac{A}{A \vee B}\ \vee\mathbf{I} \qquad \frac{B}{A \vee B}\ \vee\mathbf{I}$$

$\vee$-Elimination Rules:

$$\frac{A \vee B \quad \begin{matrix}[A] \\ \vdots \\ C\end{matrix} \quad \begin{matrix}[B] \\ \vdots \\ C\end{matrix}}{C}\ \vee E$$

The $\vee$-Introduction Rule is rather obvious: If we know a disjunct, we know the disjunction. It is $\vee$-Elimination Rule that is more interesting and features the just discussed proof by cases. The idea is that if we know $A \vee B$ and we know C whether we make the temporary assumption A or B, then we can conclude C.

1.6.12 Eliminating disjunction

We derive C from the assumption $(A \wedge C) \vee (B \wedge C)$. Let us first think why C follows from $(A \wedge C) \vee (B \wedge C)$: $(A \wedge C) \vee (B \wedge C)$ says that $A \wedge C$ or $B \wedge C$ (or both) is the case. But whether $A \wedge C$ or $B \wedge C$ is the case, we have anyway C.

1.6.13 Eliminating disjunction

Here is the natural deduction for a derivation of C from the assumption $(A \wedge C) \vee (B \wedge C)$.

$$\frac{(A \wedge C) \vee (B \wedge C) \quad \dfrac{[A \wedge C]}{C}\ \wedge\mathbf{E} \quad \dfrac{[B \wedge C]}{C}\ \wedge\mathbf{E}}{C}\ \vee\mathbf{E}$$

It is one big application of $\vee$-elimination. We just have to use $\wedge$-elimination to take care of the temporary assumptions.

1.6.14 Introducing disjunction

We derive $(A \vee C) \wedge (B \vee D)$ from the assumption $A \wedge B$. Let us first think intuitively why $(A \vee C) \wedge (B \vee D)$ seems to follow from $A \wedge B$. If $A \wedge B$ is the case, then both A and B are the case. From A follows $A \vee C$. From B follows $B \vee D$. So we have $(A \vee C) \wedge (B \vee D)$.

1.6.15 Introducing disjunction

Now we build the required natural deduction for deriving $(A \vee C) \wedge (B \vee D)$ from the assumption $A \wedge B$.

We attempt to use the $\wedge$-Introduction Rule. In order to get $A \vee C$ we need to apply first the $\wedge$-Elimination Rule to get A and then the $\vee$-Introduction Rule to get $A \vee C$. The same with $B \vee D$.

1.6.16 Derivation of A from $A \vee A$

Here we derive A from $A \vee A$. This is a somewhat singular case, but note that we are now dealing with deductions that also computers may use. It is obvious to us that A follows from $A \vee A$, but it may not be obvious to a computer. Also, there are logical systems, for example so called dependence logic [5], where one cannot derive A from $A \vee A$. In dependence logic the disjunction $A \vee A$ is satisfied by a "team" X if and only if X can be represented as a union $Y \cup Z$ such that both Y and Z satisfy A, and a team X may very well satisfy $A \vee A$ without satisfying A, roughly for a similar reason why five euros may be the exact fare for two tram tickets but not for one tram ticket.

Here is the deduction of A from $A \vee A$:

$$\frac{A \vee A \quad [A] \quad [A]}{A}$$

Note that A is both the assumption and conclusion of the one sentence derivation

$$A.$$

This is a singular deduction but a deduction all the same.

[5] en.wikipedia.org/wiki/Dependence_logic

1.6.17 Solved problems

Problem 65 *$B \wedge A$ is derivable from $A \wedge B$.*

We derive B from $A \wedge B$, and also A. Then we combine the two derivations and derive $B \wedge A$.

$$\cfrac{\cfrac{A \wedge B}{B}\,{\wedge}\mathbf{E} \qquad \cfrac{A \wedge B}{A}\,{\wedge}\mathbf{E}}{B \wedge A}\,{\wedge}\mathbf{I}$$

Problem 66 *$A \vee B$ is derivable from $A \wedge B$*

This is an easy one. We derive A from $A \wedge B$, and then immediately $A \vee B$ from A.

$$\cfrac{\cfrac{A \wedge B}{A}\,{\wedge}\mathbf{E}}{A \vee B}\,{\vee}\mathbf{I}$$

Note that there is another equally good solution which goes via B:

$$\cfrac{\cfrac{A \wedge B}{B}\,{\wedge}\mathbf{E}}{A \vee B}\,{\vee}\mathbf{I}$$

A deduction is not something that is unique.

Problem 67 *The train is moving or else both the door is open and the green light is on. Derive that the train is moving or the green light is on?*

We have to derive $p_0 \vee p_2$ from $p_0 \vee (p_1 \wedge p_2)$. One might first try to use the $\vee$-Introduction Rule, but that does not seem to lead anywhere. The next idea is to use the $\vee$-Elimination Rule to the assumption $p_0 \vee (p_1 \wedge p_2)$, and this actually works.

By assumption either p_0 or $p_1 \wedge p_2$ holds. In both cases $p_0 \vee p_2$ follows. In order to use the $\vee$-Elimination Rule we set up the stage:

$$\cfrac{p_0 \vee (p_1 \wedge p_2) \qquad \begin{matrix} \\ p_0 \\ \vdots \end{matrix} \qquad \begin{matrix} p_1 \wedge p_2 \\ \\ \vdots \end{matrix}}{p_0 \vee p_2}\,{\vee}\mathbf{E}$$

Now we have to infer $p_0 \vee p_2$ first from p_0 and then again from $p_1 \wedge p_2$. But both tasks are easy:

$$\cfrac{p_0 \vee (p_1 \wedge p_2) \qquad \cfrac{[p_0]}{p_0 \vee p_2}\,{\vee}\mathbf{I} \qquad \cfrac{\cfrac{[p_1 \wedge p_2]}{p_2}\,{\wedge}\mathbf{E}}{p_0 \vee p_2}\,{\vee}\mathbf{I}}{p_0 \vee p_2}\,{\vee}\mathbf{E}$$

Problem 68 *Derive $(A \vee B) \vee C$ from $A \vee (B \vee C)$?*

Our assumption is the disjunction of A and $B \vee C$, so we use the $\vee$-Elimination Rule. We note that from A follows $A \vee B$ and hence $(A \vee B) \vee C$. On the other hand, from $B \vee C$ also follows $(A \vee B) \vee C$ by an application of the $\vee$-Elimination Rule, that is, a proof by cases:

Case 1: Assume B. Then $A \vee B$, hence $(A \vee B) \vee C$.

Case 2: Assume C. Then $(A \vee B) \vee C$.

The deduction is ready. (See Figure 1.8.)

1.6.18 Problems

Problem 69 *Use natural deduction to derive: $A \wedge (B \vee C)$ from $A \wedge C$.*

Problem 70 *Use natural deduction to derive: $A \wedge (B \vee C)$ from $(A \wedge B) \vee (A \wedge C)$. Hint: A temporary assumption can very well be used several times before it is eliminated.*

Problem 71 *Use natural deduction to derive: $A \vee (B \wedge C)$ from $(A \vee B) \wedge (A \vee C)$.*

Problem 72 *Use natural deduction to derive: $(A \wedge B) \vee (A \wedge C)$ from $A \wedge (B \vee C)$.*

Problem 73 *Use natural deduction to derive: $(A \vee B) \wedge (A \vee C)$ from $A \vee (B \wedge C)$.*

1.7 Natural deduction: Implication

1.7.1 Rules for Implication

We shall now learn an *elimination* rule as well as an *introduction* rule for implication.

These rules are the following:

The $\rightarrow$-Elimination Rule is:

$$\cfrac{A \rightarrow B \quad A}{B}\,{\rightarrow}\mathbf{E}$$

The $\rightarrow$-Introduction Rule is:

$$\cfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \rightarrow B}\,{\rightarrow}\mathbf{I}$$

$$\dfrac{A \vee (B \vee C) \quad \dfrac{\dfrac{[A]}{A \vee B}\,\vee\mathbf{I}}{(A \vee B) \vee C}\,\vee\mathbf{I} \quad \dfrac{[B \vee C] \quad \dfrac{\dfrac{[B]}{A \vee B}\,\vee\mathbf{I}}{(A \vee B) \vee C}\,\vee\mathbf{I} \quad \dfrac{[C]}{(A \vee B) \vee C}\,\vee\mathbf{I}}{(A \vee B) \vee C}\,\vee\mathbf{E}}{(A \vee B) \vee C}\,\vee\mathbf{I}$$

Figure 1.8: Deriving $(A \vee B) \vee C$ from $A \vee (B \vee C)$

1.7.2 Rules for Implication

Learning to use these rules requires a little practice. However, let us first look at these rules a bit more carefully. The content of the $\rightarrow$-Elimination Rule is that if we have derived both A and $A \rightarrow B$, then we can conclude B. This is how we think of implication $A \rightarrow B$. We even read $A \rightarrow B$ as "If A, then B". So if we have A, we should indeed conclude B.

What about the $\rightarrow$-Introduction Rule? This rules says that if we can derive B under the assumption A, then we may conclude $A \rightarrow B$ without any assumptions. This rule establishes a tight relationship between derivability and $\rightarrow$, and corresponds to the intuition that implication is a formalization of logical consequence.

1.7.3 Eliminating Implication

Here is an example of the use of the $\rightarrow$-Elimination Rule: We derive C from A and $(A \vee B) \rightarrow C$.

We prepare ourselves to using the $\rightarrow$-Elimination Rule. For this to work we have to derive $A \vee B$ from A. But this can be done by means of the $\vee$-Introduction Rule:

$$\dfrac{(A \vee B) \rightarrow C \quad \dfrac{A}{A \vee B}\,\vee\mathbf{I}}{C}\,\rightarrow\mathbf{E}$$

1.7.4 Introducing implication

Here is an example of the use of the $\rightarrow$-Introduction Rule: We derive $A \rightarrow C$ from the assumptions $A \rightarrow B$ and $B \rightarrow C$. First we prepare ourselves for an application of the $\rightarrow$-Introduction Rule. So we make the temporary assumption A and try to derive C. Our assumption $A \rightarrow B$, together with the temporary assumption A immediately gives B. From this and the other assumption $B \rightarrow C$ we get C.

$$\dfrac{\dfrac{B \rightarrow C \quad \dfrac{A \rightarrow B \quad [A]}{B}\,\rightarrow\mathbf{E}}{C}\,\rightarrow\mathbf{E}}{A \rightarrow C}\,\rightarrow\mathbf{I}$$

1.7.5 Example: Derivation of $A \rightarrow A$

This is again a somewhat singular case! Let us again remark that proving such an extreme case as $A \rightarrow A$ may seem totally unnecessary, but it is good to check because these extremely short deductions are building blocks of bigger ones. If this is a simple case, it should be easy for us.

The formula A is both the *assumption* and *conclusion* of this one-sentence derivation:

$$A$$

Now the assumption A can be eliminated to derive $A \rightarrow A$:

$$\dfrac{[A]}{A \rightarrow A}$$

1.7.6 Rules for equivalence

Dealing with equivalence is like dealing with two implications at the same time. The rules governing equivalence are:

The $\leftrightarrow$-Elimination Rule s are:

$$\dfrac{A \leftrightarrow B \quad A}{B}\,\leftrightarrow\mathbf{E} \qquad \dfrac{A \leftrightarrow B \quad B}{A}\,\leftrightarrow\mathbf{E}$$

The $\leftrightarrow$-Introduction Rule is:

$$\dfrac{\begin{matrix}[B] \\ \vdots \\ A\end{matrix} \quad \begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \leftrightarrow B} \leftrightarrow \mathbf{I}$$

1.7.7 Rules for equivalence

What is behind the rules of the equivalence-connective? The $\leftrightarrow$-Elimination Rule says that if we can derive both A and $A \leftrightarrow B$, then we can conclude B. The intuition behind should be obvious. After all, $A \leftrightarrow B$ says that A and B are "equivalent". So if we have derived A, we should accept B, too, and vice versa. As to the $\leftrightarrow$-Introduction Rule, it says that if we can derive B under the assumption A, and also A under the assumption B, then we may conclude $A \leftrightarrow B$ without any assumptions. The derivability of A and B from each other is interpreted as the equivalence of A and B, i.e. as $A \leftrightarrow B$.

1.7.8 Eliminating equivalence

As an example of the $\leftrightarrow$-Elimination Rule we derive C from A and $(A \vee B) \leftrightarrow C$. First we prepare the ground for an application of the $\leftrightarrow$-Elimination Rule. For this to work, we have to somehow deduce $A \vee B$. So what do we have, what can we use to deduce $A \vee B$. One of our assumptions is A. But of course we can deduce $A \vee B$ from A, this is just $\vee$-Introduction Rule. So we are ready:

$$\dfrac{(A \vee B) \leftrightarrow C \qquad \dfrac{A}{A \vee B} \vee \mathbf{I}}{C} \leftrightarrow \mathbf{E}$$

1.7.9 Introducing equivalence

As an example of the $\leftrightarrow$-Introduction Rule we derive $A \leftrightarrow C$ from the assumptions $A \leftrightarrow B$ and $B \leftrightarrow C$. First we set the stage for an application of $\leftrightarrow$-Introduction Rule. We then make the temporary assumption A trying to derive C and at the same time we make the temporary assumption C trying to derive A. We now use the assumed equivalences $A \leftrightarrow B$ and $B \leftrightarrow C$, applying successively the $\leftrightarrow$-Elimination Rule, getting C on the left and A on the right. We have derived C from A and A from C:

$$\dfrac{\dfrac{B \leftrightarrow C \qquad \dfrac{A \leftrightarrow B \quad [A]}{B} \leftrightarrow \mathrm{E}}{C} \leftrightarrow \mathrm{E} \qquad \dfrac{A \leftrightarrow B \qquad \dfrac{B \leftrightarrow C \quad [C]}{B} \leftrightarrow \mathrm{E}}{A} \leftrightarrow \mathrm{E}}{A \leftrightarrow C} \leftrightarrow \mathrm{I}$$

1.7.10 Solved Problems

Problem 74 *Derive* $(A \wedge B) \rightarrow C$ *from* $A \rightarrow (B \rightarrow C)$.

What is the idea? We assume $A \wedge B$. *Thus we have* A *and* B. *From* A *and* $A \rightarrow (B \rightarrow C)$ *we get* $B \rightarrow C$. *From* B *and* $B \rightarrow C$ *we get* C. *We are done. We can draw the deduction.*

We set the stage for an application of $\rightarrow$*-Introduction Rule. We want to derive an implication and we have a conjunction as a temporary assumption. We use the* $\wedge$*-Elimination Rule twice to get* A *and* B. *Then we use the assumption to get first* $B \rightarrow C$ *and then finally* C. *Now the* $\rightarrow$*-Introduction Rule finishes the job.*

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \wedge B]}{A} \wedge \mathbf{E} \qquad A \rightarrow (B \rightarrow C)}{B \rightarrow C} \rightarrow \mathbf{E} \qquad \dfrac{[A \wedge B]}{B} \wedge \mathbf{E}}{C} \rightarrow \mathbf{E}}{(A \wedge B) \rightarrow C} \rightarrow \mathbf{I}$$

Problem 75 *Derive* $A \rightarrow (B \rightarrow A)$

We prepare ourselves to applying $\rightarrow$*-Introduction Rule. For this end we make the temporary assumption* A.

Now we use the $\rightarrow$*-Introduction Rule to conclude* $B \rightarrow A$ *and at the same time eliminate the temporary assumption* B. *Oops! We do not have* B *as a temporary assumption, so how can we eliminate it? Nothing to worry about. No need to eliminate* B *because the temporary assumption* B *was never made. We have deduced* $B \rightarrow A$ *from the temporary assumption* A*:*

$$\dfrac{A}{B \rightarrow A}$$

We get $A \rightarrow (B \rightarrow A)$ *with another application of the* $\rightarrow$*-Introduction Rule, and at the same time we can eliminate the temporary assumption* A. *Now the deduction is ready:*

$$\dfrac{\dfrac{[A]}{B \rightarrow A}}{A \rightarrow (B \rightarrow A)}$$

Problem 76 *Derive* $(A \to B) \to (A \to C)$ *from* $A \to (B \to C)$.

We assume $A \to B$. *To derive* $A \to C$ *we next assume* A *and try to derive* C. *From* A *and* $A \to (B \to C)$ *we get* $B \to C$. *From* $A \to B$ *we also get* B. *So from* $B \to C$ *we get finally* C. *Now we can use the* $\to$*-Introduction Rule to get* $A \to C$ *and at the same time we can eliminate the temporary assumption* A*, in two places. Another application of* $\to$*-Introduction Rule give the desired* $(A \to B) \to (A \to C)$*, and the temporary assumption* $A \to B$ *can be eliminated. For the complete deduction see Figure 1.9*

1.7.11 Problems

Problem 77 *Use natural deduction to derive:* $B \to (A \to C)$ *from* $A \to (B \to C)$.

Problem 78 *Use natural deduction to derive:* $(A \wedge B) \to (C \wedge D)$ *from* $(B \wedge A) \to (D \wedge C)$.

Problem 79 *Use natural deduction to derive:* $A \to (C \vee B)$ *from* $A \to (B \vee C)$.

Problem 80 *Use natural deduction to derive:* $A \to (B \vee C)$ *from* $(A \to B) \vee (A \to C)$.

Problem 81 *Use natural deduction to derive:* $A \to B$ *from* $(A \wedge C) \leftrightarrow B$ *and* C.

Problem 82 *Use natural deduction to derive:* $(A \to B) \to ((B \to A) \to (A \leftrightarrow B))$.

1.8 Natural deduction: Negation

The deduction methods we have so far learnt are so-called direct deductions, that is, deductions in which we start from given assumptions and derive step by step to the desired conclusion. We have also learnt the method of proof by cases, that is, the use of the disjunction elimination rule. Now we learn to prove negated statements.

1.8.1 Proving a contradiction

The basic idea behind proving $\neg A$ is to derive contradiction from A. In other words, we take A as a temporary assumption and derive a contradiction $B \wedge \neg B$ for some formula B. The formula B can in principle be A but it is more likely to be something else.

When we have derived a contradiction $B \wedge \neg B$ from the temporary assumption A, we consider $\neg A$ proved. At the same time the temporary assumption A can be eliminated.

Let us try to understand the idea of proving a negated sentence. In all simplicity it is the following idea: Suppose I come into a house from outside and make the statement that it is *not* raining. Someone may doubt and ask for a "proof". I can say: Look, if it rained my coat would be wet, but my coat is completely dry, so it cannot be raining outside. So we accept the negation of "it is raining" because "it is raining" would lead to a contradiction with what we can clearly see with our own eyes.

For another example, let us see why it is easy to accept the negation of "The earth is flat". If the earth was flat the shadow of the earth on the moon in a lunar eclipse would not be round, at least not always, but anyone who has observed a lunar eclipse has seen that the shadow of the earth on the moon *is* round. Also, people on long distance flights as well as captains of long distance ocean ships would observe "coming to the edge of the world"-type phenomena but what they actually observe is that the horizon is always equally far away and very distant objects seem to be partially behind the horizon as if the earth was everywhere equally round. So the assumption that the earth is flat leads to conclusions that defy known observed facts. So we accept the negation of "The earth is flat" as true.

The point of these two examples—and there are of course many more—is that in everyday language we accept negation of a statement if the statement leads to an absurdity. This is why also in natural deduction we prove $\neg A$ by assuming A and deriving a contradiction.

1.8.2 Practice

Let us practice deriving a contradiction from given assumptions. We derive a contradiction from $(A \to B) \wedge A \wedge \neg B$. Intuitively, we have a contradiction, because from $A \to B$ and A be get B, and on the other hand, we assume $\neg B$. So in the natural deduction we first derive $A \to B$ from the assumption by means of $\wedge$-Elimination Rule (See Figure 1.10). Then we also derive A from the assumption by means of another application of the

$$\dfrac{\dfrac{\dfrac{\dfrac{[A] \quad [A \to B]}{B}\,{\to}\mathbf{E} \qquad \dfrac{[A] \quad A \to (B \to C)}{B \to C}\,{\to}\mathbf{E}}{C}\,{\to}\mathbf{E}}{A \to C}\,{\to}\mathbf{I}}{(A \to B) \to (A \to C)}\,{\to}\mathbf{I}$$

Figure 1.9: Deriving $(A \to B) \to (A \to C)$ from $A \to (B \to C)$

$\wedge$-Elimination Rule. Then we apply the $\to$-Elimination Rule to $A \to B$ and A, obtaining B. Finally we get $\neg B$ from the assumption again by means of the $\wedge$-Elimination Rule. So an application of the $\wedge$-Introduction Rule finishes the deduction of the contradiction $B \wedge \neg B$ from the given assumption (See Figure 1.10).

1.8.3 Introducing negation

The rule for introducing a negation is:

The $\neg$-Introduction Rule is:

$$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B \wedge \neg B\end{array}}{\neg A}\,\neg\mathbf{I}$$

We learnt above how to derive a contradiction from $(A \to B) \wedge A \wedge \neg B$. With the $\neg$-Introduction Rule we can now write down a deduction for $\neg((A \to B) \wedge A \wedge \neg B)$ without any assumptions. We simply take $(A \to B) \wedge A \wedge \neg B$ as a temporary assumption, repeat the above derivation of $B \wedge \neg B$ from the given assumption, then use $\neg$-Introduction Rule to derive the negation of the assumption, and at the same time the temporary assumption is eliminated, in three places. The full deduction is in Figure 1.11.

1.8.4 Deriving $\neg\neg A$ from A

Let us then look at the derivation of $\neg\neg A$ from A. This is perhaps a little difficult derivation because the derivation is very short and it may be difficult to see what is going on. The rough idea is the following: We assume A and want to show that $\neg A$ leads to a contradiction. Well, of course it leads to a contradiction because it blatantly contradicts the assumption A. So let us write down this natural deduction. We make the temporary assumption $\neg A$, and get immediately $A \wedge \neg A$ by means of the $\wedge$-Introduction Rule. So now we have derived a contradiction and we can use $\neg$-Introduction Rule to derive $\neg\neg A$, eliminating at the same time the temporary assumption $\neg A$. We are done.

$$\dfrac{\dfrac{A \quad [\neg A]}{A \wedge \neg A}\,\wedge\mathbf{I}}{\neg\neg A}\,\neg\mathbf{I}$$

1.8.5 Indirect deductions

The toughest deductions are the *indirect* deductions that we shall now introduce. These deductions are called indirect because during the deduction we make a temporary assumption which seems to come out of the blue. In a typical indirect proof, often called *reductio ad absurdum*, we want to prove A and start by making the temporary assumption $\neg A$. If we can now derive a contradiction we know, intuitively, that A must be true, as $\neg A$ leads to a contradiction. Strictly speaking, what happens, is that we get $\neg\neg A$ by the $\neg$-Introduction Rule and then we let the two negations cancel each other out. For this to be permissible in a deduction we need a new rule, a $\neg$-Elimination Rule, which we shall now introduce:

The $\neg$-Elimination Rule is:

$$\dfrac{\neg\neg A}{A}\,\neg\mathbf{E}$$

Let us repeat the reasoning behind the $\neg$-Elimination Rule. The reasoning is extremely simple: Assuming $\neg\neg A$ means assuming that it is not the case that $\neg A$, so it must be the case that A, because, well, either A or $\neg A$ has to be true, so if it is not $\neg A$ it must be A. This reasoning is often

$$\dfrac{\dfrac{\dfrac{(A \to B) \wedge A \wedge \neg B}{A \to B}\ {\scriptstyle \wedge\mathbf{E}} \quad \dfrac{(A \to B) \wedge A \wedge \neg B}{A}\ {\scriptstyle \wedge\mathbf{E}}}{B}\ {\scriptstyle \to\mathbf{E}} \quad \dfrac{(A \to B) \wedge A \wedge \neg B}{\neg B}\ {\scriptstyle \wedge\mathbf{E}}}{B \wedge \neg B}\ {\scriptstyle \wedge\mathbf{I}}$$

Figure 1.10: Deriving a contradiction from $(A \to B) \wedge A \wedge \neg B$

$$\dfrac{\dfrac{\dfrac{\dfrac{[(A \to B) \wedge A \wedge \neg B]}{A \to B}\ {\scriptstyle \wedge\mathbf{E}} \quad \dfrac{[(A \to B) \wedge A \wedge \neg B]}{A}\ {\scriptstyle \wedge\mathbf{E}}}{B}\ {\scriptstyle \to\mathbf{E}} \quad \dfrac{[(A \to B) \wedge A \wedge \neg B]}{\neg B}\ {\scriptstyle \wedge\mathbf{E}}}{B \wedge \neg B}\ {\scriptstyle \wedge\mathbf{I}}}{\neg((A \to B) \wedge A \wedge \neg B)}\ {\scriptstyle \neg\mathbf{I}}$$

Figure 1.11: Deriving $\neg((A \to B) \wedge A \wedge \neg B)$

called reasoning in *classical logic* because there is also a different kind of logic, *non-classical logic*. In fact, non-classical logics come in different varieties but a common feature to them is that they are not based on the idea that there are just two truth values. A famous example from Aristotle is the sentence "Tomorrow a sea battle will take place". Let us call this sentence A. To say that A is true, seems unfounded, as something may happen which makes the sea battle impossible or unnecessary. Should we then accept $\neg A$ as true? This seems equally unfounded simply because we do not know what tomorrow brings about. So it seems most reasonable to say that neither A nor $\neg A$ is (yet) true. The sentence A is an example of a formula of *temporal logic*.

We have now introduced the two rules that govern deduction about negation in propositional logic:

The $\neg$-Introduction Rule:

$$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B \wedge \neg B\end{array}}{\neg A}\ {\scriptstyle \neg\mathbf{I}}$$

The $\neg$-Elimination Rule:

$$\dfrac{\neg\neg A}{A}\ {\scriptstyle \neg\mathbf{E}}$$

1.8.6 Deriving $A \to B$ from $\neg A$

To prove $A \to B$, we assume A. From A and the assumption $\neg A$ we get $A \wedge \neg A$, i.e. a contradiction. By the negation introduction rule we get $\neg\neg B$. We could eliminate the assumption $\neg B$, if we had made this temporary assumption, but we did not. By negation elimination rule we get B. Now we finish the deduction by appealing to the $\to$-Introduction Rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{[A] \quad \neg A}{A \wedge \neg A}\ {\scriptstyle \wedge\mathbf{I}}}{\neg\neg B}\ {\scriptstyle \neg\mathbf{I}}}{B}\ {\scriptstyle \neg\mathbf{E}}}{A \to B}\ {\scriptstyle \to\mathbf{I}}$$

1.8.7 Deriving A from $\neg(A \to B)$

This is a typical indirect proof. We have to prove A so make the temporary assumption $\neg A$. In a sense, we check how would the opposite of A, namely $\neg A$, fit our situation. So assume $\neg A$. As above, we can derive $A \to B$. From $\neg(A \to B)$ we get $\neg(A \to B) \wedge (A \to B)$. So the assumption $\neg A$ has led to a contradiction. By negation introduction we get $\neg\neg A$, and at the same time we eliminate the temporary assumption $\neg A$. By negation elimination we get A.

$$\cfrac{\cfrac{\neg(A \to B) \qquad \cfrac{\cfrac{\cfrac{\cfrac{[A] \quad [\neg A]}{A \wedge \neg A}\,{\scriptstyle \wedge\mathbf{I}}}{\neg\neg B}\,{\scriptstyle \neg\mathbf{I}}}{B}\,{\scriptstyle \neg\mathbf{E}}}{A \to B}\,{\scriptstyle \to\mathbf{I}}}{\neg(A \to B) \wedge (A \to B)}\,{\scriptstyle \wedge\mathbf{I}}}{\cfrac{\neg\neg A}{A}\,{\scriptstyle \neg\mathbf{E}}}\,{\scriptstyle \neg\mathbf{I}}$$

1.8.8 Deriving $B \wedge \neg B$ from $A \wedge \neg A$

This looks a little odd, but it shows that it does not matter which formula you have in a contradiction.

From the contradiction $A \wedge \neg A$ we get $\neg\neg B$ and then B. Note that we have to go via the double negation of B because we do not have a rule which would allow us to infer B directly from a contradiction. Similarly we infer $\neg B$ from $A \wedge \neg A$. Now we do not have to go via double negation. Finally we put B and $\neg B$ together, and we are done.

$$\cfrac{\cfrac{\cfrac{A \wedge \neg A}{\neg\neg B}\,{\scriptstyle \neg\mathbf{I}}}{B}\,{\scriptstyle \neg\mathbf{E}} \qquad \cfrac{A \wedge \neg A}{\neg B}\,{\scriptstyle \neg\mathbf{I}}}{B \wedge \neg B}\,{\scriptstyle \wedge\mathbf{I}}$$

1.8.9 Solved problems

Problem 83 $\neg A \wedge \neg B$ *can be derived from* $\neg(A \vee B)$. *(This is one of the so-called De Morgan laws..)*

Solution: Let us first think intuitively why $\neg A \wedge \neg B$ should follow from $\neg(A \vee B)$. Say, it is not true that it rains or snows. Why can we conclude that it neither rains nor snows? Well, because if it for example rained, then it would a fortiori rain or snow, so we would contradict the made assumption. We try to make this formal.

Let us continue thinking intuitively why $\neg A \wedge \neg B$ should follow from $\neg(A \vee B)$. Since we are proving a conjunction we can take each conjunct separately. Let us look at $\neg A$. Assuming A gives $A \vee B$, contradicting immediately the assumption $\neg(A \vee B)$. So we must conclude $\neg A$. Similarly we get $\neg B$.

So we start by writing down the assumption $\neg(A \vee B)$ and by making the temporary assumption A in order to derive a contradiction and be able to derive $\neg A$.

We get a contradiction immediately by a using the $\vee$-Introduction Rule and then the $\wedge$-Introduction Rule.

With an application of the $\neg$-Introduction Rule we get $\neg A$, and we can eliminate the temporary assumption A.

We do the same with B, obtaining $\neg B$.

An application of the $\wedge$-Introduction Rule finishes the proof.

$$\cfrac{\cfrac{\cfrac{\neg(A \vee B) \qquad \cfrac{[A]}{A \vee B}\,{\scriptstyle \vee\mathbf{I}}}{\neg(A \vee B) \wedge (A \vee B)}\,{\scriptstyle \wedge\mathbf{I}}}{\neg A}\,{\scriptstyle \neg\mathbf{I}} \qquad \begin{array}{c}[B] \\ \vdots \\ \neg B\end{array}}{\neg A \wedge \neg B}\,{\scriptstyle \wedge\mathbf{I}}$$

□

Problem 84 $\neg A \vee \neg B$ *can be derived from* $\neg(A \wedge B)$. *(Another example of the so-called de Morgan laws.)*

Solution: This is more difficult! We want to conclude $\neg A \vee \neg B$, so the temptation is to try to derive one of $\neg A$ and $\neg B$. But which one?? This kind of problem in logic—the problem of deriving a disjunction without seeing a way to derive either disjunct separately—is related to the difference between non-classical logic and classical logic. In this course our logic is classical, so we use indirect inference.

Let us first think intuitively why $\neg A \vee \neg B$ should follow from $\neg(A \wedge B)$. Say, a dish does not contain both cream and meat. Why can we conclude that either cream is missing or meat is missing? Well, because if both cream and meat were there, we would contradict the made assumption, so one of cream and meat must be missing. We try to make this formal.

Let us still think intuitively why $\neg A \vee \neg B$ should follow from $\neg(A \wedge B)$. Let us assume $\neg A \vee \neg B$ is false i.e. $\neg(\neg A \vee \neg B)$ and work towards a contradiction. Now clearly $\neg A$ leads to a contradiction, so $\neg\neg A$ i.e. A. Respectively B. So $A \wedge B$. This contradicts the assumption $\neg(A \wedge B)$. So we get $\neg\neg(\neg A \vee \neg B)$ i.e. $\neg A \vee \neg B$.

For the derivation, see Figure 1.12.

We start be writing down the assumption $\neg(A \wedge B)$ and the denial $\neg(\neg A \vee \neg B)$ of what we try to prove. Since we have denied the our claim, we are using an indirect inference, and we try to derive a contradiction.

We observe, that if $\neg A$ was taken as a temporary assumption, we can derive $\neg A \lor \neg B$, which contradicts our assumption $\neg(\neg A \lor \neg B)$. So with the double negation trick we can conclude A, and eliminate the temporary assumption $\neg A$.

Now we do the same for B.

In the end we can derive $A \land B$, leading to a contradiction.

The deduction is ready (see see Figure 1.12).

□

Problem 85 *$A \lor \neg A$ is derivable. (This is the so called Law of Excluded Middle.)*

Solution: Intuition: We use indirect proof. So we assume $\neg(A \lor \neg A)$ and derive a contradiction. Now A leads to $A \lor \neg A$ and hence to a contradiction. Thus we may conclude $\neg A$. But this leads to $A \lor \neg A$ and hence to a contradiction, and we are done.

For the derivation, see Figure 1.13.

We start by writing down the main temporary assumption, namely the denial of the conclusion, that is, $\neg(A \lor \neg A)$. So from now on we try to reach a contradiction, in order to be able to use the $\neg$-Introduction Rule and the $\neg$-Elimination Rule and thereby reach the claim $A \lor \neg A$.

Now we make a new temporary assumption A. From this we get immediately $A \lor \neg A$, contrary to our first temporary assumption. So we $\neg A$ and can eliminate the temporary assumption A.

But from $\neg A$ we can derive $A \lor \neg A$, again a contradiction with our first—and now the only remaining—temporary assumption.

So we can use the $\neg$-Introduction Rule to derive the negation $\neg\neg(A \lor \neg A)$ of our only remaining temporary assumption, which, by the way, is now eliminated. So we get $A \lor \neg A$ and there are no remaining temporary assumptions, so we are done (see see Figure 1.13).

□

1.8.10 Problems

Problem 86 *Derive $(A \to B) \to (\neg B \to \neg A)$.*

Problem 87 *Derive $\neg A \lor B$ from $\neg\neg B \lor \neg A$.*

Problem 88 *Derive $\neg(A \land B)$ from $\neg A \lor \neg B$.*

Problem 89 *Derive $(\neg B \to \neg A) \to (A \to B)$.*

Problem 90 *Derive $A \land B$ from $A \land (B \lor C)$ and $\neg C$.*

Problem 91 *Derive $\neg((A \lor B) \land \neg A \land \neg B)$.*

Problem 92 *Derive $\neg(A \lor B)$ from $\neg A \land \neg B$.*

Problem 93 *Derive $\neg A \lor B$ from $\neg(A \land \neg B)$.*

Problem 94 *Derive $A \land \neg B$ from $\neg(\neg A \lor B)$.*

Problem 95 *Derive $(A \to B) \lor (B \to A)$.*

1.9 Natural deduction—Recap

We can collect now all the introduction and elimination rules of propositional logic to a table, see Figure 1.14.

1.10 Soundness

Soundness of deduction means that if we accept some formulas as true and then deduce another formula from them, then also that other formula is true. This is the whole point of logic.

More exactly, soundness of natural deduction means that deductions respect truth in the following sense: If a formula A can be derived from the assumptions $B_1, \ldots, B_n$, and $v(B_1) = \ldots = v(B_n) = 1$ for some valuation v, then also $v(A) = 1$.

Theorem 1.6 *Suppose v is a valuation. If A has a natural deduction from $B_1, \ldots, B_n$, and $v(B_1) = \ldots = v(B_n) = 1$, then $v(A) = 1$.*

Proof: The proof is "by induction" on the structure of a natural deduction. We show that every deduction is sound in the sense that if in any valuation the assumptions of the deduction have value 1, then so does the conclusion. We proceed from shorter deductions to longer ones. The shortest possible deduction consists of just one formula A and this formula is both the assumption and the conclusion of the deduction. Of course such a deduction is sound.

Next we look at deductions in which some rules have been actually used. Every deduction has a unique *conclusion* and some *last* rule that has been used to derive that

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg(\neg A \vee \neg B)] \quad \dfrac{[\neg A]}{\neg A \vee \neg B}\ {\scriptstyle\vee\mathbf{I}}}{(\neg A \vee \neg B) \wedge \neg(\neg A \vee \neg B)))}\ {\scriptstyle\wedge\mathbf{I}}}{\neg\neg A}\ {\scriptstyle\neg\mathbf{I}}}{A}\ {\scriptstyle\neg\mathbf{E}} \qquad \begin{array}{c}\neg B\\ \vdots\\ B\end{array}}{A \wedge B}\ {\scriptstyle\wedge\mathbf{I}} \qquad \neg(A \wedge B)}{(A \wedge B) \wedge \neg(A \wedge B)}\ {\scriptstyle\wedge\mathbf{I}}}{\neg\neg(\neg A \vee \neg B)}\ {\scriptstyle\neg\mathbf{I}}}{\neg A \vee \neg B}\ {\scriptstyle\neg\mathbf{E}}$$

Figure 1.12: Deriving $\neg A \vee \neg B$ from $\neg(A \wedge B)$

$$\dfrac{\dfrac{\dfrac{[\neg(A \vee \neg A)] \qquad \dfrac{\dfrac{\dfrac{[\neg(A \vee \neg A))] \quad \dfrac{[A]}{A \vee \neg A}\ {\scriptstyle\vee\mathbf{I}}}{\neg(A \vee \neg A) \wedge (A \vee \neg A)}\ {\scriptstyle\wedge\mathbf{I}}}{\neg A}\ {\scriptstyle\neg\mathbf{I}}}{A \vee \neg A}\ {\scriptstyle\vee\mathbf{I}}}{\neg(A \vee \neg A) \wedge (A \vee \neg A)}\ {\scriptstyle\wedge\mathbf{I}}}{\neg\neg(A \vee \neg A)}\ {\scriptstyle\neg\mathbf{I}}}{A \vee \neg A}\ {\scriptstyle\neg\mathbf{E}}$$

Figure 1.13: Deriving $A \vee \neg A$.

Connective	Introduction	Elimination
Conjunction	$\dfrac{A \quad B}{A \wedge B} \wedge\mathbf{I}$	$\dfrac{A \wedge B}{A} \wedge\mathbf{E}$ $\quad$ $\dfrac{A \wedge B}{B} \wedge\mathbf{E}$
Disjunction	$\dfrac{A}{A \vee B} \vee\mathbf{I}$ $\quad$ $\dfrac{B}{A \vee B} \vee\mathbf{I}$	$\dfrac{A \vee B \quad \begin{array}{c}[A]\\ \vdots \\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ C\end{array}}{C} \vee\mathbf{E}$
Implication	$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \to B} \to\mathbf{I}$	$\dfrac{A \to B \quad A}{B} \to\mathbf{E}$
Equivalence	$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ A\end{array}}{A \leftrightarrow B} \leftrightarrow\mathbf{I}$	$\dfrac{A \leftrightarrow B \quad A}{B} \leftrightarrow\mathbf{E}$ $\quad$ $\dfrac{A \leftrightarrow B \quad B}{A} \leftrightarrow\mathbf{E}$
Negation	$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B \wedge \neg B\end{array}}{\neg A} \neg\mathbf{I}$	$\dfrac{\neg\neg A}{A} \neg\mathbf{E}$

Figure 1.14: The rules of natural deduction.

conclusion. All the deductions that have been built before the last rule was applied are shorter. Since we assume that shorter deductions are sound, we may assume that when the last rule is applied all the previous deductions are sound.

So if we assume that a valuation v give value 1 to the assumptions of a big deduction, the value 1 in a sense "flows" along the rules until it reaches the conclusion, and then the conclusion also has truth value 1.

1. Conjunction introduction rule

$$\frac{A \quad B}{A \wedge B} \wedge\mathbf{I}$$

We assume $v(A) = v(B) = 1$. We show $v(A \wedge B) = 1$. But this is trivial! Almost all cases of this proof are trivial, because the inference rule and the definition of the truth value are both based on the same idea. So we have set up the system so that this proof goes through smoothly.

2. Conjunction elimination rule

$$\frac{A \wedge B}{A} \wedge\mathbf{E} \qquad \frac{A \wedge B}{B} \wedge\mathbf{E}$$

We assume $v(A \wedge B) = 1$. We show $v(A) = v(B) = 1$. But this is again trivial!

3. Disjunction introduction rule

$$\frac{A}{A \vee B} \vee\mathbf{I} \qquad \frac{B}{A \vee B} \vee\mathbf{I}$$

We assume $v(A) = 1$. We show $v(A \vee B) = 1$. But this is trivial!

In the other case, we assume $v(B) = 1$. We show $v(A \vee B) = 1$. Again, this is trivial!

4. Disjunction elimination rule

$$\frac{A \vee B \quad \begin{array}{c}[A]\\ \vdots\\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots\\ C\end{array}}{C} \vee\mathbf{E}$$

We assume $v(A \vee B) = 1$. We also assume that the derivation of C from A, as well as the derivation of C from B, are sound i.e. if $v(A) = 1$, then $v(C) = 1$, and if $v(B) = 1$, then $v(C) = 1$. We show $v(C) = 1$. But $v(A \vee B) = 1$ implies $v(A) = 1$ or $v(B) = 1$. In either case we have $v(C) = 1$. Hence, indeed $v(C) = 1$.

5. Implication introduction rule

$$\frac{\begin{array}{c}[A]\\ \vdots\\ B\end{array}}{A \to B} \to\mathbf{I}$$

We assume that the derivation of B from A is sound, i.e. if $v(A) = 1$, then $v(B) = 1$. We prove $v(A \to B) = 1$. Case 1: $v(A) = 0$. Clear. Case 2: $v(A) = 1$. By assumption, in this case $v(B) = 1$, so $v(A \to B)$=1.

6. Implication elimination rule

$$\frac{A \to B \quad A}{B} \to\mathbf{E}$$

We assume $v(A \to B) = v(A) = 1$. We show $v(B) = 1$. This is trivial!

7. Equivalence introduction rule

$$\frac{\begin{array}{c}[A]\\ \vdots\\ B\end{array} \quad \begin{array}{c}[B]\\ \vdots\\ A\end{array}}{A \leftrightarrow B} \leftrightarrow\mathbf{I}$$

We leave both the formulation of the claim, and the details of the proof as an exercise.

8. Equivalence elimination rule

$$\frac{A \leftrightarrow B \quad A}{B} \leftrightarrow\mathbf{E} \qquad \frac{A \leftrightarrow B \quad B}{A} \leftrightarrow\mathbf{E}$$

We leave both the formulation of the claim, and the details of the proof as an exercise.

9. Negation introduction rule

$$\frac{\begin{array}{c}[A]\\ \vdots\\ B \wedge \neg B\end{array}}{\neg A} \neg\mathbf{I}$$

We assume that the inference of $B \wedge \neg B$ from A is sound i.e. if $v(A) = 1$, then $v(B \wedge \neg B) = 1$. But $v(B \wedge \neg B) = 0$ always. So $v(A) = 0$. Hence $v(\neg A) = 1$.

10. Negation elimination rule

$$\frac{\neg\neg A}{A}\ \neg\mathbf{E}$$

We assume $v(\neg\neg A) = 1$. We show $v(A) = 1$. Clear!

□

1.10.1 Soundness Theorem

The **Soundness Theorem** of propositional logic says that if a propositional formula has a natural deduction, then it is a tautology. More generally, if a propositional formula A has a natural deduction from assumptions which have truth value 1 in a valuation v, then also $v(A) = 1$.

We have just proved this important basic fact. It is remarkable that also the converse is true: if a propositional formula is a tautology, then it has a natural deduction. This is called the **Completeness Theorem** of propositional logic. Its proof is not very difficult but we omit it.

1.10.2 Applications of Soundness

We can show that a formula B is not derivable by natural deduction from a formula A by finding a valuation v such that $v(A) = 1$ and $v(B) = 0$. For example, we can show that $p_0 \vee (p_1 \wedge p_2)$ is not derivable from $(p_0 \vee p_2) \rightarrow p_1$ by letting $v(p_0) = v(p_1) = v(p_2) = 0$. Then $v((p_0 \vee p_2) \rightarrow p_1) = 1$, but $v(p_0 \vee (p_1 \wedge p_2)) = 0$. It is important and useful to be able to write derivations, but it is equally important to be able to say why in some cases a derivation is not possible.

1.10.3 Solved Problems

Problem 96 *Show that the following inference is not correct:*

- *Suppose $x > 10$ or $y > 10$.*
- *Suppose additionally that not both $x > 10$ and $y > 10$.*
- *Then if not $x > 10$, then not $y > 10$.*

Solution: Denote "$x > 10$" by p_0 and "$y > 10$" by p_1. The assumptions are $p_0 \vee p_1$ and $\neg(p_0 \wedge p_1)$. The conclusion is $\neg p_0 \rightarrow \neg p_1$.

The problem is to show that there is no natural deduction of $\neg p_0 \rightarrow \neg p_1$ from $p_0 \vee p_1$ and $\neg(p_0 \wedge p_1)$.

The claim follows from the Soundness Theorem as follows: If we let $v(p_0) = 0$ and $v(p_1) = 1$, then $v(p_0 \vee p_1) = 1$ and $v(\neg(p_0 \wedge p_1)) = 1$ but $v(\neg p_0 \rightarrow \neg p_1) = 0$. So there cannot be any natural deduction of $\neg p_0 \rightarrow \neg p_1$ from $p_0 \vee p_1$ and $\neg(p_0 \wedge p_1)$.

Note that to find valuations v with prescribed properties one can use truth tables! □

Problem 97 *Show that there is no natural deduction of $p_0 \wedge \neg p_1$ from $\neg(p_0 \vee p_1)$*

Solution: If $v(p_0) = v(p_1) = 0$, then $v(\neg(p_0 \vee p_1)) = 1$ but $v(p_0 \wedge \neg p_1) = 0$. So by the Soundness Theorem $p_0 \wedge \neg p_1$ cannot have a natural deduction from $\neg(p_0 \vee p_1)$. □

Problem 98 *Show that the following inference is incorrect:*

- *The train is moving and in addition either the door is open or the green light is on.*
- *It is not the case that the green light is not on.*
- *Hence the train is moving and the door is open.*

Solution: Denote "train is moving" by p_0, "door is open" by p_1, and "green light is on" by p_2.

Let A be the formula $p_0 \wedge p_1$, B the formula $p_0 \wedge (p_1 \vee p_2)$ and C the formula $\neg\neg p_2$.

We are asked, why we cannot infer A from B and C? In other words, we are asked to show that there is no natural deduction of A from B and C.

If $v(p_1) = 0$ and $v(p_0) = v(p_2) = 1$, then $v(B) = 1$ and $v(C) = 1$ but $v(A) = 0$. So by the Soundness Theorem, A cannot have a natural deduction from B and C.

□

1.10.4 Problems

Problem 99 *Show that the following inference is not correct:*

1. *If grandmother can fly, then grandmother is not a stone.*
2. *Grandmother cannot fly.*
3. *Hence grandmother is a stone.*

Problem 100 *Show that the following inference is not correct:*

1. *If it rains in Warsaw, then it rains in Vienna or it snows in Helsinki.*
2. *It does not rain in Vienna, but it snows in Helsinki.*
3. *Hence it rains in Warsaw.*

Problem 101 *Show that there is no natural deduction of* $\neg(p_0 \to p_1)$ *from* $\neg p_0 \to p_1$

Problem 102 *Show that there is no natural deduction of* $\neg p_0 \land \neg p_1$ *from* $\neg(p_0 \land p_1)$

Problem 103 *Show that there is no natural deduction of* $\neg(p_0 \lor p_1)$ *from* $\neg p_0 \lor \neg p_1$

Problem 104 *Show that the following inference is incorrect:*

1. *If the envelope contains a password and the green light is on, then the door can be opened.*
2. *The green light is not on.*
3. *Hence if the door cannot be opened, the envelope does not contain a password.*

1.11 Semantic trees

Semantic proofs (also known as tableaux methods) are among the most effective and easiest to use. In a semantic proof of A we show that the negation $\neg A$ of A is not satisfiable, by building a so called semantic tree for $\neg A$ and observing that the tree *closes*, as we say. The closing of the tree implies that A must be a tautology. Intuitively the closing of the tree means that all alternatives for satisfying $\neg A$ have been checked and found impossible, so A must be a tautology.

A semantic tree for a formula A codes a contemplation on what else must be true if A is true. For example, if $A \land B$ is true, then A and B must be true, so we write them underneath $A \land B$:

$$\begin{array}{c} A \land B \\ | \\ A \\ | \\ B \end{array}$$

If $A \lor B$ is true, then A or B is true but we do not know which, so we split the tree at this point into two branches, one for A and the other for B.

$$\begin{array}{c} A \lor B \\ \widehat{A \quad B} \end{array}$$

1.11.1 The rules for semantic trees

The rules of forming semantic trees follow the idea that we write underneath a formula whatever immediately follows from the formula in view of its main connective. For each connective (except negation) we need two rules. The rules are as follows (See also Figure 2.26):

- Disjunction:

$$\begin{array}{c} A \lor B \\ \widehat{A \quad B} \end{array} \qquad\qquad \begin{array}{c} \neg(A \lor B) \\ | \\ \neg A \\ | \\ \neg B \end{array}$$

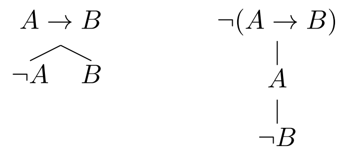
- Conjunction:

$$\begin{array}{c} A \land B \\ | \\ A \\ | \\ B \end{array} \qquad\qquad \begin{array}{c} \neg(A \land B) \\ \widehat{\neg A \quad \neg B} \end{array}$$

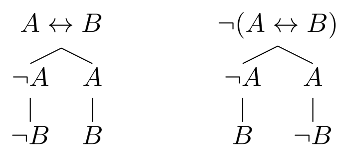
- Negation:

$$\begin{array}{c} \neg\neg A \\ | \\ A \end{array}$$

- Implication:



- Equivalence:

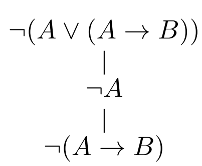


1.11.2 Semantic proofs

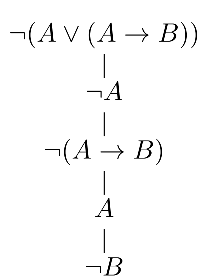
A branch of a semantic tree is *closed* (or *closes*) if it contains both B and $\neg B$ for some B. A semantic proof of A is a semantic tree for $\neg A$ in which all branches are closed. The semantic proof demonstrates that A is a tautology by showing that the assumption that $\neg A$ is true leads to a contradiction, namely that both B and $\neg B$ are true for some B.

1.11.3 An example

A semantic proof of $A \vee (A \to B)$ is a semantic tree for $\neg(A \vee (A \to B))$ in which all branches close. We shall now build this tree, step by step. First we use the disjunction rule:



Then we use the implication rule:

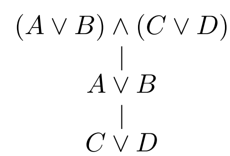


The resulting tree has only one branch and this branch closes because it has both A and $\neg A$. Hence this tree is a semantic proof of $A \vee (A \to B)$. One can say that we tried what would $\neg(A \vee (A \to B))$ be like, and we found that it leads to a contradiction (A and $\neg A$), so we have to abandon $\neg(A \vee (A \to B))$, and this is a proof that $A \vee (A \to B)$ itself is a tautology, as it is.

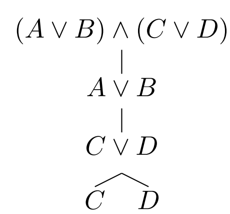
1.11.4 Splitting

Applying the relevant rule to a formula of the form $A \vee B$ or to a formula of the form $\neg(A \wedge B)$ causes the tree to split. It is important that the splitting takes place at the end of each branch going through the formula. See next example!

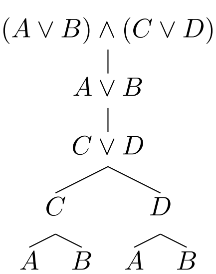
We form the semantic tree of $(A \vee B) \wedge (C \vee D)$. First we use the conjunction rule:



Then we use the disjunction rule to $C \vee D$ and the tree splits:



But we can still use the disjunction rule to $A \vee B$. SO we split both branches that go through $A \vee B$:



Connective	Rule	Rule for the negation
Disjunction	$A \vee B$ $A \quad B$	$\neg(A \vee B)$ $\vert$ $\neg A$ $\vert$ $\neg B$
Conjunction	$A \wedge B$ $\vert$ A $\vert$ B	$\neg(A \wedge B)$ $\neg A \quad \neg B$
Negation	$\neg\neg A$ $\vert$ A	
Implication	$A \rightarrow B$ $\neg A \quad B$	$\neg(A \rightarrow B)$ $\vert$ A $\vert$ $\neg B$
Equivalence	$A \leftrightarrow B$ $\neg A \quad A$ $\vert \quad \vert$ $\neg B \quad B$	$\neg(A \leftrightarrow B)$ $\neg A \quad A$ $\vert \quad \vert$ $B \quad \neg B$

Figure 1.15: The rules of semantic trees.

1.11.5 Completeness of the semantic tree method

The method of semantics tree*s* satisfies the *Soundness Theorem*: Every formula that has a semantic proof, is a tautology.

Even better: The method of semantics tree*s* satisfies the *Completeness Theorem*: A formula has a semantic proof if and only if it is a tautology.

Proofs are not hard, but still omitted.

1.11.6 Solved Problems

Problem 105 *Give a semantic proof of* $\neg(A \wedge B) \rightarrow (\neg A \vee \neg B)$

Solution:

$$\begin{array}{c}
\neg(\neg(A \wedge B) \rightarrow (\neg A \vee \neg B)) \\
| \\
\neg(A \wedge B) \\
| \\
\neg(\neg A \vee \neg B) \\
| \\
\neg\neg A \\
| \\
\neg\neg B \\
\diagup \quad \diagdown \\
\neg A \qquad \neg B
\end{array}$$

The tree has two branches. One has $\neg A$ and $\neg\neg A$ on it, the other has $\neg B$ and $\neg\neg B$ on it. So the tree is a closed tree. □

Problem 106 *Give a semantic proof of* $(A \vee B) \rightarrow (B \vee A)$.

Solution:

$$\begin{array}{c}
\neg((A \vee B) \rightarrow (B \vee A)) \\
| \\
A \vee B \\
| \\
\neg(B \vee A) \\
| \\
\neg B \\
| \\
\neg A \\
\diagup \quad \diagdown \\
A \qquad B
\end{array}$$

The tree has two branches. One has A and $\neg A$ on it, the other has B and $\neg B$ on it. So the tree is a closed tree. □

Problem 107 *Give a semantic proof of* $(A \wedge B) \rightarrow (B \wedge A)$.

Solution:

$$\begin{array}{c}
\neg((A \wedge B) \rightarrow (B \wedge A)) \\
| \\
A \wedge B \\
| \\
\neg(B \wedge A) \\
| \\
A \\
| \\
B \\
\diagup \quad \diagdown \\
\neg A \qquad \neg B
\end{array}$$

This tree is closed like the previous tree. □

Problem 108 *Give a semantic proof of* $\neg A \rightarrow (A \rightarrow B)$.

Solution:

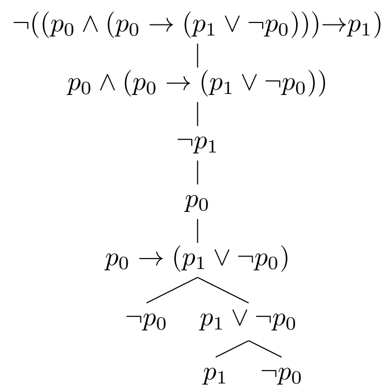
$$\begin{array}{c}
\neg(\neg A \rightarrow (A \rightarrow B)) \\
| \\
\neg A \\
| \\
\neg(A \rightarrow B) \\
| \\
A \\
| \\
\neg B
\end{array}$$

There is only one branch. On this single branch we have A and $\neg A$, so the branch s closed, and so is the entire tree.

□

Problem 109 *Example: Give a semantic proof of* $(p_0 \wedge (p_0 \rightarrow (p_1 \vee \neg p_0))) \rightarrow p_1$.

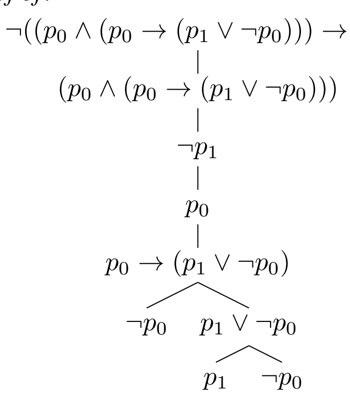
Solution:



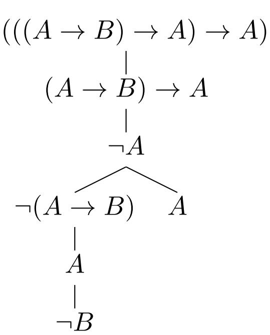
Each branch of this tree closes. So this tree is the required semantic proof. □

1.11.7 Problems

Problem 110 *Analyze the below semantic tree: Which rules are used, why branches close? What is this a semantic proof of?*



Problem 111 *Analyze the below semantic tree: Which rules are used, why branches close? What is this a semantic proof of?*



Problem 112 *Give a semantic proof of* $(A \wedge C) \to (A \wedge (B \vee C))$.

Problem 113 *Give a semantic proof of* $(A \wedge (B \vee C) \wedge \neg C) \to (A \wedge B)$.

Problem 114 *Give a semantic proof of* $((A \wedge B) \vee (A \wedge C)) \to (A \wedge (B \vee C))$.

Problem 115 *Give a semantic proof of* $(A \vee (B \wedge C)) \to ((A \vee B) \wedge (A \vee C))$.

Problem 116 *Give a semantic proof of* $((A \vee B) \wedge (A \vee C)) \to (A \vee (B \wedge C))$.

Problem 117 *Give a semantic proof of* $(A \wedge C) \to (A \wedge (B \vee C))$.

Chapter 2

Predicate logic

2.1 Introduction

When we move from propositional logic to predicate logic, it is a whole new ball game. We try to do the same as in propositional logic—describe situations in the world, in a database, in mathematics, etc—in an exact way so that we can coherently define and study concepts such as proof and truth, perhaps in a way that even a computer can understand. But in predicate logic we have a far richer language than in propositional logic. Predicate logic deals with properties of elements and relations between elements of a domain[1]. We can talk about universal properties and existence of solutions of equations. We can go much deeper into the phenomena that we are interested in. While in propositional logic the sentence

"Some birds do not fly but some mammals do."

is formalized as a mere conjunction

$$p_0 \wedge p_1,$$

in predicate logic it can be written as

$$\exists x(B(x) \wedge \neg F(x)) \wedge \exists y(M(y) \wedge F(y)),$$

which is clearly much more informative. But now we need new rules. Introduction and elimination rules for conjunction, disjunction, negation, implication and equivalence say nothing about the symbol $\exists$. It also turns out that the concept of valuation is not informative enough to account for the much richer semantics—meaning theory—of predicate logic. We need the concept of a structure—or a model—and that is how we start our investigation of predicate logic.

[1]Also called *universe.*

2.2 Some structures

We shall first discuss some particular examples of structures i.e. models.

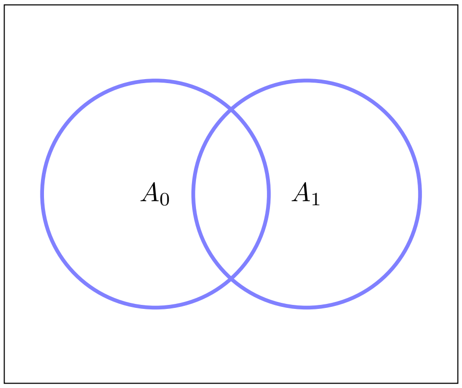
2.2.1 Unary structures

A *unary structure* $\mathcal{M}$ consists of a non-empty *domain* M, also called the *universe*, and a number of subsets of it, called *(unary) predicates*. The predicates are denoted $A_0, A_1, ...$ Examples of unary predicates on any set M are

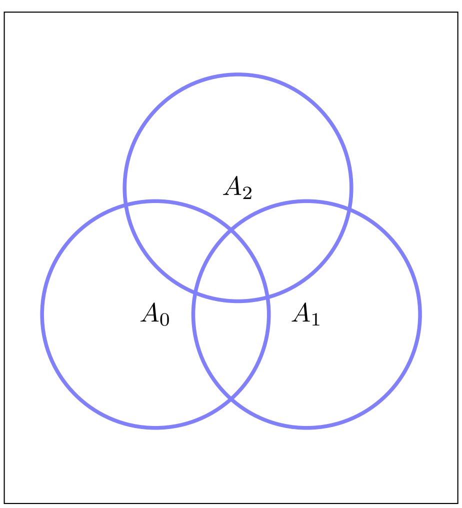
Empty predicate	$\emptyset$
Full predicate	M
Singleton predicate	$\{a\}$

One predicate divides the domain into up to two parts, two predicates divide the domain into up to four parts, etc. We denote a unary structure with with domain M and predicates $A_1, \ldots, A_n$ by

$$\mathcal{M} = (M, A_1, \ldots, A_n).$$



For example, M could be a set of people and A_0 the set of women in M. Now M divides into people who are in A_0, i.e. are women, and those who are not in A_0, i.e. are men. Or, if again M is a set of people, A_0 could be the set of country music lovers in M and A_1 the set of jazz fans in M. Then M would divide into those who are in A_0 but not in A_1, those who are in both in A_0 and A_1, those who are in A_1 but not in A_0, and finally those who are neither in A_0 nor in A_1. This kind of classification of elements according to what is known about them is typical in predicate logic. Note that some of the four sets can be empty, but altogether they have as many elements as M has, as every element of M is in one of the sets. If we add a third predicate A_2, those in M who like classical music, we get a division of M into 8 parts:



1. Likes classical music, but neither country nor jazz music.
2. Likes country music, but neither classical nor jazz music.
3. Likes jazz, but neither classical nor country music.
4. Likes classical and country music, but not jazz.
5. Likes classical music and jazz, but not country music.
6. Likes country music and jazz, but not classical music.
7. Does not like classical, country or jazz music.
8. Likes classical, country and jazz music.

Again, some of those sets may be empty, depending on M, but all in all they have all the elements of M so the sum of the numbers of people in each set equals the total number of people in M.

2.2.2 Tile models

A *tile model* consists of colored tiles arranged in a row as the five tiles below (we indicate colors by letters: “R” is red, “B” is blue and “Y” is yellow):

| R | B | B | R | Y |

Example 2.1 *Examples of tile models*

| *R* | *R* | *R* | *R* | *R* |

| *B* | *B* | *B* | *B* |

| *R* | *B* | *R* |

| *Y* | *R* | *R* | *B* | *B* | *R* | *Y* |

| *B* | *B* | *Y* | *R* | *B* | *Y* | *B* | *R* | *R* | *B* | *R* | *Y* |

The relevant properties of the tiles are color and position (which is the property of being left or right of another tile).

2.2.3 A mathematical definition of tile models

A *tile model* $\mathcal{T}$ consists of a finite set T of objects we call *tiles*. For each tile x exactly one of the predicates $B^{\mathcal{T}}(x)$ i.e. "x is blue", $R^{\mathcal{T}}(x)$ i.e. "x is red", $Y^{\mathcal{T}}(x)$ i.e. "x is Yellow" holds. There is a linear order $<^{\mathcal{T}}$ defined on T. If $x <^{\mathcal{T}} y$, we say x is "left of" y and "y is right of x". A *linear order* on a finite set is a specification of the order of the elements: which is the first, which comes next, etc.

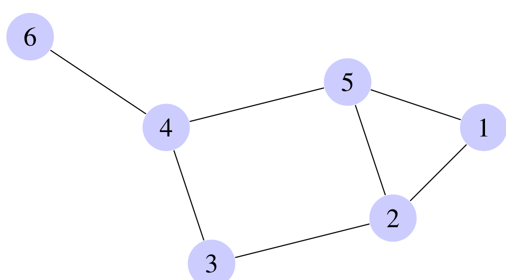
Tile models can appear in various disguises. For example, it is not important what the colors are as long as we distinguish them from each other. We can think words made of three letters, say B, R and Y, or even just any three distinct symbols, and interpret them as tile models.

We denote tiles

$$\mathcal{T} = (T, B^{\mathcal{T}}, R^{\mathcal{T}}, Y^{\mathcal{T}}, <^{\mathcal{T}}).$$

2.2.4 Graphs

A graph consists of vertices (or nodes) and edges (or lines) between the vertices as in the following picture:



Vertices connected by an edge are called *neighbors*.

Many problems of practical interest can be represented by graphs. Here are some examples: data organization, telecommunication, communication networks, road networks, networks of communication, various structures in natural language, bioinformatics, genomics, molecular chemistry, etc.

in mathematics graphs are used for example in combinatorics, geometry, topology, and group theory.

A mathematical definition of graphs is as follows:

Definition 2.2 *A graph $\mathcal{G}$ consists of a domain G, called the set (or* universe*) of vertices, and a binary predicate xEy (more exactly $xE^{\mathcal{G}}y$) called the edge relation.*

We denote graphs with domain G and edge relation E by

$$\mathcal{G} = (G, E).$$

If xEy, then x is called a neighbor of y and vice versa. In a graph no vertex is a neighbor of itself (Antireflexivity). Also, if xEy then yEx (Symmetry). Sometimes we associate colors to vertices and then the graph is called a *colored graph*.

2.2.5 Natural numbers

The structure $\mathcal{N}$ of the natural numbers consists of the set $\mathbb{N}$ of non-negative integers $0, 1, 2, \ldots$. They have a natural order $x < y$ (more exactly $<^{\mathcal{N}}$) in which 0 is the smallest element and for every element x there is a bigger one, namely $x + 1$. We denote the ordered set of natural numbers by

$$\mathcal{N} = (N, <).$$

In picture:

2.2.6 Other structures (some with functions—see Section 2.20)

Examples of other common structures in mathematics and computer science are

- Directed graphs.
- Equivalence relations.
- Groups.
- Fields.
- Boolean algebras.
- Lattices.
- Linear orders.
- Partial orders.
- Trees.

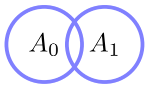
Unary structure 	$\mathcal{M} = (M, A_0, \ldots, A_{n-1})$	$A_0 \subseteq M, \ldots, A_{n-1} \subseteq M$
Tile model R B B R Y	$\mathcal{T} = (T, B^{\mathcal{T}}, R^{\mathcal{T}}, Y^{\mathcal{T}}, <^{\mathcal{T}})$	$B^{\mathcal{T}} \subseteq T$, $R^{\mathcal{T}} \subseteq T$, $Y^{\mathcal{T}} \subseteq T$, $<^{\mathcal{T}} \subseteq T \times T$, T is the disjoint union of $B^{\mathcal{T}}$, $R^{\mathcal{T}}$ and $Y^{\mathcal{T}}$, $<^{\mathcal{T}}$ is a linear order (See Subsection 2.15.3).
Natural numbers	$\mathcal{N} = (\mathbb{N}, <)$	$<$ is the natural order $0 < 1 < 2 < 3 < \ldots$.

Figure 2.1: Some structures i.e. models

2.2.7 Solved problems

Problem 118 *Argue that the following statement is true in general:*

> *If every millionaire is happy or not busy,*
> *then every busy millionaire is happy.*

Solution: Take a busy millionaire, call it x. We know that x is happy or not busy. But we assumed x is busy. So x must be happy. This informal inference is of the kind that is very typical to predicate logic, as we shall see when we define natural deduction in predicate logic in a moment. □

Problem 119 *Use unary structures to show that the following statement is not true in general:*

> *If every rainy day in August is windy,*
> *then some windy day in August is rainy.*

Solution: Let M be a non-empty set of (hypothetical) August days, A_0 the set of rainy days in M, and A_1 the set of windy days in M. The statement then says "If A_0 is contained in A_1, then some element of A_1 is in A_0." It is clear that in such generality this cannot be true. For example, if A_0 is empty, then A_0 is contained in A_1 but no element of A_1 is in A_0. □

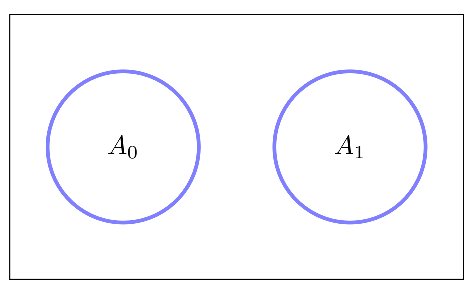
Problem 120 *Use unary structures to argue that the following statement is not true in general:*

> *If there are millionaires and every millionaire is*
> *happy or not busy, then no busy millionaire is happy.*

Solution: Let M be a non-empty set of (hypothetical) millionaires, A_0 the set of happy millionaires in M, and A_1 the set of busy millionaires in M. The statement then says "If every element of M is in A_0 or in the complement of A_1, then no element of A_1 is in A_0." In such generality this cannot be true. For example, if $M = \{Tom\}$, $A_0 = M$, and $A_1 = M$, then every element of M is in A_0 or in the complement of A_1, but some element of A_1 is in A_0. □

Problem 121 *Draw a unary structure where two predicates divide the domain into three parts. Give an everyday life example.*

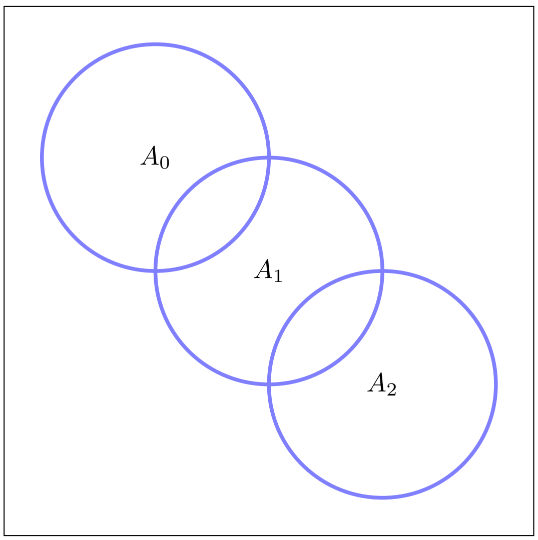
Solution: Let M be the set of all Finns, A_0 the set of Finns that are taller than 190 cm, and A_1 the set of Finns that are shorter than 170 cm.



□

Problem 122 *Draw a unary structure where three predicates divide the domain into six parts. Give an everyday life example.*

Solution: Let M be the set of all Europeans, A_0 the set of Europeans who live in Finland, A_1 the set of Europeans who speak French, and A_2 the set of Europeans who live in Italy.



□

Problem 123 *Is it true that if a tile model has left of every blue tile a red tile, then it has right of every red tile a blue tile?*

Solution: In model (2.1) it is indeed true that left of every blue tile is a red tile, and right of every red tile is a blue tile.

R	B	R	B	(2.1)

However in (2.2), again left of every blue tile is a red tile, but right of the last red tile there are no blue (or other) tiles.

R	B	R	B	R	(2.2)

The case (2.3) is interesting: Left of each of the zero blue tiles is a red tile, but it is certainly not true that right of every red tile there is a blue tile

R	R	R	(2.3)

So the answer to the question is no.
□

Problem 124 *Is it true that if a tile model has left of every blue or yellow tile a red tile, and the model has a red tile, then it has a yellow tile ?*

Solution: In model (2.4) it is true that left of every blue or yellow tile is a red tile, there is a red tile, and indeed there is also a yellow tile.

R	Y	B	(2.4)

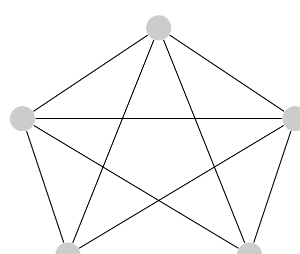
However in (2.5) left of every blue or yellow tile there is a red tile, and there is a red tile, but there are no yellow tiles.

R	B	R	B	(2.5)

So the answer to the question is no.
□

Problem 125 *Suppose a graph has 10 edges. How many vertices must it at least have? What if the graph has 100 edges?*

Solution: Every edge is connected to two vertices. From n vertices one gets thus at most $n(n-1)/2$ edges. For $n < 5$ we get < 7 edges. For $n = 5$ it is possible to get 10 edges, as the below graph shows. For 100 edges we need 15 vertices.



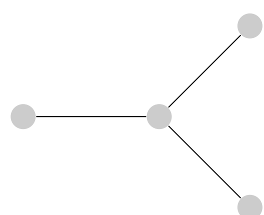
□

Problem 126 *Argue that if a graph has at least two vertices and some vertex has every other vertex as a neighbor, then every vertex has a neighbor.*

Solution: Suppose x is a vertex which has every other vertex as its neighbor. Suppose then y is an arbitrary vertex. If y is not x, then x is a neighbor of y, so y has a neighbor. If $y = x$, we argue as follows: Since the graph has at least two vertices, there is a vertex $z \neq x$. By our choice of x, z is a neighbor of x. But $x = y$. So again y has a neighbor. □

Problem 127 *Show that the following statement is not true in general in graphs: If some vertex has every other vertex as its neighbor, then every vertex is neighbor to a vertex that has every other vertex as a neighbor.*

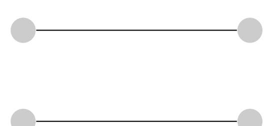
Solution: In the graph of the picture the middle vertex has every other vertex as its neighbor, but its neighbors are the three extreme vertices and they do not have the property that every other vertex is their neighbor.



□

Problem 128 *Show that the following statement is not true in general in graphs: If every vertex has a neighbor, then some vertex has every other vertex as a neighbor.*

Solution: In the graph of the picture every vertex has a neighbor, but no vertex has every other vertex as its neighbor.



□

Problem 129 *Suppose a graph has 6 vertices. Show that there are 3 vertices that are all neighbors of each other or all non-neighbors of each other.*

Solution: Take one vertex, call it v. The remaining vertices are divided into a set A of neighbors of v and a set B of non-neighbors of v. Since A and B together have 5 elements, one of them must have at least 3 elements. Suppose it is A. If the elements of A are not neighbors of each other, we are done.

So let us assume two elements of A are neighbors. Let us call them u and w. Now u,v and w are all neighbors.

We have to consider also the case that it is B which has at least 3 elements.

We proceed in the same way. If all the elements of B are neighbors, then we are done. Otherwise there are u and w in B that are non-neighbors. Now u, v and w form a triple of elements that are all non-neighbors of each other. □

2.2.8 Problems

Problem 130 *Draw a unary structure where three predicates divide the domain into seven parts. Give an everyday life example.*

Problem 131 *Give an example of a tile model that has several tiles in each of the three colors, and between any two tiles of the same color a tile of a different color? Why cannot we require that between any two tiles of the same color there is a tile of the same color?*

Problem 132 *Is it true that if a tile model has left of every blue or yellow tile a red tile, then the leftmost tile is red?*

Problem 133 *Is it true that if a tile model has left of every blue or yellow tile exactly three red tiles, then all the red tiles are left of all the blue or yellow tiles?*

Problem 134 *Show that the following claim is false: Suppose a graph has 5 vertices. Then there are 3 vertices that are all neighbors of each other or all non-neighbors of each other.*

2.3 More structures

2.3.1 The general concept of structure

All the structures that we have considered have the following in common: There is a universe (or a domain) which is an arbitrary non-empty set. There are some unary predicates, such as "red", "man". There are binary relations (also called binary predicates), such as "right of", "left of", "greater than". There are some distinguished elements, such as "zero". Names of these predicates, relations and elements constitute the *vocabulary* of the structure.

2.3.2 Relations

A binary relation on a set M is any collection R of elements of the Cartesian product

$$M^2 = M \times M = \{(a, b) : a, b \in M\}.$$

If (a, b) is in R, we write aRb, for simplicity. Examples of relations on any set M are

Empty relation	$\emptyset$
Full relation	M^2
Identity relation	$\{(a, b) \in M^2 : a = b\}$
Non-identity relation	$\{(a, b) \in M^2 : a \neq b\}$
Singleton relation	$\{(a, b)\}$
1st projection relation	$\{(a, b) \in M^2 : a = c\}$
2nd projection relation	$\{(a, b) \in M^2 : b = c\}$

2.3.3 Kinds of relations

Binary relations are very common. Already among people there are many familiar binary relations, such as "x knows y", "x and y are cousins", etc. Special important properties of binary relations have emerged and have been given a name:

A binary relation is

- *Symmetric* if aRb implies bRa
- *Reflexive* if always aRa
- *Transitive* if aRb and bRc imply aRc
- *Antisymmetric* if aRb implies not bRa
- *Antireflexive* if never aRa

Which of these properties do the relations "x knows y" and "x and y are cousins" have? What about the relation $x < y$ in the structure of natural numbers?

2.3.4 Vocabulary

Ordinarily the vocabulary of a language is the set of all the words that the sentences of the language are made of. In predicate logic we use the word "vocabulary" a little differently, although the spirit is the same:

A vocabulary is a finite collection of

- Unary predicate symbols $P_0, P_1, ...$
- Binary predicate symbols $R_0, R_1, ...$
- Constant symbols $c_0, c_1, ...$

We could treat also n-ary predicate symbols for $n > 2$ as well as function symbols. In practice we use also other symbols for predicate and constant symbols.

2.3.5 Structure

Definition 2.3 *A structure (or a model) $\mathcal{M}$ for a vocabulary L is a non-empty set M, called the universe, or domain, of $\mathcal{M}$, and:*

- *A unary predicate $P^{\mathcal{M}}$ on M for every unary predicate symbol P in L.*
- *A binary relation $R^{\mathcal{M}}$ of $M \times M$ for every binary predicate symbol R in L.*
- *An element $c^{\mathcal{M}}$ of M for every constant symbol c in L.*

The set $P^{\mathcal{M}}$ is called the *interpretation* of the symbol P in $\mathcal{M}$. Similarly, The relation $R^{\mathcal{M}}$ is called the *interpretation* of the symbol R in $\mathcal{M}$. Finally, the element $c^{\mathcal{M}}$ is the interpretation of the symbol c in $\mathcal{M}$.

A common notation for structures is the following:

Vocabulary	Structure
P	$M = (M, P^{\mathcal{M}})$
P, R	$M = (M, P^{\mathcal{M}}, R^{\mathcal{M}})$
P, R, c	$M = (M, P^{\mathcal{M}}, R^{\mathcal{M}}, c^{\mathcal{M}})$
etc	

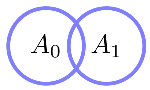
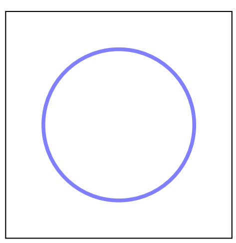
Unary structure 	$L = \{P_0, \ldots, P_{n-1}\}$	$\mathcal{M} = (M, P_0^{\mathcal{M}}, \ldots, P_{n-1}^{\mathcal{M}})$
Binary structure	$L = \{R_0, \ldots, R_{n-1}\}$	$\mathcal{M} = (M, R_0^{\mathcal{M}}, \ldots, R_{n-1}^{\mathcal{M}})$
Mixed structure	$L = \{P_0, R_0, c\}$	$\mathcal{M} = (M, P_0^{\mathcal{M}}, R_0^{\mathcal{M}}, c^{\mathcal{M}})$
Tile model R B B R Y	$L = \{B, R, Y, <\}$	$\mathcal{T} = (T, B^{\mathcal{T}}, R^{\mathcal{T}}, Y^{\mathcal{T}}, <^{\mathcal{T}})$
Natural numbers	$L = \{<\}$	$\mathcal{N} = (\mathbb{N}, <^{\mathcal{N}})$

Figure 2.2: Some vocabularies and structures

2.3.6 Solved problems

Problem 135 *Which vocabulary does the below unary structure $\mathcal{M}$ have:*



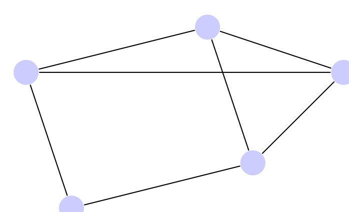
Solution: The vocabulary consists of one unary predicate symbol. By the way, it is not really possible to be absolutely sure what the vocabulary is, by merely looking at the picture. This could also be a picture of a structure with two unary predicates which are complements of each other. Or there could be many unary predicates, most of which are empty and therefore not seen in the picture. So the question given in this problem is a little vague. □

Problem 136 *Which vocabulary does the below tile model $\mathcal{M}$ have:*

R	*B*	*B*	*R*	*Y*

Solution: The vocabulary consists of one binary predicate symbol $<$ and three unary predicate symbols Y, R and B. This is indeed the typical vocabulary in a tile model. Of course, if there are fewer colors, the vocabulary can be taken to be smaller. □

Problem 137 *Which vocabulary does the following graph $\mathcal{M}$ have:*



Solution: As for all graphs, the vocabulary consists of one binary predicate symbol E. Many structures have a graph as part of the structure. In such cases the other structure determines what else there is in the vocabulary apart from the mere predicate symbol E. □

Problem 138 *Which vocabulary does the structure of the natural numbers with their natural order have:*

Solution: Although there are many different kinds of relations and functions on the natural structures, in this example we have taken only their natural order $n < m$. So the vocabulary consists of one binary predicate symbol $<$ only. □

Problem 139 *Is the following sentence true or false?*

> *If every rainy day in August is windy, and August 15th is not windy, then August 15th is not rainy.*

Solution: We recognize this as a true statement irrespective of which year we are talking about. It is even immaterial what "August", "rainy" and "windy" mean. The point is that every structure in which the predicates (rainy, windy) of the sentence are interpreted in any way, satisfies the sentence.
□

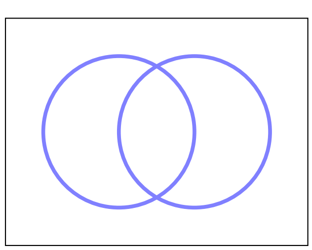
Problem 140 *Is the following sentence true or false?*

> *If every rainy day in August is windy, and August 15th is windy, then August 15th is rainy.*

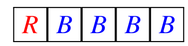
Solution: We recognize this as a statement that has "wrong logic". We can point out the logical error by describing a structure where the conclusion is false. An example of such a structure would be a hypothetical August where it rains only on August 1st but it is windy both on August 1st and August 15th.
□

2.3.7 Problems

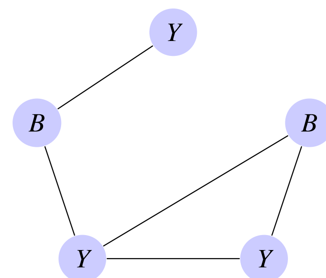
Problem 141 *Which vocabulary does the following unary structure $\mathcal{M}$ have:*



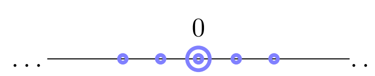
Problem 142 *Which vocabulary does the following tile model $\mathcal{M}$ have:*



Problem 143 *Which vocabulary does the following colored graph $\mathcal{M}$ have? The colors are indicated with letters: "B" means blue and "Y" means yellow.*



Problem 144 *Which vocabulary does the structure of the real numbers with their natural order and zero have:*



Problem 145 *True or false? Consider the sentence*

> *If every rainy day in August was windy, and every day in August was rainy, then every day in August was windy.*

Can you recognize this as a true statement irrespective of which year we are talking about and even irrespective of what "August", "rainy" and "windy" mean?

Problem 146 *True or false? Consider the sentence*

> *If every windy day in August was rainy, and August 15th was not windy, then August 15th was not rainy.*

Can you recognize this as a statement that has "wrong logic"? Can you point out the logical error by describing a structure where the assumption is true but the conclusion is false?

2.4 Atomic formulas

2.4.1 Introduction

Atomic formulas denote basic relations, such as $x = y$ and $x < c$, which can be true or false, depending on the values of variables and constants occurring in the formula. There is a resemblance to the atomic formulas of propositional logic. Only, this time the atomic formulas carry much more information because we have the variables. Somewhat vaguely one can say that formulas that do not contain connectives "and", "or", "not", "if...then", "if and only if" and also do not contain what we will later call *quantifiers*, "for all" and "exists", are atomic formulas. Atomic formulas may have internal structure that we cannot analyze with the means provided by predicate logic, as in the sentences "Tomorrow there will be a sea battle" and "It is possible that it will rain".

Here are some examples of atomic formulas of predicate logic:

- x is yellow
- x is taller than y
- $4 < z$
- $x = 10$
- xEy
- $x < z$
- $P_0(x)$

2.4.2 Variables

Variables are used in general statements, such as:

- If x and y are natural numbers then either $x < y$, $y < x$ or $x = y$.
- Any two distinct vertices are neighbors.
- Every tile is red or blue.

Variables are also used in existential statements, such as:

- There is a natural number x such that $x > 1010$ and x is a prime (has no divisors).
- Some vertices are not neighbors.
- Some tiles are left of a yellow tile.

Variables are denoted by x, y, z, u, v etc; also with indexes x_0,$x_1, \ldots$

2.4.3 Atomic formulas defined

Statements built up from variables using logical operations (connectives and quantifiers, see below) are called *formulas*. The simplest among them are the atomic formulas, which we will now define:

Definition 2.4 *Suppose L is a vocabulary. The* atomic formulas *of L consist of* equations

$$x = y, c = x, x = c, c = d,$$

where $c, d \in L$, and x, y can be any variables, and relational atomic formulas

$$P_n(x), P_n(c), R_n(x, y), R_n(c, d),$$

where $P_n, R_n, c, d \in L$ and x, y can be any variables.

2.4.4 Assignments

Assignments are functions that assign values to variables.

When we consider the truth of a formula with variables, we must have a structure in mind, otherwise the concept is meaningless. For example it does not make sense to ask whether the formula xEy is true or false, because we ought to know what the values of x and y are and, in addition, what the meaning of E is. Even if we know that E is the edge relation in a graph, we have to know which graph we are talking about, since there are lot and lots of different graphs.

So let us assume we have a structure $\mathcal{M}$ with universe M, such as

- a set of tiles
- the set of vertices of a graph
- integer numbers
- rational numbers.

We think that the variables range over the set M.

Fixing the value of a variable is called an *assignment*. Mathematically speaking, an assignment is a function that maps variables to M.

Here is a table of three assignments of values in the integers for the variables x, y and z.

- $s_0(x) = 1, s_0(y) = 5, s_0(z) = 1$
- $s_1(x) = 1, s_1(y) = 1, s_1(z) = 3$
- $s_2(x) = 1, s_2(y) = 2, s_2(z) = 4$

We can present such a table of values in the below compact form:

	x	y	z
s_0	1	5	1
s_1	1	1	3
s_2	1	2	4

2.4.5 Atomic formulas revisited

If an atomic formula contains no variables, such as

$$c = d, P_0(c), R_0(c, d),$$

its *truth* in a model $\mathcal{M}$ is easy to define:

Definition 2.5 *1.* $c = d$ *is* true *in* $\mathcal{M}$ *if* $c^{\mathcal{M}} = d^{\mathcal{M}}$.

2. $P_i(c)$ *is* true *in* $\mathcal{M}$ *if* $c^{\mathcal{M}} \in P_i^{\mathcal{M}}$.

3. $R_i(c, d)$ *is* true *in* $\mathcal{M}$ *if* $(c^{\mathcal{M}}, d^{\mathcal{M}}) \in R_i^{\mathcal{M}}$.

With variables we have to assignments. The assignment s_0 of the above table satisfies the atomic formula $x = z$, because $s_0(x) = s_0(z)$. The assignment s_1 of the above table satisfies the atomic formula $x = y$, because $s_1(x) = s_1(y)$.

Definition 2.6 *1. An assignment s satisfies the atomic formula $x = y$ if s gives the same value to x and y, i.e. $s(x) = s(y)$.*

2. An assignment s satisfies the atomic formula $x = c$ if s gives the value $c^{\mathcal{M}}$ to x, i.e. $s(x) = c^{\mathcal{M}}$.

3. An assignment s satisfies the atomic formula $c = y$ if s gives the value $c^{\mathcal{M}}$ to y, i.e. $s(y) = c^{\mathcal{M}}$.

4. An assignment s satisfies the atomic formula $P_n(x)$ in a model $\mathcal{M}$, if $s(x) \in P_n^{\mathcal{M}}$.

5. An assignment s satisfies the atomic formula $R(x, y)$ in a model $\mathcal{M}$, if $(s(x), s(y)) \in R^{\mathcal{M}}$.

6. An assignment s satisfies the atomic formula $R(x, c)$ in a model $\mathcal{M}$, if $(s(x), c^{\mathcal{M}}) \in R^{\mathcal{M}}$.

7. An assignment s satisfies the atomic formula $R(c, y)$ in a model $\mathcal{M}$, if $(c^{\mathcal{M}}, s(y)) \in R^{\mathcal{M}}$.

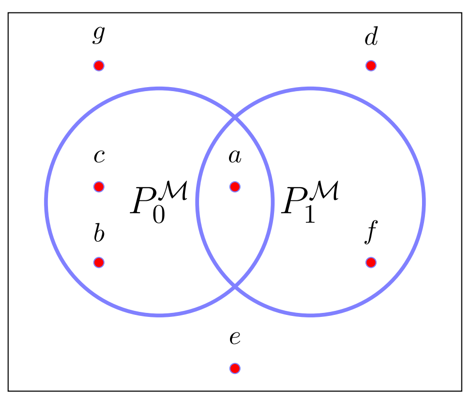
As a consequence of the definition, an assignment s satisfies the atomic formula xEy in a graph G, if $s(x)$ and $s(y)$ are neighbors in the graph. Likewise, an assignment s satisfies the atomic formula $R(x)$, "x is red", in a tile model, if $s(x)$ is red in the model.

Note: Instead of x and y we can have any other variables in the above definitions.

2.4.6 Solved problems

Problem 147 *Which assignments satisfy the atomic formula $P_0(x)$ in the below unary structure?*

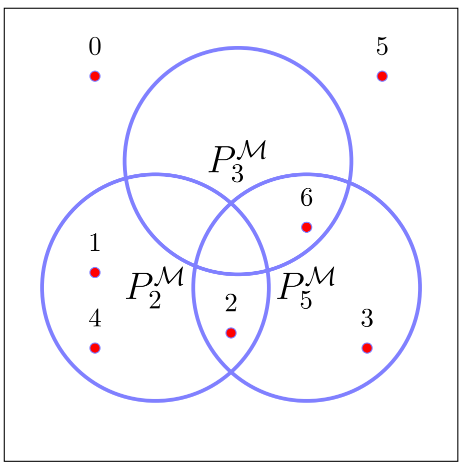
	x	y	z
s_0	b	a	d
s_1	c	b	c
s_2	g	a	g



Solution: Assignments s_0 and s_1 satisfy the atomic formula $P_0(x)$ since the elements $s_0(x) = b$ and $s_1(x) = c$ are in the set $P_0^{\mathcal{M}}$. Assignment s_2 does not satisfy the atomic formula $P_0(x)$ since $s_2(x) = g$ is not in the set $P_0^{\mathcal{M}}$. □

Problem 148 *Which assignments satisfy the atomic formula $P_2(z)$ in the below unary structure?*

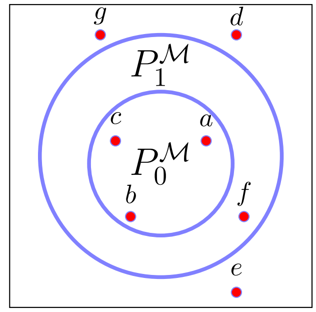
	x	y	z
s_0	1	2	5
s_1	1	6	0
s_2	1	4	6



Solution: None of the assignments satisfy the atomic formula $P_2(z)$ since no value of z is in the set $P_2^{\mathcal{M}}$. □

Problem 149 *Which assignments satisfy the formula $P_1(x)$ in the below unary structure?*

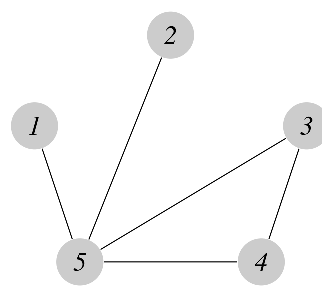
	x	y	z
s_0	b	a	d
s_1	c	b	c
s_2	g	a	g



Solution: The assignments s_0 and s_1 satisfy $P_1(x)$, because $s_0(x) = b \in P_1^{\mathcal{M}}$ and $s_1(x) = c \in P_1^{\mathcal{M}}$. However, s_2 does not satisfy, as $s_2(x) = g \notin P_1^{\mathcal{M}}$. □

Problem 150 *Which assignments satisfy the formula xEy in the below graph?*

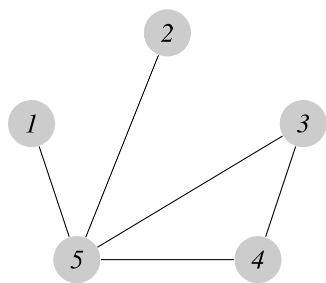
	x	y	z
s_0	4	5	1
s_1	1	1	3
s_2	1	2	4



Solution: The assignment s_0 satisfies xEy, because $s_0(x) = 4$, $s_0(y) = 5$, and indeed 4 and 5 are neighbors in the graph. The assignment s_1 does not satisfy xEy, because $s_1(x) = 1$, $s_1(y) = 1$, and in no graph is a vertex a neighbor of itself. The assignment s_2 does not satisfy xEy, because $s_2(x) = 1$, $s_2(y) = 2$, and in this graph the vertices 1 and 2 are not neighbors. □

Problem 151 *Which assignments satisfy the formula zEy in the below graph?*

	x	y	z
s_0	1	1	1
s_1	1	5	3
s_2	1	2	4



Solution: The assignment s_0 does not satisfy zEy, because $s_0(z) = s_1(y) = 1$ and in no graph is a vertex a neighbor of itself. The assignment s_1 satisfies zEy, because $s_1(z) = 3$, $s_1(y) = 5$, and in this graph the vertices 3 and 5 are neighbors. The assignment s_2 does not satisfy zEy, because $s_2(z) = 4$, $s_2(y) = 2$, and in this graph the vertices 4 and 2 are not neighbors. □

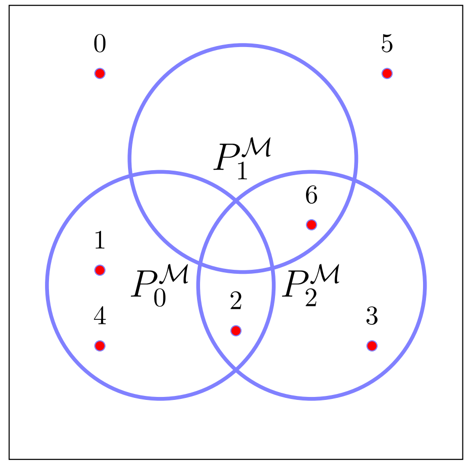
Problem 152 *Give in each case a unary structure and one assignment that satisfies the formula and one that does not:*

1. $P_0(x)$
2. $P_1(y)$
3. $P_1(z)$
4. $P_2(x)$

Solution:

	x	y	z	Satisfies
s_1	1	1	1	$P_0(x)$
s_2	6	6	6	$P_1(y)$
s_3	6	6	6	$P_1(z)$
s_4	3	3	3	$P_2(x)$

	x	y	z	Does not satisfy
s_1	6	6	6	$P_0(x)$
s_2	3	3	3	$P_1(y)$
s_3	2	2	2	$P_1(z)$
s_4	1	1	1	$P_2(x)$

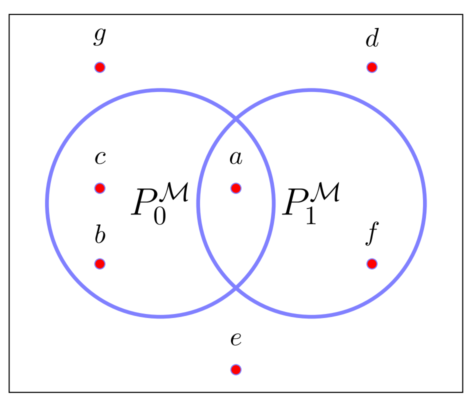


□

2.4.7 Problems

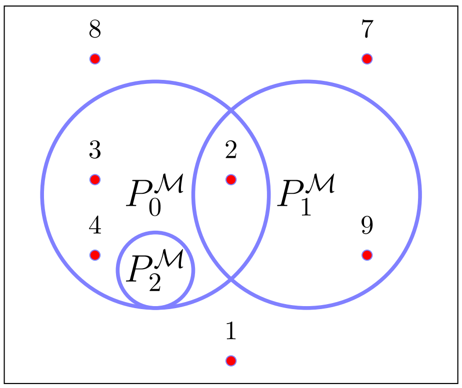
Problem 153 *Which assignments satisfy the atomic formula $P_1(y)$ in the below unary structure?*

	x	y	z
s_0	b	a	d
s_1	c	b	c
s_2	g	g	g



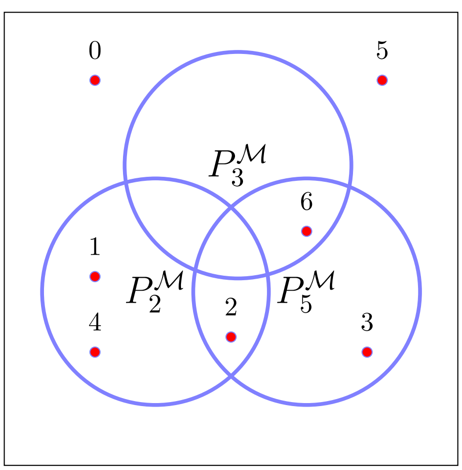
Problem 154 *Which assignments satisfy the atomic formula $P_0(z)$ in the below unary structure?*

	x	y	z
s_0	4	2	7
s_1	3	2	3
s_2	8	8	8



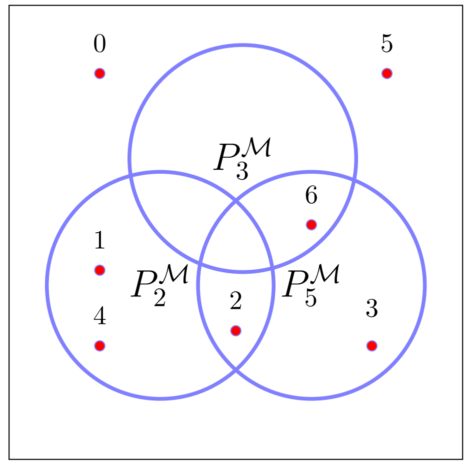
Problem 155 *Which assignments satisfy the atomic formula $P_5(y)$ in the below unary structure?*

	x	y	z
s_0	1	2	5
s_1	1	6	0
s_2	1	4	6



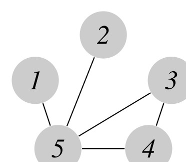
Problem 156 *Which assignments satisfy the atomic formula $P_2(x)$ in the below unary structure?*

	x	y	z
s_0	1	2	5
s_1	2	6	0
s_2	3	4	6



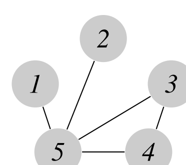
Problem 157 *Which assignments satisfy the formula xEy in the below graph?*

	x	y	z
s_0	4	1	1
s_1	3	5	3
s_2	2	2	4



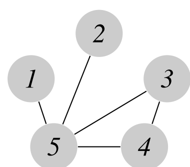
Problem 158 *Which assignments satisfy the formula xEz in the below graph?*

	x	y	z
s_0	1	2	1
s_1	1	2	3
s_2	1	2	4



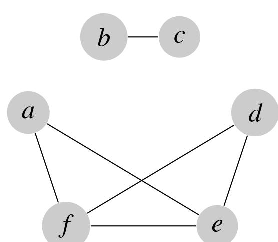
Problem 159 *Which assignments satisfy the formula* zEz *in the below graph?*

	x	y	z
s_0	1	1	1
s_1	1	5	3
s_2	1	2	4



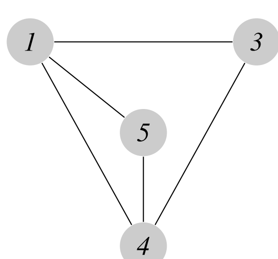
Problem 160 *Which assignments satisfy the formula* xEz *in the below graph?*

	x	y	z
s_0	a	b	c
s_1	a	d	d
s_2	f	c	e



Problem 161 *Which assignments satisfy the formula* xEy *in the below graph?*

	x	y	z
s_0	1	5	1
s_1	1	3	3
s_2	3	5	4



Problem 162 *Give in each case a tile model and one assignment that satisfies the formula, and another that does not:*

1. $R(x)$
2. $B(y)$
3. $y < x$
4. $x < y$
5. $Y(z)$

Problem 163 *Give in each case a tile model and an assignment that satisfies the first formula but not the second.*

1. $R(x), B(y)$
2. $B(y), x < y$
3. $y < x, R(y)$
4. $x < y, B(y)$
5. $z < x, z < y$

Problem 164 *Give in each case a unary structure and one assignment that satisfies the first formula but not the second:*

1. $P_0(x), P_1(x)$
2. $P_1(y), P_0(x)$
3. $P_1(y), P_0(y)$
4. $P_2(x), P_0(z)$

Problem 165 *Give a graph and an assignment that satisfies all the formulas in the left box but none of the formulas in the right box.*

xEy	yEu
$z = w$	$x = y$
yEz	zEu
xEu	wEu

2.5 Formulas

The formulas of predicate logic are expressions built up from atomic formulas by means of the familiar connectives of propositional logic but also by means of quantifiers, such as "for all" and "there exists", something propositional logic did not have. We can say more but at the same time the mathematics of predicate logic is more complicated than the mathematics of propositional logic. There are more things to keep an eye on.

2.5.1 Predicate logic formulas

Here is the exact definition of formulas of predicate logic:

Definition 2.7 *Predicate logic* formulas *are built up from atomic formulas by means of logical operations:*

Negation	$\neg A$	
Conjunction	$A \wedge B$	
Disjunction	$A \vee B$	
Implication	$A \to B$	
Equivalence	$A \leftrightarrow B$	
Existential quantifier	$\exists x A,$	*for any variable* x
Universal quantifier	$\forall x A,$	*for any variable* $x.$

Parentheses (,) are used for clarity.

For example, from the atomic formulas xEy and $x = z$ we can form the following formulas, among many others, of course:

- $xEy \wedge x = z$
- $xEy \wedge \neg x = z$
- $\exists x(xEy \wedge \neg x = z)$
- $\forall y \exists x(xEy \wedge \neg x = z)$
- $\exists x(xEy \wedge x = z) \wedge \exists x(xEy \wedge \neg x = z)$
- $\forall y(\exists x(xEy \wedge x = z) \wedge \exists x(xEy \wedge \neg x = z))$

All the formulas of predicate logic, like the ones above, have an intuitive meaning as soon as a structure is fixed so that we know what the domain is and what the symbols, such as "E" in xEy, mean. Moreover we have to decide what the assignment for giving values to the variables is. When these are known the meaning of the formulas is intuitively:

Formula	meaning
$\neg A$	not A
$A \wedge B$	A and B
$A \vee B$	A or B
$A \to B$	if A, then B
$A \leftrightarrow B$	A if and only if B
$\exists x A$	A holds for some value of x
$\forall x A$	A holds for all values of x.

The concept of an assignment is designed to make "value of x" in the above table completely exact and mathematical.

2.5.2 Disjunction, conjunction

Disjunction and conjunction are understood in predicate logic just as in propositional logic:

Definition 2.8 *An assignment s satisfies $A \vee B$ in $\mathcal{M}$ if and only if s satisfies A in $\mathcal{M}$ or s satisfies B in $\mathcal{M}$.*

In other words, to conclude that s satisfies $A \vee B$ in $\mathcal{M}$ it suffices to establish that s satisfies A or that s satisfies B in $\mathcal{M}$. Conversely, if we have established that s satisfies $A \vee B$, we know that s satisfies A or B in $\mathcal{M}$, but unfortunately we do not know which.

Definition 2.9 *An assignment s satisfies $A \wedge B$ in $\mathcal{M}$ if and only if s satisfies A in $\mathcal{M}$ and s satisfies B in $\mathcal{M}$.*

In other words, to conclude that s satisfies $A \wedge B$ in $\mathcal{M}$ it is necessary to establish both that s satisfies A in $\mathcal{M}$ and s satisfies B in $\mathcal{M}$. Conversely, if we have established that s satisfies $A \wedge B$ in $\mathcal{M}$, then we know that s satisfied A and also that s satisfies B in $\mathcal{M}$.

2.5.3 Negation

Negation—the denial—is treated as it is in propositional logic, nothing new:

Definition 2.10 *Assignment s satisfies $\neg A$ in $\mathcal{M}$ if and only if s does not satisfy A in $\mathcal{M}$.*

This is an easy case: To conclude that s satisfies $\neg A$ in $\mathcal{M}$ we just have to show that it is impossible that s would satisfy A in $\mathcal{M}$. Conversely, if we already know that s satisfies $\neg A$ in $\mathcal{M}$, then we know that surely s does not satisfy A in $\mathcal{M}$.

Equations	$x = y, c = x, x = c, c = d$
Relational	$P_n(x), P_n(c), R_n(x, y),$
	$R_n(c, x), R_n(x, c), R_n(c, d),$
Negation	$\neg A$
Conjunction	$A \wedge B$
Disjunction	$A \vee B$
Implication	$A \to B$
Equivalence	$A \leftrightarrow B$
Existential quantifier	$\exists xA$, for any variable x
Universal quantifier	$\forall xA$, for any variable x.

Figure 2.3: Formulas

2.5.4 Implication and equivalence

We follow here the treatment of implication in propositional logic: An implication is true exactly when the hypothesis is false or the conclusion is true. What we said earlier, in the section on propositional logic, about the unintuitive consequences of this, is of course still valid.

Definition 2.11 *An assignment s satisfies $A \to B$ in $\mathcal{M}$ if and only if s does not satisfy A in $\mathcal{M}$ or s satisfies B in $\mathcal{M}$.*

So to conclude that s satisfies $A \to B$ in $\mathcal{M}$ we assume s satisfies A in $\mathcal{M}$ and try to establish that s satisfies B in $\mathcal{M}$. Conversely, if we already know that s satisfies $A \to B$ in $\mathcal{M}$, then we have to consider two possibilities: In the first case s does not satisfy A in $\mathcal{M}$ and in the second case s satisfies B in $\mathcal{M}$. In a typical situation we know that s satisfies $A \to B$ in $\mathcal{M}$ and also that s satisfies A in $\mathcal{M}$, and then we can conclude that s satisfies B in $\mathcal{M}$.

Definition 2.12 *An assignment s satisfies $A \leftrightarrow B$ in $\mathcal{M}$ if and only if s satisfies both A and B in $\mathcal{M}$ or neither.*

Perhaps paradoxically, this is simpler than implication. To establish that s satisfies $A \leftrightarrow B$ in $\mathcal{M}$ one just assumes that s satisfies A in $\mathcal{M}$ and then tries to argue that s must satisfy also B in $\mathcal{M}$, and after this the same in the other direction: assuming that s satisfies B in $\mathcal{M}$ one tries to argue that s must satisfy A in $\mathcal{M}$. Conversely, if we know that s satisfies $A \leftrightarrow B$ in $\mathcal{M}$, then we know that if s satisfies one of A and B in $\mathcal{M}$, it satisfies also the other.

2.5.5 Satisfaction

We have defined when an assignment s satisfies a formula A without quantifiers in a structure $\mathcal{M}$. When this is the case, we write

$$M \models_s A.$$

With this notation we can rewrite the above definitions as follows:

Definition 2.13 *Satisfaction of A by s in $\mathcal{M}$, in symbols $M \models_s A$, is defined as follows in the case that A does not contain quantifiers:*

1. $\mathcal{M} \models_s x = y$ *if and only if* $s(x) = s(y)$.
2. $\mathcal{M} \models_s P_n(x)$ *if and only if* $s(x)$ *is in the set* $P_n^{\mathcal{M}}$.
3. $\mathcal{M} \models_s R(x, y)$ *if and only if* $(s(x), s(y))$ *is in the relation* $R^{\mathcal{M}}$, *i.e.* $s(x)R^{\mathcal{M}}s(y)$.
4. $\mathcal{M} \models_s A \vee B$ *if and only if* $(\mathcal{M} \models_s A$ *or* $\mathcal{M} \models_s B)$.
5. $\mathcal{M} \models_s A \wedge B$ *if and only if* $(\mathcal{M} \models_s A$ *and* $\mathcal{M} \models_s B)$.
6. $\mathcal{M} \models_s \neg A$ *if and only if* $\mathcal{M} \not\models_s A$.

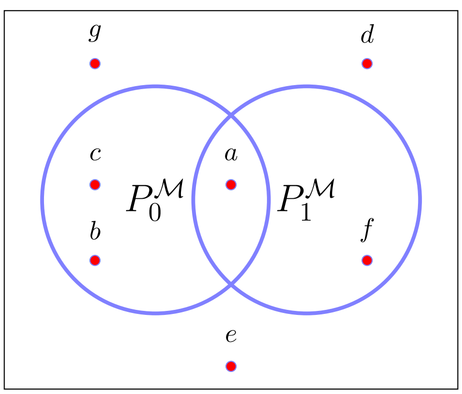
7. $\mathcal{M} \models_s A \to B$ *if and only if* ($\mathcal{M} \not\models_s A$ *or* $\mathcal{M} \models_s B$).
8. $\mathcal{M} \models_s A \leftrightarrow B$ *if and only if* [($\mathcal{M} \models_s A$ *and* $\mathcal{M} \models_s B$) *or* ($\mathcal{M} \not\models_s A$ *and* $\mathcal{M} \not\models_s B$)].

Note that Definition 2.13 is an inductive definition in the sense that $\mathcal{M} \models_s A$ is defined in terms of $\mathcal{M} \models_s B$ for subformulas B of A. So this is analogous to, albeit more complicated than, the definition of e.g. the Fibonacci sequence $a_0 = 0, a_1 = 1, a_{n+2} = a_n + a_{n+1}$, where a_{n+2} is defined in terms of the smaller numbers a_n and a_{n+1}.

2.5.6 Solved problems

Problem 166 *Which assignments satisfy the formula* $P_0(x) \vee P_1(y)$ *in the below unary structure?*

	x	y	z
s_0	b	a	d
s_1	c	b	c
s_2	g	g	g

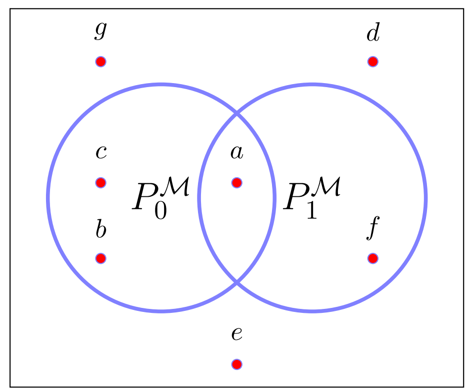


Solution: Even a casual look reveals that s_0 satisfies the formula, as s_0 gives x the value b which is in $P_0^{\mathcal{M}}$. Also s_1 satisfies the formula as it gives x the value c and also c is in $P_0^{\mathcal{M}}$. Finally, s_2 does not satisfy the formulas as it gives both x and y the value g, which is neither in $P_0^{\mathcal{M}}$ nor in $P_1^{\mathcal{M}}$.

More exactly, let us look at the assignments. The value of $s_0(x)$ is b, and $b \in P_0^{\mathcal{M}}$, so definitely $\mathcal{M} \models_{s_0} P_0(x) \vee P_1(y)$. The value of $s_1(x)$ is c, and $c \in P_0^{\mathcal{M}}$, so again $\mathcal{M} \models_{s_0} P_0(x) \vee P_1(y)$. Finally, $s_2(x) = g$ and $g \notin P_0^{\mathcal{M}}$, so we cannot conclude $\mathcal{M} \models_{s_0} P_0(x) \vee P_1(y)$ yet. We have to check also $s_2(y) = g$. But here likewise $g \notin P_1^{\mathcal{M}}$. So in the end, $\mathcal{M} \not\models_{s_0} P_0(x) \vee P_1(y)$. The answer is: s_0 and s_1 satisfy, but s_2 does not. □

Problem 167 *Which assignments satisfy the formula* $P_1(y) \to (P_0(x) \to P_1(z))$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	a	c
s_2	e	e	e

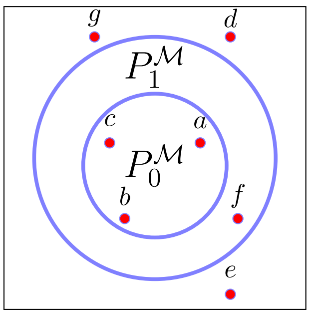


Solution: A quick look reveals that s_0 and s_2 satisfy the implication as both fail to satisfy the hypothesis of the implication, namely $P_1^{\mathcal{M}}$. But s_1 does satisfy the hypothesis $P_1^{\mathcal{M}}$ of the implication. However, it does not satisfy the conclusion, hence it fails to satisfy the implication itself.

More exactly, let us look at the assignments. The value of $s_0(y)$ is c, and $c \notin P_1^{\mathcal{M}}$. So $\mathcal{M} \not\models_{s_0} P_1(y)$ and hence s_0 satisfies the formula. The value of $s_1(y)$ is a, and $a \in P_1^{\mathcal{M}}$, so we go on. The value of $s_1(x)$ is a, and $a \in P_0^{\mathcal{M}}$, so we go on. The value of $s_1(z)$ is c, and $c \notin P_1^{\mathcal{M}}$. So $\mathcal{M} \not\models_{s_1} P_0(x) \to P_1(z)$ although $\mathcal{M} \models_{s_1} P_1(y)$. So the conclusion is that s_1 does not satisfy the formula. Finally, $s_2(y) = e$ and $e \notin P_1^{\mathcal{M}}$. So $\mathcal{M} \not\models_{s_2} P_1(y)$ and hence s_2 satisfies the formula. □

Problem 168 *Which assignments satisfy the formula* $P_1(y) \vee (P_0(z) \wedge P_1(x))$ *in the below unary structure?*

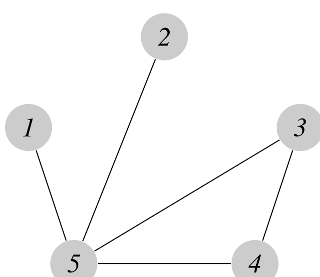
	x	y	z
s_0	c	c	c
s_1	a	e	c
s_2	f	e	f



Solution: Let us look at the assignments. The value of $s_0(y)$ is c, and $c \in P_1^{\mathcal{M}}$. So s_0 satisfies the formula. The value of $s_1(y)$ is e, and $e \notin P_1^{\mathcal{M}}$, so we go on. The value of $s_1(z)$ is c, and $c \in P_0^{\mathcal{M}}$, so we go on. The value of $s_1(x)$ is a, and $a \in P_1^{\mathcal{M}}$. So s_1 satisfies the formula. Finally, $s_2(y) = e$ and $e \notin P_1^{\mathcal{M}}$, so we go on. Now $s_2(z) = f$ and $f \notin P_0^{\mathcal{M}}$. So s_2 does not satisfy the formula. □

Problem 169 *Which assignments satisfy the formula $xEy \wedge yEz$ in the below graph?*

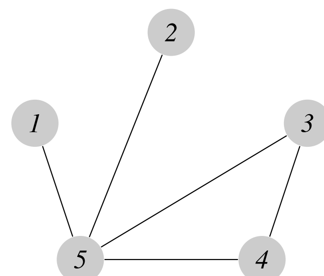
	x	y	z
s_0	1	5	1
s_1	1	1	3
s_2	1	2	4



Solution: Let us look at the assignments. The value of $s_0(x)$ is 1, and $s_0(y) = 5$. We have $1E^{\mathcal{M}}5$, so s_0 satisfies xEy. The value of $s_0(z)$ is 1, so s_0 satisfies also yEz, and hence the given formula. Now $s_1(x) = 1$ and $s_1(y) = 1$, but $(1,1) \notin E^{\mathcal{M}}$. So s_1 does not satisfy the formula. Finally, $s_2(x) = 1$ and $s_2(y) = 2$, but $(1,2) \notin E^{\mathcal{M}}$. So s_2 does not satisfy the formula. □

Problem 170 *Which assignments satisfy the formula $xEy \to xEz$ in the below graph?*

	x	y	z
s_0	1	5	2
s_1	1	1	3
s_2	1	2	4



Solution: The assignment s_0 satisfies xEy but not xEz. So s_0 does not satisfy $xEy \to xEz$. The assignment s_1 fails to satisfy xEy, and hence it satisfies the implication. The same happens with s_2.□

Problem 171 *Which assignments satisfy the formula $R(x) \vee B(y)$ in the below tile model? In order to be able to refer to the individual tiles we give them names.*

	x	y	z
s_0	1	5	2
s_1	2	1	3
s_2	2	2	4

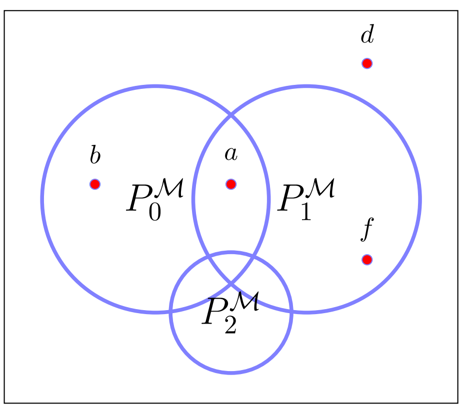
R	*B*	*B*	*R*	*Y*
1	*2*	*3*	*4*	*5*

Solution: Intuitively, the question is "Is x red or y blue?". So let us see what the assignments say about x and y. The assignment s_0 says x has value 1, i.e. $s_0(x) = 1$. Well, 1 is red, so s_0 does satisfy the given formula. The assignment s_1 says x has value 2, which is not red, and y has value 1, which is not blue. So s_1 does not satisfy the formula. Finally, s_2 says x has value 2, which is not red, but y has also value 2, and 2 *is* blue. So s_2 satisfies the formula. □

Problem 172 *Give in each case a unary structure and an assignment that satisfies the given formula in the structure:*

1. $P_0(x) \wedge \neg P_1(x)$
2. $\neg(P_0(x) \vee \neg P_1(x))$
3. $P_0(x) \to (P_1(x) \vee P_2(x))$
4. $P_0(x) \vee \neg P_1(x) \vee P_2(x)$

Solution: We use the same model in each case:

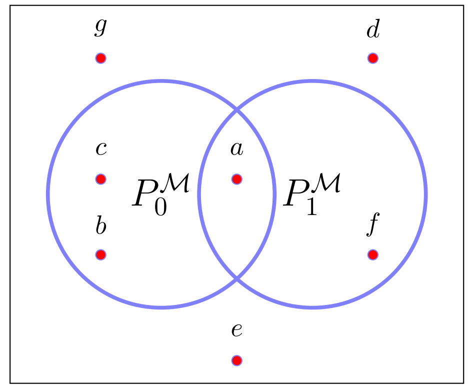


In Case 1 it suffices to let $s(x) = b$. In Case 2 we let $s(x) = f$. In Case 3, let $s(x) = f$. Finally, in Case 4, let $s(x) = b$.
□

2.5.7 Problems

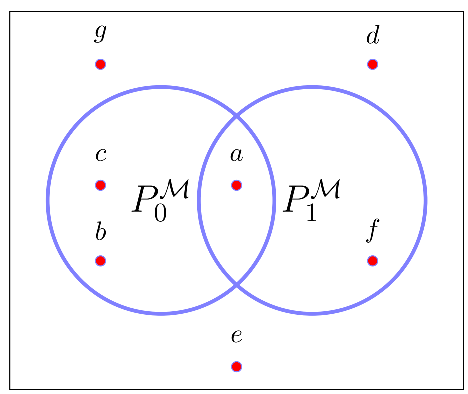
Problem 173 *Which assignments satisfy the formula* $\neg P_0(x) \wedge P_1(y)$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	b	c
s_2	e	f	a



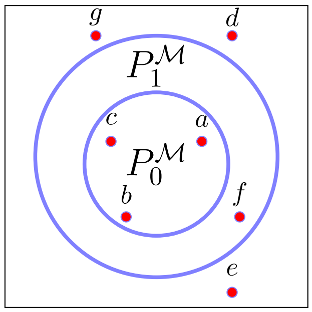
Problem 174 *Which assignments satisfy the formula* $P_1(z) \to (\neg P_0(x) \to P_1(x))$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	a	f
s_2	e	e	a



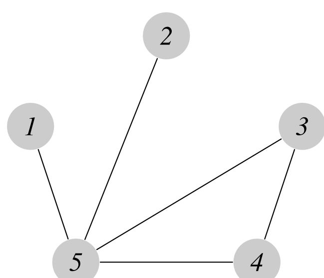
Problem 175 *Which assignments satisfy the formula* $P_1(z) \to (P_0(y) \to P_1(x))$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	g	a	c
s_2	e	e	a



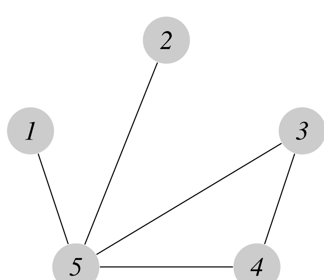
Problem 176 *Which assignments satisfy the formula $xEy \to yEz$ in the below graph?*

	x	y	z
s_0	1	5	1
s_1	1	1	3
s_2	3	4	2



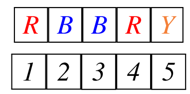
Problem 177 *Which assignments satisfy the formula $xEy \vee xEz$ in the below graph?*

	x	y	z
s_0	1	5	1
s_1	1	2	3
s_2	5	2	4



Problem 178 *Which assignments satisfy the formula $R(x) \to x < y$ in the below tile model? In order to be able to refer to the individual tiles we have given them names $1, 2, 3, 4, 5$.*

	x	y	z
s_0	1	5	2
s_1	2	1	3
s_2	4	2	4



Problem 179 *Give in each case a tile model and an assignment that satisfies the given formula in the model:*

1. $R(x) \wedge \neg B(x)$
2. $\neg(R(x) \wedge \neg B(x))$
3. $B(x) \to (R(y) \wedge y < x)$
4. $\neg(B(x) \to (R(y) \wedge y < x))$
5. $\neg((B(x) \wedge Y(y) \wedge x < y) \vee (B(x) \wedge Y(y) \wedge y < x))$

2.6 Quantifiers

2.6.1 Introduction

Quantifiers are the final step in our description of the building blocks that first order (i.e. predicate logic) formulas are built up from. With quantifiers we can finally describe a wealth of phenomena around us, inside computers, and in mathematics. A famous example of the use of quantifiers is the $\epsilon - \delta$ definition of continuity[2].

2.6.2 Predicate logic (i.e. first order) formulas

Let us recall that predicate logic formulas are of the form

- atomic i.e. $x = y$, $P_n(x)$, $R(x, y)$

[2]A function $f : \mathbb{R} \to \mathbb{R}$ is *continuous* at x_0 if for every $\epsilon > 0$ there is $\delta > 0$ such that for all x, if $|x - x_0| < \delta$, then $|f(x) - f(x_0)| < \epsilon$.

- $\neg A$
- $A \wedge B$
- $A \vee B$
- $A \rightarrow B$
- $A \leftrightarrow B$
- $\forall x A$
- $\exists x A$,

where A and B are first order formulas. Parentheses (,) are used for clarity, just as in propositional logic. Thus, for example, if we want to put a universal quantifier in front of $A \rightarrow B$ we write $\forall x(A \rightarrow B)$.

2.6.3 Examples

- $P_0(x) \rightarrow P_1(x)$
- $\neg(x < y \vee y < x)$
- $\exists x(xEy \wedge \exists z(xEz \wedge \neg zEy))$
- $\forall x(B(x) \rightarrow \exists z(Y(z) \wedge z < x))$

2.6.4 Universal quantifier explained

The intuitive meaning of $\forall x A$ is that *every* value of x satisfies A, as in

- Every tile is red.
- Every x satisfies $x^2 \geq 0$.
- All vertices x and y are neighbors.
- All men are mortal.
- Everybody loves her.

2.6.5 Existential quantifier explained

The intuitive meaning of $\exists x A$ is that *some* value of x satisfies A as in

- Some tiles are red.
- Some reals x satisfy $x^2 = 2$.
- Some vertices x and y are neighbors.
- There is a yellow tile.
- There is a vertex with two neighbors.

2.6.6 Assignments and quantifiers

In order to *define* when an assignment satisfies a quantified formula, we need the concept of a *modified assignment*. From any assignment s, and variable x and any element a of the universe of our structure we can form a *modified* assignment $s(a/x)$ as follows:

	x	y	z	
s	1	5	1	
$s(2/x)$	2	5	1	A modified assignment.
$s(8/z)$	1	5	8	Another modified assignment.

The assignment $s(a/x)$ is exactly like the assignment s except that the value of x is changed to a.

2.6.7 Assignment satisfying a quantified formula

We can now extend the previous definition of satisfaction from formulas without quantifiers to formulas that also contain quantifiers:

Definition 2.14

- *Assignment s satisfies $\forall x A$ in $\mathcal{M}$ if the modified assignment $s(a/x)$ satisfies A in $\mathcal{M}$ for every a in M.*
- *Assignment s satisfies $\exists x A$ in $\mathcal{M}$ if the modified assignment $s(a/x)$ satisfies A in $\mathcal{M}$ for some a in M.*

2.6.8 Tarski Truth Definition

We have now defined in full generality, when an assignment s satisfies a formula A in a structure $\mathcal{M}$. When this is the case, we write $\mathcal{M} \models_s A$.

With this notation we can rewrite the above definition as follows:

Definition 2.15 *Satisfaction of A by s in $\mathcal{M}$, in symbols $M \models_s A$, is defined as in Table 2.4.*

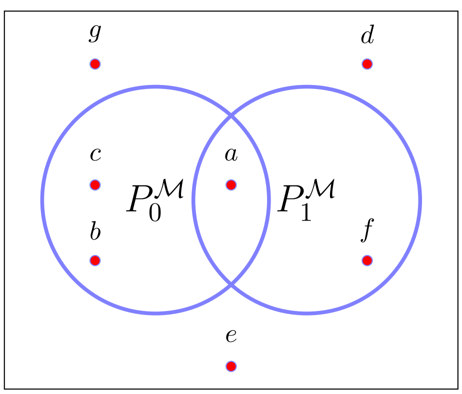
Note that Table 2.4 is an inductive definition in the sense that $\mathcal{M} \models_s A$ is defined in terms of $\mathcal{M} \models_{s'} B$ for subformulas B of A and modification s' of s.

This is called the Tarski Truth Definition.

2.6.9 Solved problems

Problem 180 *Which assignments satisfy the formula $\exists y(P_0(x) \wedge P_1(y))$ in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	b	c
s_2	e	e	a



Solution: Let us first show that s_0 satisfies the given formula in $\mathcal{M}$. The modified assignment $s_0(f/y)$ satisfies $P_1(y)$ in this model as $f \in P_1^{\mathcal{M}}$. The assignment $s_0(f/y)$ satisfies also $P_0(x)$ in this model, as $s_0(f/y)(x) = c \in P_0^{\mathcal{M}}$. So $s_0(f/y)$ satisfies $P_0(x) \wedge P_1(y)$ in $\mathcal{M}$. Hence s_0 satisfies $\exists y(P_0(x) \wedge P_1(y))$ in $\mathcal{M}$.

Next we show that s_1 also satisfies the given formula in $\mathcal{M}$. We already observed that $s_1(f/y)$ satisfies $P_1(y)$ in $\mathcal{M}$. The assignment $s_1(f/y)$ satisfies also $P_0(x)$ in this model, as $s_1(f/y)(x) = a \in P_0^{\mathcal{M}}$. Hence s_1 satisfies $\exists y(P_0(x) \wedge P_1(y))$ in $\mathcal{M}$.

Let us finally show that s_2 does not satisfy the given formula in $\mathcal{M}$. Whatever element h we pick, the modified assignment $s_2(h/y)$ fails to satisfy $P_0(x)$ in this model, as $s_2(h/y)(x) = e \notin P_0^{\mathcal{M}}$. So $s_2(h/y)$ fails to satisfy $P_0(x) \wedge P_1(y)$ in $\mathcal{M}$, whatever h is. Hence s_2 does not satisfy $\exists y(P_0(x) \wedge P_1(y))$ in $\mathcal{M}$.

So the answer is, that the first two assignments satisfy the formula in $\mathcal{M}$ and the last does not.

We can write the whole solution also using the notation $\mathcal{M} \models_s A$:

Let us first show that

$$\mathcal{M} \models_{s_0} \exists y(P_0(x) \wedge P_1(y)).$$

For this, note that

$$\mathcal{M} \models_{s_0(f/y)} P_1(y)$$

as $f \in P_1^{\mathcal{M}}$. Note also that

$$\mathcal{M} \models_{s_0(f/y)} P_0(x),$$

as $s_0(f/y)(x) = c \in P_0^{\mathcal{M}}$. So

$$\mathcal{M} \models_{s_0(f/y)} P_0(x) \wedge P_1(y).$$

Hence

$$\mathcal{M} \models_{s_0} \exists y(P_0(x) \wedge P_1(y)).$$

Next we show that also

$$\mathcal{M} \models_{s_1} \exists y(P_0(x) \wedge P_1(y)).$$

We already observed that

$$\mathcal{M} \models_{s_1(f/y)} P_1(y).$$

Note that also

$$\mathcal{M} \models_{s_1(f/y)} P_0(x),$$

as $s_1(f/y)(x) = a \in P_0^{\mathcal{M}}$. Hence

$$\mathcal{M} \models_{s_1} \exists y(P_0(x) \wedge P_1(y)).$$

Equations	$\mathcal{M} \models_s x = y$	if and only if	$s(x) = s(y)$
	$\mathcal{M} \models_s c = y$	if and only if	$c^{\mathcal{M}} = s(y)$, similarly $x = d$
	$\mathcal{M} \models_s c = d$	if and only if	$c^{\mathcal{M}} = d^{\mathcal{M}}$
Relations	$\mathcal{M} \models_s P_n(x)$	if and only if	$s(x) \in P_n^{\mathcal{M}}$
	$\mathcal{M} \models_s P_n(c)$	if and only if	$c^{\mathcal{M}} \in P_n^{\mathcal{M}}$
	$\mathcal{M} \models_s R(x, y)$	if and only if	$(s(x), s(y)) \in R^{\mathcal{M}}$.
	$\mathcal{M} \models_s R(c, y)$	if and only if	$(c^{\mathcal{M}}, s(y)) \in R^{\mathcal{M}}$, similarly $R(x, d)$.
	$\mathcal{M} \models_s R(c, d)$	if and only if	$(c^{\mathcal{M}}, d^{\mathcal{M}}) \in R^{\mathcal{M}}$.
Connectives	$\mathcal{M} \models_s A \vee B$	if and only if	$\mathcal{M} \models_s A$ or $\mathcal{M} \models_s B$.
	$\mathcal{M} \models_s A \wedge B$	if and only if	$\mathcal{M} \models_s A$ and $\mathcal{M} \models_s B$.
	$\mathcal{M} \models_s \neg A$	if and only if	$\mathcal{M} \not\models_s A$.
	$\mathcal{M} \models_s A \to B$	if and only if	$\mathcal{M} \not\models_s A$ or $\mathcal{M} \models_s B$.
	$\mathcal{M} \models_s A \leftrightarrow B$	if and only if	$(\mathcal{M} \models_s A$ and $\mathcal{M} \models_s B)$ or
			$(\mathcal{M} \not\models_s A$ and $\mathcal{M} \not\models_s B)$
Quantifiers	$\mathcal{M} \models_s \forall x A$	if and only if	$\mathcal{M} \models_{s(a/x)} A$ for all $a \in M$
	$\mathcal{M} \models_s \exists x A$	if and only if	$\mathcal{M} \models_{s(a/x)} A$ for some $a \in M$

Figure 2.4: The Tarski Truth Definition

Let us finally show that

$$\mathcal{M} \models_{s_2} \exists y(P_0(x) \wedge P_1(y)).$$

Whatever element h we pick,

$$\mathcal{M} \not\models_{s_2(h/y)} P_0(x),$$

as $s_2(h/y)(x) = e \notin P_0^{\mathcal{M}}$. So

$$\mathcal{M} \not\models_{s_2(h/y)} P_0(x) \wedge P_1(y),$$

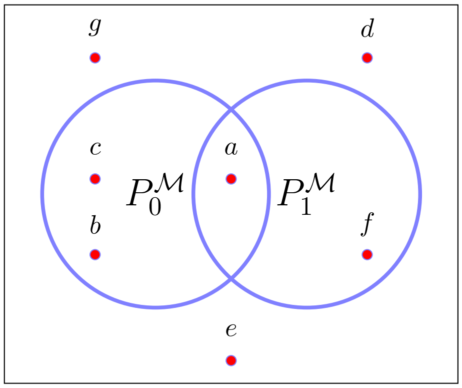
whatever h is. Hence

$$\mathcal{M} \not\models_{s_2} \exists y(P_0(x) \wedge P_1(y)).$$

So the answer is, that the first two assignments satisfy the formula in $\mathcal{M}$ and the last does not. □

Problem 181 *Which assignments satisfy the formula* $P_1(y) \to \forall x(P_0(x) \to P_1(x))$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	a	c
s_2	e	e	a



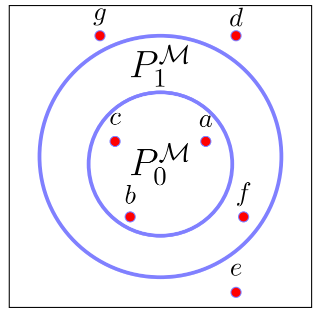
Solution: The formula we are considering is an implication. Implication if true if the hypothesis is false. The value of y in s_0 and s_2 is outside the set $P_1^{\mathcal{M}}$. So both s_0 and s_2 fail to satisfy $P_1(y)$ in $\mathcal{M}$. Hence they both satisfy the given formula. We are left with s_1. In this case $s_1(y) = a \in P_1^{\mathcal{M}}$, so the hypothesis of the implication is true. We have to next consider the conclusion. The conclusion says intuitively, that $P_0^{\mathcal{M}}$ is contained in $P_1^{\mathcal{M}}$, which is clearly false. Let us see why it is clearly false. Well, because e.g. c is in $P_0^{\mathcal{M}}$ but not in $P_1^{\mathcal{M}}$. So $s_1(c/x)$ fails to satisfy $P_0(x) \to P_1(x)$. Hence s_1 fails to satisfy $\forall x(P_0(x) \to P_1(x))$. Hence s_1 does not satisfy the given formula in $\mathcal{M}$.

So the answer is, that the first and the last assignment satisfies the formula in $\mathcal{M}$ and the middle one does not.

□

Problem 182 *Which assignments satisfy the formula $P_1(y) \to \forall x(P_0(z) \to P_1(x))$ in the below unary structure?*

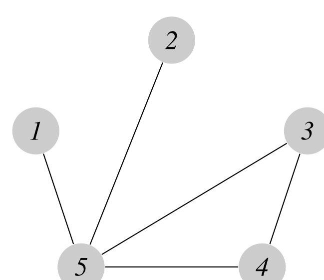
	x	y	z
s_0	c	c	f
s_1	a	a	c
s_2	e	e	a



Solution: We can immediately observe that s_2 satisfies the implication because it does not satisfy the hypothesis: $s_2(y) = e \notin P_1^{\mathcal{M}}$. Now the conclusion of the implication does *not* say that $P_0^{\mathcal{M}} \subseteq P_1^{\mathcal{M}}$, but something quite different, namely that if the z is in P_0, then every element is in P_1. In the assignment s_0 we have $s_0(z) = f \notin P_0$. So whatever h is chosen, $s_0(h/x)$ fails to satisfy $P_0(z)$, whence it *does* satisfy $P_0(z) \to P_1(x)$. So s_0 satisfies the conclusion of the given implication, and hence it satisfies the given formula. Let us finally look at s_1. Let us look at $s_1(g/x)$. Since $s_1(g/x)(z) = c \in P_0^{\mathcal{M}}$, $s_1(g/x)$ satisfies the hypothesis of the implication $P_0(z) \to P_1(x)$. But it does not satisfy the conclusion of the implication, hence it does not satisfy the implication itself. Hence s_1 does not satisfy $\forall x(P_0(z) \to P_1(x))$. Since $s_1(y) = a \in P_0^{\mathcal{M}}$, s_1 does not satisfy the given formula in $\mathcal{M}$. □

Problem 183 *Which assignment satisfies the formula $\exists y(xEy \wedge yEz)$ in the below graph?*

	x	y	z
s_0	1	5	1
s_1	1	1	3
s_2	1	2	4



Solution: This one is quite easy. In each case the value of x is 1. The formula claims that some neighbor of 1 is a neighbor of the value of z. We can choose 5, the only neighbor of 1, in each case. So all three assignments satisfy the formula. Let us do this a bit more exactly, using the $\mathcal{M} \models_s A$ notation. First s_0. We have

$$\mathcal{M} \models_{s_0(5/y)} xEy,$$

because there is an edge between $s_0(x) = 1$ and 5. Moreover,

$$\mathcal{M} \models_{s_0(5/y)} yEz,$$

because $s_0(z)$ is also 1. So

$$\mathcal{M} \models_{s_0(5/y)} xEy \wedge yEz,$$

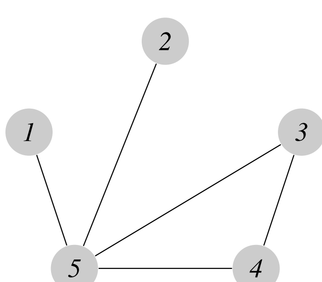
whence

$$\mathcal{M} \models_{s_0} \exists y(xEy \wedge yEz).$$

Then s_1. Now we use the prose approach: The assignment $s_1(5/y)$ satisfies xEy, because there is an edge between $s_1(x) = 1$ and 5. Moreover, $s_1(5/y)$ satisfies also yEz, because there is an edge between $s_1(z) = 3$ and 5. So, summa summa summarum, $s_1(5/y)$ satisfies $xEy \wedge yEz$, and therefore s_1 satisfies $\exists y(xEy \wedge yEz)$. Finally the assignment s_2. The argument is exactly the same, as there is an edge between $s_2(z) = 4$ and 5. □

Problem 184 *Which assignment satisfies the formula* $\forall x(x = y \vee xEy)$ *in the below graph?*

	x	y	z
s_0	1	5	1
s_1	1	1	3
s_2	1	2	4



Solution: The formula says, intuitively, that the vertex that is the interpretation of y, is a neighbor of every other vertex. In this graph only vertex 5 is a neighbor of every other vertex. This means that only s_0 satisfies the formula. More exactly, $s_0(a/x)$ satisfies $x = y \vee xEy$ for every vertex a, and therefore s_0 satisfies the formula. On the other hand, $s_1(2/x)$ does not satisfy xEy, and also $s_2(3/x)$ does not satisfy xEy. In conclusion, s_1 and s_2 do not satisfy the given formula. □

Problem 185 *Which assignment satisfies the formula* $\exists y(R(x) \wedge x < y)$ *in the below tile model?*

	x	y	z
s_0	1	5	2
s_1	2	1	3
s_2	4	2	4

R	*B*	*B*	*R*	*Y*
1	*2*	*3*	*4*	*5*

Solution: The formula says, intuitively, that the tile that is the interpretation of x, is red and it has a tile right of it. In this tile model tiles number 1 and 4 both are red and both have a tile right of it. This means that s_0 and s_2 satisfy the formula. More exactly, $s_0(2/y)$ satisfies $R(x) \wedge xEy$, and therefore s_0 satisfies the formula. Likewise, $s_2(5/y)$ satisfies $R(x) \wedge xEy$, so s_2 satisfies the given formula. On the other hand, $s_1(a/y)$ does not satisfy $R(x)$ for any a, so s_1 cannot satisfy the given formula. In conclusion, s_0 and s_2 satisfy the given formula, s_1 does not. □

Problem 186 *Which assignment satisfies the formula* $\forall y(B(y) \rightarrow y < x)$ *in the below tile model?*

	x	y	z
s_0	1	5	2
s_1	5	1	3
s_2	4	2	4

R	*B*	*B*	*R*	*Y*
1	*2*	*3*	*4*	*5*

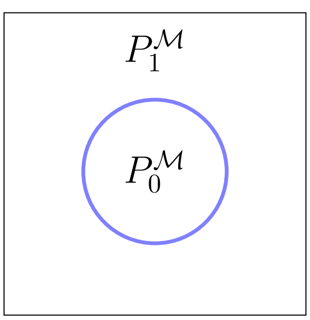
Solution: The formula says, intuitively, that all the blue vertices are left of the vertex that is the interpretation of x. In this tile model all the blue vertices are left of only the vertices number 4 and 5. This means that s_1 and s_2 satisfy the formula, but s_0 not. More exactly, $s_0(2/y)$ satisfies $B(x)$ but not $y < x$, and therefore s_0 does not satisfy the given formula. On the other hand, if $s_1(a/y)$ satisfies $B(y)$, then $a = 2$ or $a = 3$ so then $s_1(a/y)$ also satisfies $a < x$. Hence s_1 satisfies the given formula. The same applies to s_2. In conclusion, s_1 and s_2 satisfy the given formula, s_0 does not. □

Problem 187 *Give in each case a unary structure and an assignment that satisfies the sentence:*

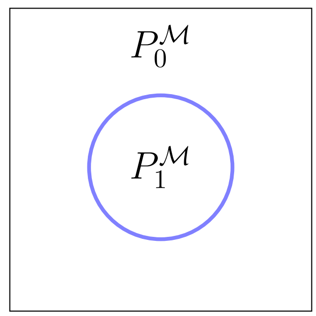
1. $\exists x P_0(x) \wedge \forall x \neg P_1(x)$
2. $\neg(\exists x P_0(x) \vee \forall x \neg P_1(x))$
3. $\forall x(P_0(x) \to (P_1(x) \vee P_2(x)))$

Solution:

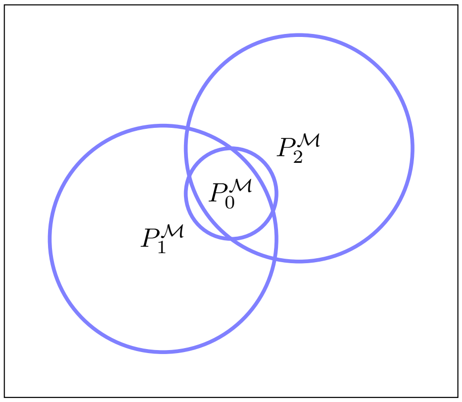
1. $\exists x P_0(x) \wedge \forall x \neg P_1(x)$: The sentence says, intuitively, that P_0 is non-empty, but P_1 is empty. We let, for example $M = \{0,1,2,3,4,5\}, P_0^{\mathcal{M}} = 3,4,5$, $P_1^{\mathcal{M}} = \emptyset$, $s(x) = 0$. Now s satisfies $\exists x P_0(x)$ because $s(3/x)$ satisfies $P_0(x)$ i.e. $3 \in P_0^{\mathcal{M}}$. Also, s satisfies $\forall x \neg P_1(x)$ because, for all $a \in M$, $s(a/x)$ satisfies $\neg P_1(x)$ i.e. $a \notin P_1^{\mathcal{M}}$. Hence s satisfies $\exists x P_0(x) \wedge \forall x \neg P_1(x)$.



2. $\neg(\exists x P_0(x) \vee \forall x \neg P_1(x))$: Intuitively, this sentence says that it is not the case that P_0 is non-empty or that P_1 is empty. Our model should therefore have both P_0 empty and P_1 non-empty. Let, for example, $M = \{0,1,2,3,4,5\}$, $P_0^{\mathcal{M}} = \emptyset$, $P_1^{\mathcal{M}} = \{3,4,5\}$, $s(x) = 0$. Now s does not satisfy $\forall x \neg P_1(x)$ because $s(3/x)$ satisfies $P_1(x)$ i.e. $3 \in P_1^{\mathcal{M}}$. Also, s does not satisfy $\exists x P_0(x)$ because, for all $a \in M$, $s(a/x)$ fails to satisfy $P_0(x)$ i.e. $a \notin P_0^{\mathcal{M}}$. Hence s does not satisfy $\exists x P_0(x) \vee \forall x \neg P_1(x)$. Hence s satisfies $\neg(\exists x P_0(x) \vee \forall x \neg P_1(x))$.



3. $\forall x(P_0(x) \to (P_1(x) \vee P_2(x)))$ Intuitively, this sentence says, that all elements of P_0 are either in P_1 or in P_2. Let, for example, $M = \{0,1,2,3,4,5\}$, $P_0^{\mathcal{M}} = \{3,4\}$, $P_1^{\mathcal{M}} = \{1,3\}$, $P_2^{\mathcal{M}} = \{4,5\}$, and $s(x) = 0$. Let us take an arbitrary a in M and show that $s(a/x)$ satisfies $P_0(x) \to (P_1(x) \vee P_2(x))$. Assume therefore that $s(a/x)$ satisfies $P_0(x)$ i.e. that a is in $P_0^{\mathcal{M}}$. Then $a = 3$ or $a = 4$. In the first case a is in $P_1^{\mathcal{M}}$. In the second case a is in $P_2^{\mathcal{M}}$. In either case $s(a/x)$ satisfies $P_1(x) \vee P_2(x)$. We have shown that $s(a/x)$ satisfies $P_0(x) \to (P_1(x) \vee P_2(x))$ for all a in M. Hence s satisfies $\forall x(P_0(x) \to (P_1(x) \vee P_2(x)))$.

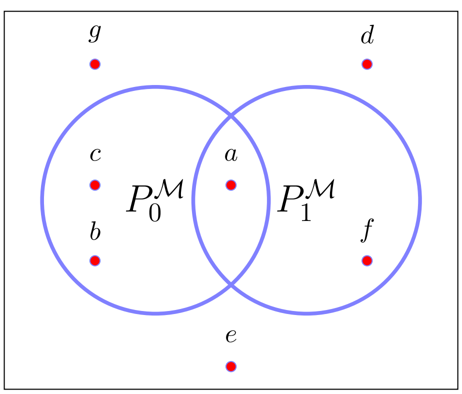


□

2.6.10 Problems

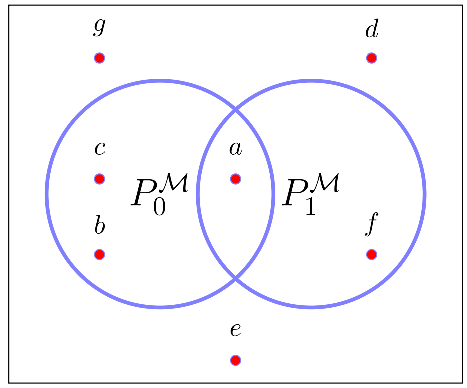
Problem 188 *Which assignments satisfy the formula* $\forall y(P_0(x) \to P_1(y))$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	b	c
s_2	e	e	a



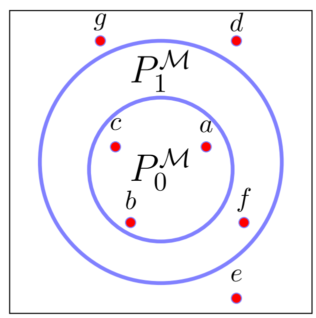
Problem 189 *Which assignments satisfy the formula* $P_1(z) \vee \forall x(P_0(x) \vee P_1(x))$ *in the below unary structure?*

	x	y	z
s_0	c	c	a
s_1	a	a	e
s_2	e	e	b



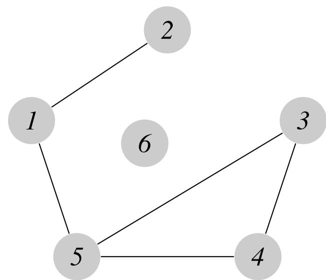
Problem 190 *Which assignments satisfy the formula* $P_1(y) \wedge \forall x(P_0(x) \to P_1(x))$ *in the below unary structure?*

	x	y	z
s_0	c	c	c
s_1	a	a	c
s_2	e	e	a



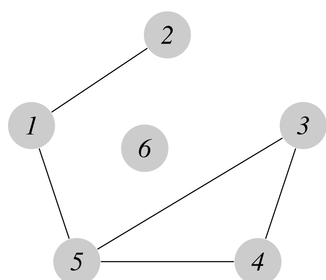
Problem 191 *Which assignment satisfies the formula* $\exists y(xEy \wedge yEz)$ *in the below graph?*

	x	y	z
s_0	2	6	5
s_1	1	1	3
s_2	1	6	2



Problem 192 *Which assignment satisfies the formula* $\exists x \exists y(\neg x = y \wedge xEy \wedge yEz)$ *in the below graph?*

	x	y	z
s_0	1	5	1
s_1	1	1	2
s_2	1	2	4



Problem 193 *Give a unary structure* $\mathcal{M}$ *and an assignment that satisfies the sentence*

$$\forall x(P_0(x) \vee \neg P_1(x) \vee P_2(x))$$

but not the sentence

$$\exists x(P_0(x) \wedge \neg P_1(x) \wedge P_2(x))$$

in $\mathcal{M}$.

Problem 194 *Give in each case a tile model that satisfies the sentence*

1. $\exists x R(x) \wedge \forall x \neg B(x)$
2. $\neg(\exists x R(x) \wedge \forall x \neg B(x))$
3. $\forall x(B(x) \rightarrow \exists y(R(y) \wedge y < x))$

Problem 195 *Give in each case a tile model that satisfies the sentence*

1. $\neg \forall x(B(x) \rightarrow \exists y(R(y) \wedge y < x))$
2. $\neg(\exists x \exists y(B(x) \wedge Y(y) \wedge x < y) \vee \exists x \exists y(B(x) \wedge Y(y) \wedge y < x))$

2.7 Validity

2.7.1 Introduction

Now, having learnt the basic concepts of predicate logic, we can enter the heartland of logic. We can use our concepts to analyze why certain inferences seem correct and others don't.

Why do we strongly believe that "Some days are rainy" follows from "Some days are rainy and windy", and even with no regard to what "day", "rainy" and "windy" mean? And why do we strongly believe that "Some days are rainy" does *not* follow from "Some days are rainy or windy"? What we are talking about here is the question of logical consequence and validity.

A first order formula of vocabulary L is *valid* if it is satisfied by every assignment in every structure for L. A valid formula expresses a general logical truth, something which is always true whatever is the meaning of the predicate and constant symbols of the formula.

2.7.2 Examples

Here are some examples of valid formulas of predicate logic. In each case the proof of the validity is easy. One takes an arbitrary model and an arbitrary assignment and shows that the assignment satisfies the formula in the model.

Tautologies, that is, formulas that have the appearance of a tautology although they are not propositional formulas (this is made exact in Problem 200) are valid

- $(A \vee B) \leftrightarrow (B \vee A)$
- $A \vee \neg A$
- $\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)$
- $\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$
- $(A \rightarrow B) \leftrightarrow (\neg A \vee B)$
- $\forall x P_0(x) \vee \neg \forall x P_0(x)$

Here are some valid formulas related to equations:

- $x = x$
- $x = y \to y = x$
- $(x = y \wedge y = z) \to x = z$

Valid quantifier statements

- $\forall x P_n(x) \to P_n(y)$
- $P_n(y) \to \exists x P_n(x)$
- $\forall x P_n(x) \to \forall y P_n(y)$

2.7.3 Logical consequence

Suppose A and B are first order formulas of a vocabulary L. We say that B is a *logical consequence* of A if in any model $\mathcal{M}$, every assignment that satisfies A satisfies B. Equivalently, $A \to B$ is valid. Note: It can be proved that there is no mechanical method for deciding logical consequence. One has to be creative.

2.7.4 Equivalence

Suppose A and B are first order formulas of a vocabulary L. We say that A and B are *(logically) equivalent* if they are logical consequences of each other. Equivalently, $A \leftrightarrow B$ is valid.

Here is a table of simple equivalences:

Formula	Equivalent formula
$\neg\exists x A$	$\forall x \neg A$
$\neg\forall x A$	$\exists x \neg A$
$\forall x(A \wedge B)$	$\forall x A \wedge \forall x B$
$\exists x(A \vee B)$	$\exists x A \vee \exists x B$
$\exists x \exists y A$	$\exists y \exists x A$
$\forall x \forall y A$	$\forall y \forall x A$

2.7.5 Solved problems

Problem 196 *Show that* $\exists x A \vee \exists x B$ *is a logical consequence of* $\exists x(A \vee B)$.

Solution: Suppose $\mathcal{M}$ is a structure and s is an assignment such that s satisfies $\exists x(A \vee B)$ in $\mathcal{M}$. There is an a in M such that $s(a/x)$ satisfies $A \vee B$ in $\mathcal{M}$. Thus $s(a/x)$ satisfies A or B in $\mathcal{M}$. If $s(a/x)$ satisfies A in $\mathcal{M}$, then s satisfies $\exists x A$ and hence $\exists x A \vee \exists x B$ in $\mathcal{M}$. On the other hand, if $s(a/x)$ satisfies B in $\mathcal{M}$, then s satisfies $\exists x B$ and hence again $\exists x A \vee \exists x B$ in $\mathcal{M}$. □

Problem 197 *Show that* $\forall x(A \wedge B)$ *is a logical consequence of* $\forall x A \wedge \forall x B$.

Solution: Suppose $\mathcal{M}$ is a structure and s is an assignment such that s satisfies $\forall x A \wedge \forall x B$ in $\mathcal{M}$. To prove that s satisfies $\forall x(A \wedge B)$ in $\mathcal{M}$, let a be an arbitrary element of M. Since s satisfies $\forall x A \wedge \forall x B$, $s(a/x)$ satisfies both A and B in $\mathcal{M}$. We have shown that s satisfies $\forall x(A \wedge B)$ in $\mathcal{M}$. □

Problem 198 *Show that* $\exists x(P_0(x) \wedge P_1(x))$ *is* not *a logical consequence of* $\exists x P_0(x) \wedge \exists x P_1(x)$.

Solution: Now we have to come up with a model and an assignment. In principle, this could be a haunting challenge. Fortunately, in this case the model and the assignment are easy to find. It makes sense, in general, to first try some very simple models. Either they work, or else they may indicate a direction where a better candidate could be found.

Suppose $\mathcal{M}$ is a structure such that $M = \{0, 1\}$, $P_0^{\mathcal{M}} = \{0\}$ and $P_1^{\mathcal{M}} = \{1\}$. Let s be any assignment. Then s satisfies $\exists x P_0(x) \wedge \exists x P_1(x)$ in $\mathcal{M}$. However, s does not satisfy $\exists x(P_0(x) \wedge P_1(x))$ in $\mathcal{M}$. □

Problem 199 *Show that* $\forall x P_0(x) \vee \forall x P_1(x)$ *is not a logical consequence of* $\forall x(P_0(x) \vee P_1(x))$.

Solution: Suppose $\mathcal{M}$ is a structure such that $M = \{0, 1\}$, $P_0^{\mathcal{M}} = \{0\}$ and $P_1^{\mathcal{M}} = \{1\}$. Let s be any assignment. Then s satisfies $\forall x(P_0(x) \vee P_1(x))$ in $\mathcal{M}$. However, s does not satisfy $\forall x P_0(x) \vee \forall x P_1(x)$ in $\mathcal{M}$. □

Problem 200 *Show that if propositional symbols of a tautology are replaced by first order formulas, a valid formula results.*

Solution: This is a somewhat long proof by induction on the structure of the propositional formulas. We start by formulating the problem properly.

If A is a propositional formula, let A' be the result of replacing each proposition symbol p_i systematically by a first order formula B_i.

- $(p_0 \vee p_1)' = B_0 \vee B_1$.
- $(p_2 \wedge \neg p_1)' = B_2 \wedge \neg B_1$.
- $(p_0 \rightarrow (p_1 \rightarrow p_2)) = B_0 \rightarrow (B_1 \rightarrow B_2)$.

Suppose $\mathcal{M}$ is a structure and s an assignment. Let v be a valuation such that $v(p_i)$ is 1 or 0 according to whether B_i is satisfied by s in $\mathcal{M}$ or not.

Now we can formulate the given task in an exact way:

Claim: $v(A) = 1$ if and only if A' is satisfied by s in $\mathcal{M}$.

Note that when the Claim has been proved, it follows that if A is a tautology, then A' is valid.

We use induction on the structure of (or equivalently, on the number of symbols of) A.

Case 1: A is just p_i. The claim is true by the choice of v.

Case 2: A is not a proposition symbol.

We make an **Induction Hypothesis**: The claim holds for all subformulas of (or equivalently, for all formulas shorter than) A.

Case 2.1: A is $B \vee C$. Then A' is $B' \vee C'$. Now $v(A') = 1$ iff $v(B') = 1$ or $v(C') = 1$. By Induction Hypothesis this is equivalent to s satisfying B' or C'. This is equivalent to s satisfying A'.

Case 2.2: A is $B \wedge C$. Then A' is $B' \wedge C'$. Now $v(A') = 1$ iff $v(B') = 1$ and $v(C') = 1$. By Induction Hypothesis this is equivalent to s satisfying B' and C'. This is equivalent to s satisfying A'.

Case 2.3: A is $\neg B$. Then A' is $\neg B'$. Now $v(A') = 1$ iff $v(B') = 0$. By Induction Hypothesis this is equivalent to s not satisfying B'. This is equivalent to s satisfying A'.

Case 2.4: A is $B \rightarrow C$. Then A' is $B' \rightarrow C'$. Now $v(A') = 1$ iff $v(B') = 0$ or $v(C') = 1$. By Induction Hypothesis this is equivalent to s not satisfying B' or satisfying C'. This is equivalent to s satisfying A'.

Case 2.5: A is $B \leftrightarrow C$. Then A' is $B' \leftrightarrow C'$. Now $v(A') = 1$ iff $v(B') = v(C')$. By Induction Hypothesis this is equivalent to s satisfying B' iff it satisfies C'. This is equivalent to s satisfying A'. QED

□

2.7.6 Problems

Problem 201 *Show that*

1. $\exists x \exists y A$ *is equivalent to* $\exists y \exists x A$.
2. $\forall x \forall y A$ *is equivalent to* $\forall y \forall x A$.

Problem 202 *Show that*

1. $\neg \exists x A$ *is equivalent to* $\forall x \neg A$.
2. $\neg \forall x A$ *is equivalent to* $\exists x \neg A$.

Problem 203 *Show that* $\forall y \exists x A$ *is a logical consequence of* $\exists x \forall y A$.

Problem 204 *Show that* $\exists x \forall y R(x, y)$ *is not equivalent to* $\forall y \exists x R(x, y)$.

Problem 205 *Show that* $\exists x \forall y R(x, y)$ *is not equivalent to* $\forall x \exists y R(x, y)$.

2.8 Free and bound variables

2.8.1 Free and bound

We take a closer look at the variables. If we look carefully, we can see that variables play two different roles in predicate logic. The meaning of $\exists x(xEy)$ is that y has a neighbor. This is a property of y and may be true or false depending on what y is. So the role of y here is that it is the vertex we have "in mind" and we are in a sense "free" to choose the value of y, bearing in mind though, that the formulas may turn out to be either true or falls depending on how we choose y.

The role of x in $\exists x(xEy)$ is different. Its role is to *bind* the quantifier $\exists x$ and the formula xEy together. In fact it does not matter which symbol we use for this binding, and it certainly does not matter what value we have hitherto given to x, as the quantifier $\exists x$ will search for a new value anyway.

2.8.2 Occurrences

Every occurrence of a variable x in a formula of the form $\exists xB$ or of the form $\forall xB$ is called a *bound occurrence*. Occurrences which are not bound are called *free*. Below boxed occurrences are bound, bold face occurrences are free:

$$\exists x(xEy \wedge \forall z(zEy \rightarrow z = x))$$
$$\exists \boxed{x}(\boxed{x}E\mathbf{y} \wedge \forall \boxed{z}(\boxed{z}E\mathbf{y} \rightarrow \boxed{z} = \boxed{x}))$$

$$\exists x(xEy \wedge \exists y(\neg yEx))$$
$$\exists \boxed{x}(\boxed{x}E\mathbf{y} \wedge \exists \boxed{y}(\neg \boxed{y}E\boxed{x}))$$

2.8.3 Assignments and free variables

Whether an assignment s satisfies a formula in a model or not, depends only on the values of s on variables that occur free in the formula (See Problem 211 for an exact statement).

The reason is that the quantifiers mess up anyway the values that the assignment gives to the bound variables. When we form the modified assignment $s(a/x)$, it does not matter what $s(x)$ was, whatever it was, it is now gone.

For example, whether s satisfies $\exists x(xEy \wedge \exists y(\neg yEx))$ or not, depends only on $s(y)$, not on $s(x)$.

2.8.4 Sentences

Some formulas have no free variables. They are called *sentences*. Sentence have a truth value independently of any assignment. So sentences express some property of the model itself, not a property of some predetermined elements.

- $\forall y \exists x(xEy \wedge \exists z(\neg zEx \wedge \neg z = x))$ is a sentence that says of a graph that every vertex has a neighbor with a non-neighbor.

2.8.5 Truth

We finally come to the fundamental concept of logic, namely the concept of truth. Physics, chemistry, geology, biology all talk about truth, but what is special about logic is that we can actually define what truth means.

A sentence A is *true* in a structure $\mathcal{M}$, if some (equivalently, all—see Problem 211) assignment satisfies it. This is denoted

$$M \models A.$$

Otherwise the sentence A is *false* in $\mathcal{M}$. If a sentence A is true in a structure M, the structure $\mathcal{M}$ is called a model of the sentence A.

Note that truth is defined by means of the auxiliary concept of satisfaction. So the primary concept is that of an assignment s satisfying a formula A in a structure $\mathcal{M}$, in symbols $\mathcal{M} \models_s A$, and the secondary concept is that of a sentence being true in a structure $\mathcal{M}$, denote $\mathcal{M} \models A$. It is because of the quantifiers that we cannot get away with using the concept of truth only. We need the concept of satisfaction.

2.8.6 Solved problems

Problem 206 *Tell of each occurrence of a variable in the below formula whether it is bound or free.*

$$\exists x(P_0(x) \wedge P_1(y)).$$

Solution: Boxed occurrences are bound, bold face occurrences are free:

$$\exists \boxed{x}(P_0(\boxed{x}) \wedge P_1(\mathbf{y})).$$

□

Problem 207 *Tell of each occurrence of a variable in the below formula whether it is bound or free.*

$$\forall x(R(x, z) \rightarrow S(x, z))$$

Solution: Boxed occurrences are bound, bold face occurrences are free:

$$\forall \boxed{\mathbf{x}}(R(\boxed{x}, \mathbf{z}) \rightarrow S(\boxed{x}, \mathbf{z}))$$

□

Problem 208 *Which of the following formulas are sentences.*

1. $\forall y \exists x(x < y)$

2. $P_0(c) \vee P_0(d)$

3. $\forall y \exists x(x < y \vee x = c)$

4. $\forall y \exists x(x < y) \vee x = c$

Solution: The first three. The last has two bound occurrences of x, but also a free occurrence of x.
□

2.8.7 Problems

Problem 209 *Tell of each occurrence of a variable in the below formula whether it is bound or free.*

1. $\forall x(P_0(x) \to P_1(y))$

2. $\forall x(xEy \vee yEx)$

3. $\forall x(\forall y(xEy) \vee \forall z(yEz))$

Problem 210 *Which of the following formulas are sentences.*

1. $P_0(x)$

2. $\forall x P_0(x)$

3. $\forall x P_0(y)$

4. $\forall y(\exists x(x < y) \vee \exists x(y < x))$

5. $\forall y(\exists x(x < y) \vee y < x)$

Problem 211 *Suppose A is a formula and $\mathcal{M}$ is a structure. Show that if s and s' are assignments that agree on every variable that occurs free in A, then s satisfies A in $\mathcal{M}$ if and only if s' does.*

2.9 Definability

2.9.1 Set defined by a formula

Like truth, the concept of definability belongs to the center of logic. It is precisely because we want to *define* things that we have formulas in the first place. So let us see what definability means.

Suppose A has only x free. The set P *defined by* the formula A on a structure $\mathcal{M}$ is the set of elements a such

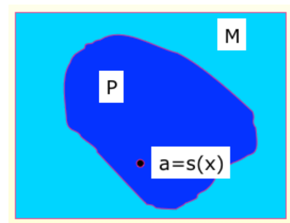


Figure 2.5: Set defined by a formula

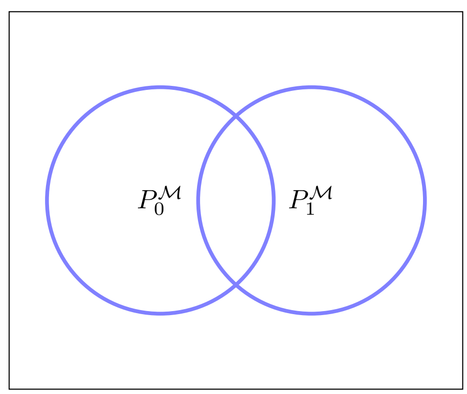


Figure 2.6: Definability on a unary structure

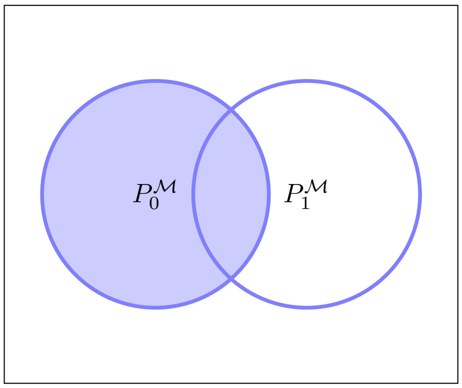


Figure 2.7: The set defined by the formula $P_0(x)$.

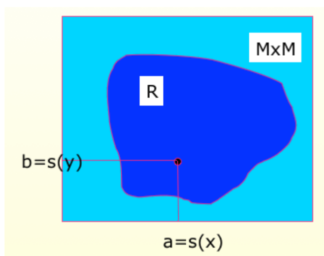


Figure 2.8: Relation defined by a formula

that some (equivalently all) s with $s(x) = a$ satisfies A (See Figure 2.5).

The set defined by the formula $P_0(x)$ on the unary structure on Figure 2.6 is the shaded area in the unary structure of Figure 2.7.

2.9.2 Binary relation defined by a formula

Suppose A is a formula with just x and y free. The binary relation R *defined by* the formula A on a structure $\mathcal{M}$ is the set of pairs (a, b) such that some s with $a = s(x)$ and $b = s(y)$ satisfies A (See Figure 2.8).

See Figure 2.9 for the binary relation defined by the formula formula $x > c \wedge y < c$ on the structure $M = (\mathbb{R}, <, 0)$, $c^M = 0$.

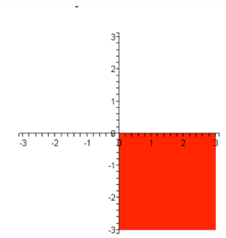


Figure 2.9: Relation defined by a formula

2.9.3 Properties of definable sets

The family of definable subsets of a given structure is closed under union, intersection and complement. I.e. if P and P' are definable on $\mathcal{M}$, then so are $P \cup P'$, $P \cap P'$ and $M - P$. (This family is a so called *Boolean algebra*.)

The family of definable *binary relations* on a given structure is also closed under union, intersection and complement. I.e. if R and R' are definable relations on $\mathcal{M}$, then so are $R \cup R'$, $R \cap R'$ and $M - R$. (This family is also a Boolean algebra.)

2.9.4 Projections

The first projection of a binary relation R on a set M is the set of a in M for which aRb holds for some b in M (See Figure 2.10).

The second projection of a binary relation R on a set $\mathcal{M}$ is the set of b in $\mathcal{M}$ for which aRb holds for some a in $\mathcal{M}$ (See Figure 2.10).

The first and second projections of a definable binary relation are definable sets.

Proof: Suppose R is defined by A on $\mathcal{M}$. The first projection is defined by the formula $\exists y A$. The second by the formula $\exists x A$.

Why? Suppose a is in the first projection. Then there is b such that aRb. Since A defines R, there is an assignment s such that $s(x) = a$, $s(y) = b$ and s satisfies A in $\mathcal{M}$. In particular, $s(b/y)(= s)$ satisfies $\exists y A$ in $\mathcal{M}$ and $s(b/y)(x) = a$. Thus a is in the set defined by $\exists y A$. Conversely, suppose a is in the set defined by $\exists y A$. Then $a = s'(x)$ for some s' which satisfies $\exists y A$ in $\mathcal{M}$. Thus there is b such that $s'(b/y)$ satisfies A in $\mathcal{M}$. Thus

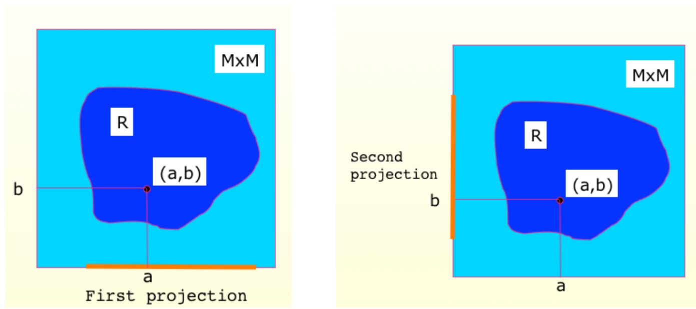


Figure 2.10: The first and second projection of a relation.

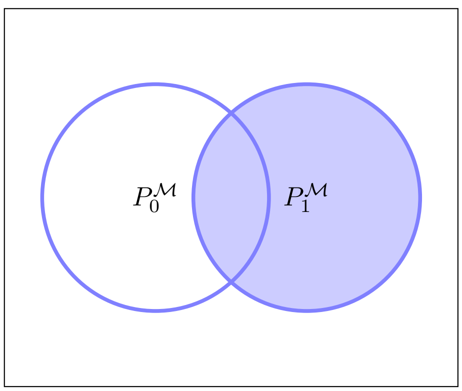


Figure 2.11: The set defined by the formula $P_1(x)$.

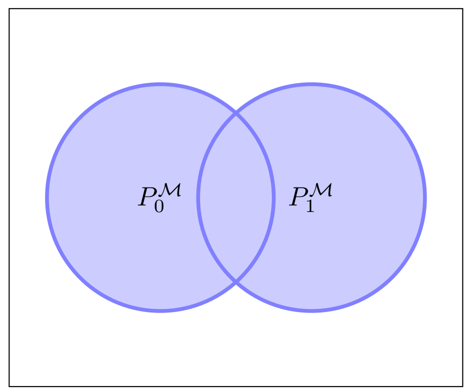


Figure 2.12: The set defined by $P_0(x) \vee P_1(x)$.

$(s'(b/y)(x), s'(b/y)(y))(= (a, b))$ is in R, and we have shown that a is in the first projection of R.

2.9.5 Solved problems

Problem 212 *Find the set defined by the formula $P_1(x)$ on the unary structure of Figure 2.6:*

Solution: The formula $P_1(x)$ defines the set of elements of the model that are in the set $P_1^{\mathcal{M}}$. In other words, the formula $P_1(x)$ defines the set $P_1^{\mathcal{M}}$ (Figure 2.11).

□

Problem 213 *Find the set defined by the formula $P_0(x) \vee P_1(x)$ on the unary structure of Figure 2.6:*

Solution: The formula $P_0(x) \vee P_1(x)$ defines the set of elements of the model that are in the set $P_0^{\mathcal{M}}$ *or* in the set $P_1^{\mathcal{M}}$. In other words, the formula $P_0(x) \vee P_1(x)$ defines the set $P_0^{\mathcal{M}} \cup P_1^{\mathcal{M}}$ (Figure 2.12).

□

Problem 214 *Find the set defined by the formula $P_0(x) \wedge P_1(x)$ on the unary structure of Figure 2.6:*

Solution: The formula $P_0(x) \wedge P_1(x)$ defines the set of elements of the model that are in the set $P_0^{\mathcal{M}}$ *and* in the set $P_1^{\mathcal{M}}$. In other words, the formula $P_0(x) \wedge P_1(x)$ defines the set $P_0^{\mathcal{M}} \cap P_1^{\mathcal{M}}$ (Figure 2.13).

□

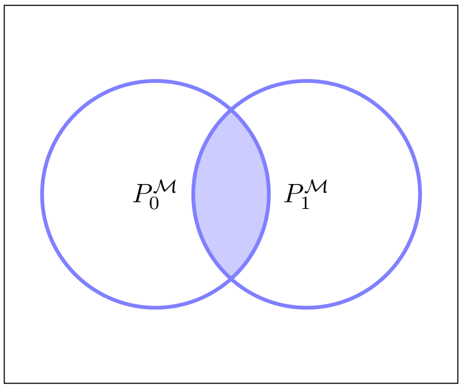
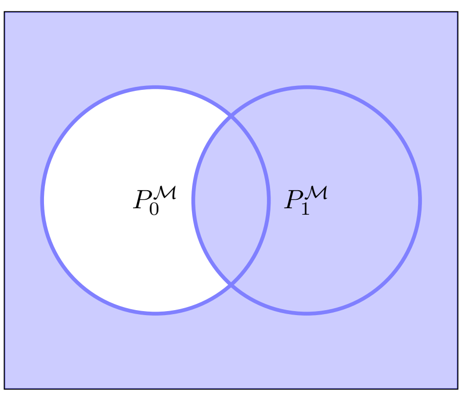


Figure 2.13: The set defined by $P_0(x) \wedge P_1(x)$.

Problem 215 *Find the set defined by the formula $P_0(x) \to P_1(x)$ on the unary structure of Figure 2.6:*

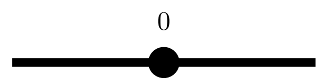
Solution: The formula $P_0(x) \to P_1(x)$ defines the set of elements of the model that either are not in the set $P_0^{\mathcal{M}}$ or else are in the set $P_1^{\mathcal{M}}$. In other words, the formula $P_0(x) \to P_1(x)$ defines the set $(M \setminus P_0^{\mathcal{M}}) \cup P_1^{\mathcal{M}}$.



□

Problem 216 *Describe the ordered structure of the reals with* 0*.*

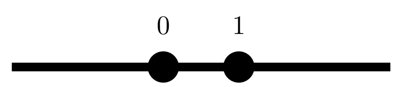
Solution: $\mathcal{M} = (\mathbb{R}, <, 0)$ is the following structure: Its vocabulary is $\{R_0, c\}$, where R_0 is a binary relation symbol and c is a constant symbol. $R_0^{\mathcal{M}} = \{(a, b) : a < b\}$, $c^{\mathcal{M}} = 0$. Note that since our vocabulary has but one constant symbol we can explicitly talk only about one real number namely 0. The other reals can be accessed by means of quantifiers, but the only thing we can really say about them is whether they are negative or positive. So this structure gives a very poor picture of the real numbers. However, it has its uses thanks to its simplicity.



□

Problem 217 *Describe the ordered structure of the reals with* 0 *and* 1*.*

Solution: $\mathcal{M} = (\mathbb{R}, <, 0, 1)$ is the following structure: Its vocabulary is $\{R_0, c, d\}$, where R_0 is a binary relation symbol and c, d are constant symbols. $R_0^{\mathcal{M}} = \{(a, b) : a < b\}$, $c^{\mathcal{M}} = 0$ and $d^{\mathcal{M}} = 1$. This structure is an iota richer than the mere $(\mathbb{R}, <, 0)$, after all, now we can distinguish 1 as a very special real number.



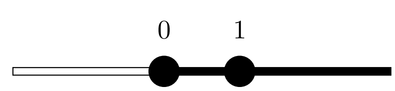
□

Problem 218 *Find the sets defined by the below formulas on the structure $M = (\mathbb{R}, <, 0, 1)$, $c^{\mathcal{M}} = 0$ and $d^{\mathcal{M}} = 1$:*

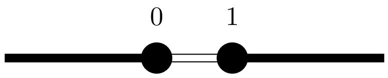
1. $x < c$
2. $c < x \wedge x < d$
3. $d < x$
4. $x < c \vee d < x$

Solution: The defined set is in each picture hollow.

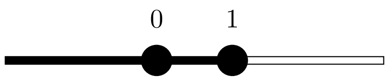
1. $x < 0$:



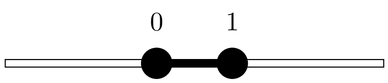
2. $0 < x \wedge x < 1$:



3. $1 < x$:



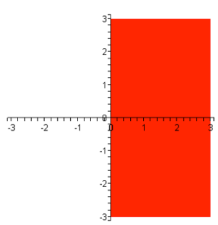
4. $x < 0 \vee 1 < x$:



□

Problem 219 *Find the binary relation defined by the formula $c < x$ on the structure $M = (\mathbb{R}, <, 0)$, $c^{\mathcal{M}} = 0$.*

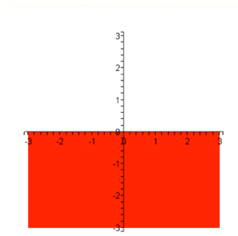
Solution: Note that we want a binary relation. This may look odd, because the formula $c < x$ has only one variable x. But we can think of it simply not making any requirements on y. So it is as if the formula was $c < x \wedge y = y$.



□

Problem 220 *Find the binary relation defined by the formula $y < c$ on the structure $M = (\mathbb{R}, <, 0)$, $c^{\mathcal{M}} = 0$ and $d^{\mathcal{M}} = 1$:*

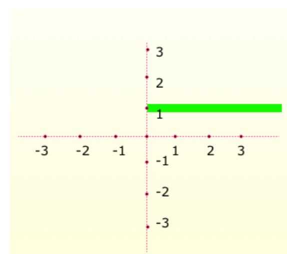
Solution:



□

Problem 221 *Find the binary relation defined by the formula $c < x \wedge y = d$ on the structure $M = (\mathbb{R}, <, 0, 1)$, $c^{\mathcal{M}} = 0$ and $d^{\mathcal{M}} = 1$:*

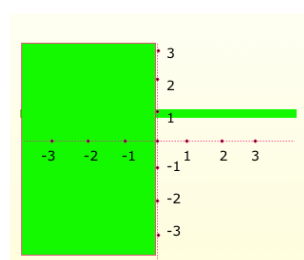
Solution:



□

Problem 222 *Find the binary relation defined by the formula $x < c \vee y = d$ on the structure $M = (\mathbb{R}, <, 0, 1)$, $c^{\mathcal{M}} = 0$ and $d^{\mathcal{M}} = 1$:*

Solution:



□

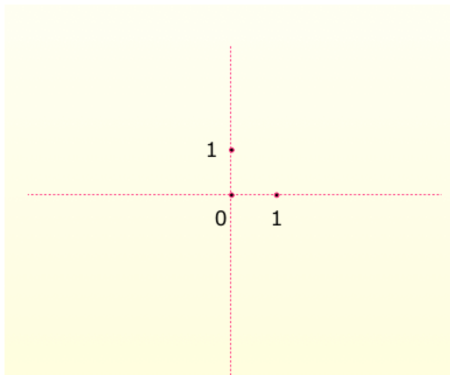


Figure 2.14: Definability on $(\mathbb{R}, <, 0, 1)$.

2.9.6 Problems

Problem 223 *Indicate the set defined by the formula* $P_0(x) \wedge \neg P_1(x)$ *on the unary structure of Figure 2.6*

Problem 224 *Indicate the set defined by the formula* $P_0(x) \leftrightarrow P_1(x)$ *on the unary structure of Figure 2.6*

Problem 225 *Draw the binary relation defined by the formula* $x > d \vee y < c$ *on the structure* $M = (\mathbb{R}, <, 0, 1)$, $c^{\mathcal{M}} = 0$ *and* $d^{\mathcal{M}} = 1$ *(See Figure 2.14).*

Problem 226 *Draw the binary relation defined by the formula* $x < d \rightarrow x = y$ *on the structure* $M = (\mathbb{R}, <, 0, 1)$, $c^{\mathcal{M}} = 0$ *and* $d^{\mathcal{M}} = 1$. *(See Figure 2.14).*

2.10 Terms and substitution

2.10.1 Terms

Constants $\{c, d, \ldots\}$ and variables $\{x, y, z, \ldots\}$ are called *terms*. Characteristic of terms is that they have a *value* if we fix a model and an assignment. In logic sentences have a truth value, which is 0 or 1, just two possibilities, but terms can have any element of the model as their value.

The *value*, denoted $t^{\mathcal{M}}\langle s\rangle$, of a term t in a model $\mathcal{M}$ under the assignment s is defined as follows: If t is a constant c, then $t^{\mathcal{M}}\langle s\rangle$ is $c^{\mathcal{M}}$. If t is a variable x, then $t^{\mathcal{M}}\langle s\rangle$ is $s(x)$.

If we had function symbols such as $+$, $-$ and $\cdot$ there would be more terms: $x + y$, $x \cdot y$, $(x + y) \cdot (x - y)$, $(x \cdot x) \cdot x$, etc. Indeed, we could form *polynomials*.

2.10.2 Changing a bound variable

Within certain limits, a bound variable can be changed to another without changing the meaning of the formula. For example, if you change x to z in $\forall x R_0(x, y)$, the meaning does not change, because the following are equivalent:

$$\mathcal{M} \models_s \forall x R_0(x, y)$$
$$\mathcal{M} \models_s \forall z R_0(z, y)$$

The meaning of both is

$$\{a \in M : (a, s(y)) \in R^{\mathcal{M}}\} = M$$

and neither x nor z plays any role in this.

This is like in algebra

$$\sum_{i=1}^{5} a_i = \sum_{j=1}^{5} a_j.$$

Both are equal to

$$a_1 + a_2 + a_3 + a_4 + a_5.$$

But one has to be careful when changing a bound variable! If you change x to y in $\forall x R_0(x, y)$, the meaning does change: The following are not equivalent in general:

$$\mathcal{M} \models_s \forall x R_0(x, y)$$
$$\mathcal{M} \models_s \forall y R_0(y, y)$$

The meaning of the former is

$$\{a \in M : (a, s(y)) \in R_0^{\mathcal{M}}\} = M$$

while the meaning of the latter is

$$\{a \in M : (a, a) \in R_0^{\mathcal{M}}\} = M.$$

This is like

$$\sum_{i=1}^{5} a_{i,k} \neq \sum_{k=1}^{5} a_{k,k}.$$

The former is

$$a_{1,k} + a_{2,k} + a_{3,k} + a_{4,k} + a_{5,k}$$

but the latter is

$$a_{1,1} + a_{2,2} + a_{3,3} + a_{4,4} + a_{5,5}.$$

The general rule is: When a bound variable is changed to another, no free occurrence of any variable should become bound, and the new bound variable should not conflict with old bound variables, as would happen if in $\forall x \forall y R(x, y)$ the bound variable y was changed to x, resulting with $\forall x \forall x R(x, x)$.

There is an easy way to obey the rule: If a bound variable is changed to a completely *new* variable, no free occurrence of any variable can become bound, and no conflict with existing bound variables can arise.

2.10.3 The concept of *free for*

We shall now dig a little deeper into the question, when can we substitute a variable to another variable in a formula. When we deal with simple short formulas the matter is easy and can be decided by common sense. But we aim at an exact definition which could be even programmed to a computer. After all, computer manipulation of formulas is nowadays important in industrial applications of logic.

We define the following auxiliary concept:

Definition 2.16 *A variable x is* free for *another variable y in a formula A, if no free occurrence of y in A becomes a bound occurrence of x if x is substituted for y in A.*

For example, x is free for y in $\forall z R_0(z, y)$. Substitution yields $\forall z R_0(z, x)$. On the other hand, x is not free for y in $\forall x R_0(x, y)$. Substitution yields $\forall x R_0(x, x)$.

The concept of a variable being free for another variable in a formula can be given also a more exact inductive definition, but we omit it here.

We agree that a *constant* is always free for any variable in any formula. The reason for this is that if a constant is substituted for a variable, it cannot give rise to new occurrences of bound variables, because a constant is not a variable at all.

2.10.4 Substitution

Substitution is a common feature in mathematics. If we have a polynomial $P(x) = x^2 + 3x + 1$, we can substitute values for x and investigate the arising values of the polynomial, e.g. $P(0) = 1, P(1) = 5, P(-1) = -1$.

Similarly, if we have a formula A with a variable y that has a free occurrence in A, then intuitively A says something about y in each model. For some values of y the formula A is true, depending of course on whether there are other variables that occur free in A, and for some other it is false. If we substitute a constant symbol c in the free occurrences of y in A, then the resulting new formula says something about the value of the constant c. If we substitute a completely new variable w in the free occurrences of y in A, then the resulting formula says something about—not any more y, but—the value of w.

More exactly, $A(t/y)$ is the formula obtained from A by substituting the term t for y in every free occurrence of y in A. We never use this notation unless we know that t is free for y in A.

A	$A(x/y)$
$P_0(y)$	$P_0(x)$
$\exists z(zEy)$	$\exists z(zEx)$
$\exists z(R_0(z, y) \to \forall x R_1(x, z))$	$\exists z(R_0(z, x) \to \forall x R_1(x, z))$
$\exists z(R_0(z, y) \to \forall x R_1(x, y))$	(not allowed)

The key property of substitution is:

Lemma 2.17 (Substitution lemma) *If x is free for y in A, then the following are equivalent:*

1. $\mathcal{M} \models_{s(a/y)} A$
2. $\mathcal{M} \models_{s(a/x)} A(x/y)$

This lemma is easy to prove by induction on the length (structure) of A, but we omit it here.

2.10.5 Valid formulas about quantifiers

There are two fundamental valid formulas related to quantifiers. The first is

$$\forall yA \to A(t/y),$$

whenever t is free for y in A.

So why is this valid? Suppose $\mathcal{M}$ is a model and s is an assignment. Let us suppose s satisfies $\forall yA$ in $\mathcal{M}$. Let $a = t^{\mathcal{M}}\langle s\rangle$. Then $s(a/y)$ satisfies A in $\mathcal{M}$. By the Substitution Lemma, s satisfies $A(t/y)$ in $\mathcal{M}$.

The second fundamental valid formula related to quantifiers is

$$A(t/y) \to \exists yA,$$

whenever t is free for y in A.

So why is this valid? Suppose $\mathcal{M}$ is a model and s is an assignment. Let us suppose s satisfies $A(t/y)$ in $\mathcal{M}$. Let $a = t^{\mathcal{M}}\langle s\rangle$. By the Substitution Lemma, $s(a/y)$ satisfies A in $\mathcal{M}$. Thus s satisfies $\exists yA$ in $\mathcal{M}$.

Neither fundamental valid formula is valid if t is not free for y in A.

Recap: We need substitution in formulating logical laws concerning quantifiers. In order that substitution goes right we need the concept of "free for".

2.10.6 Solved problems

Problem 227 *Which of the given terms are free for y in the formula $\exists x(P_0(x) \wedge P_1(y))$?*

1. x
2. c
3. y
4. z

Solution:

term	free for y?
x	no
c	yes
y	yes
z	yes

□

Problem 228 *Which of the given terms are free for z in the formula $\forall x(R(x,z) \to S(x,z))$?*

1. y
2. c
3. x
4. z

Solution:

term	free for y?
y	yes
c	yes
x	no
z	yes

□

Problem 229 *To which of the given variables can the bound variable x be changed in the formula $\exists x(P_0(x) \wedge P_1(y))$*

1. z
2. y
3. x

Solution:

term	can?	result
z	yes	$\exists z(P_0(z) \wedge P_1(y))$
y	no	$\exists y(P_0(y) \wedge P_1(y))$
x	yes	$\exists x(P_0(x) \wedge P_1(y))$

□

Problem 230 *To which of the given variables can the bound variable x be changed in the formula $\forall x(R(x,z) \to S(x,z))$:*

1. z
2. y
3. x

Solution:

term	can?	result
z	no	$\forall z(R(z,z) \to S(z,z))$
y	yes	$\forall y(R(y,z) \to S(y,z))$
x	yes	$\forall x(R(x,z) \to S(x,z))$

□

Problem 231 *Show that the implications*

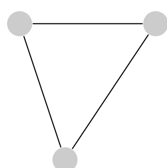
$$\forall yA \to A(t/y),$$

and

$$A(t/y) \to \exists yA,$$

are not valid if the condition "where t is free for y in A" is dropped.

Solution: Let $\mathcal{M}$ be the below graph, and let s be an arbitrary assignment. Then $\mathcal{M} \models_s \forall y \exists x(xEy)$, but it is not true that $\mathcal{M} \models_s \exists x(xEx)$. Indeed, x is not free for y in $\exists x(xEy)$. The other claim is similar.



□

2.10.7 Problems

Problem 232 *Which of the given terms are free for y in the formula $\exists x R_0(y, x) \wedge P_1(y)$?*

1. x
2. c
3. y
4. z

Problem 233 *To which of the given variables can the bound variable x be changed in the formula $\exists x R_0(x, z) \wedge \exists y R_1(z, y)$:*

1. z
2. y
3. x

Problem 234 *Prove the following special case of the Substitution Lemma (without using the Substitution Lemma itself, of course): Suppose A is the formula $\forall z(R_0(y, z) \to P_0(z))$, and the term t is the variable x. Then the following are equivalent, whatever $\mathcal{M}$ and s are:*

1. $\mathcal{M} \models_s A(t/y)$
2. $\mathcal{M} \models_{s(a/y)} A$, *where* $a = t^{\mathcal{M}}\langle s \rangle$.

Problem 235 *Prove the Substitution Lemma.*

2.11 Natural deduction

In propositional logic natural deduction was a way to draw conclusions when the truth table method is too cumbersome. At the same time natural deduction imitates our actual inferences in natural language. When we move to predicate logic, the truth table method is not any more available. Instead of truth tables we have to consider arbitrary models. Since models may be infinite and very complicated there is no mechanical way to go through all models. In fact, validity in predicate logic is a so called *undecidable* problem in the sense that there is no mechanical method to check whether a given sentence is valid or not. For this reason natural deduction is even more important in predicate logic than it was in propositional logic.

2.11.1 What is a natural deduction in predicate logic?

Natural deduction in predicate logic is similar to natural deduction in propositional logic. In particular, the rules for connectives are exactly the same. In addition, we have new rules for quantifiers.

2.11.2 Rules for the universal quantifier

The elimination and introduction rules for the universal quantifier are:

$\forall$-Elimination Rule: The term t has to be free for x in A:

$$\frac{\forall x A}{A(t/x)} \; \forall \mathbf{E}$$

$\forall$-Introduction Rule: The variable x should not occur free in any (uneliminated) assumption in the deduction of A.

$$\frac{A}{\forall x A} \; \forall \mathbf{I}$$

Let us discuss the motivation of these rules. The motivation for the $\forall$-Elimination Rule is the following contemplation: The intuitive meaning of $\forall xA$ is that every object in the universe (of the model) satisfies A when used as a value for x. Therefore, whatever the value of t is, should satisfy A. More exactly, if $\mathcal{M} \models_s \forall xA$, then for any a in M, $\mathcal{M} \models_{s(a/x)} A$, in particular if $a = t^{\mathcal{M}}\langle s\rangle$. By the Substitution Lemma $\mathcal{M} \models_s A(t/x)$.

The motivation for the $\forall$-Introduction Rule is the following: We know A and in a typical situation x occurs free in A. So A says something about x. However, we have made no assumptions about x, because x is not free in any of our assumptions. So x is completely arbitrary, can be anything. Therefore we are justified in concluding $\forall xA$.

More exactly, we have a deduction of A. Intuitively (but this needs a proof) any assignment s that satisfies the assumptions of the deduction of A also satisfies A, because this is what deduction means. We should argue that any assignment that satisfies the assumptions of the deduction of A also satisfies $\forall xA$. For this end, suppose $\mathcal{M}$ is a model and s satisfies all the assumptions of the deduction of A. Then for all a in $\mathcal{M}$, also $s(a/x)$ satisfies those assumptions since x is not free in them (this needs a simple proof but we omit it). Hence s satisfies $\forall xA$.

2.11.3 Example of deduction

Let us look at the following natural language inference:

> If every millionaire is happy or not busy,
> then every busy millionaire is happy.

The reason why we think that this is a valid argument is the following: Take x and assume it is a busy millionaire. By assumption x is happy or not busy. But we assumed x is busy. So x must be happy.

In the language of predicate logic this is:

> If $\forall x(P_0(x) \to (P_1(x) \vee \neg P_2(x)))$,
> then $\forall x((P_0(x) \wedge P_2(x)) \to P_1(x))$.

We can argue informally in predicate logic as follows: Take x and assume $P_0(x) \wedge P_2(x)$. We try to derive $P_1(x)$. By assumption, $P_0(x) \to (P_1(x) \vee \neg P_2(x))$. From this and $P_0(x)$ we get $P_1(x) \vee \neg P_2(x)$. The assumption is $P_2(x)$, so $P_1(x)$. Since x was arbitrary, we get $\forall x(P_0(x) \to (P_1(x) \vee \neg P_2(x)))$. With this in mind we can actually build a real natural deduction (see Figure 2.15).

2.11.4 Solved problems

Problem 236 *Derivation of* $\forall zR_0(z, y)$ *from* $\forall xR_0(x, y)$.

Solution: Note that in the below inference z is free for x in $R_0(x, y)$, and z does not occur free in $\forall zR_0(z, y)$.

$$\cfrac{\cfrac{\forall xR_0(x, y)}{R_0(z, y)}\ \forall\mathbf{E}}{\forall zR_0(z, y)}\ \forall\mathbf{I}$$

□

Problem 237 *Derivation of* $\forall y\forall x(R_0(x, y) \vee R_1(y, z))$ *from* $\forall x\forall y(R_0(x, y) \vee R_1(y, z))$.

Solution: Below in the first two inferences, applications of the $\forall$-Elimination Rule we can observe that x is always free for x and y is always free for y. In the next two inferences, applications of the $\forall$-Introduction Rule we can observe that x and y do not occur free in the assumption $\forall x\forall y(R_0(x, y) \vee R_1(y, z))$.

$$\cfrac{\cfrac{\cfrac{\cfrac{\forall x\forall y(R_0(x, y) \vee R_1(y, z))}{\forall y(R_0(x, y) \vee R_1(y, z))}\ \forall\mathbf{E}}{R_0(x, y) \vee R_1(y, z)}\ \forall\mathbf{E}}{\forall x(R_0(x, y) \vee R_1(y, z))}\ \forall\mathbf{I}}{\forall y\forall x(R_0(x, y) \vee R_1(y, z))}\ \forall\mathbf{I}$$

□

Problem 238 *Derivation of* $\forall xP_1(x)$ *from* $\forall xP_0(x)$ *and* $\forall x(P_0(x) \to P_1(x))$.

Solution: In the below deduction it is necessary to bear in mind that x is always free for x, and that x does not occur free in the assumptions $\forall xP_0(x)$ and $\forall x(P_0(x) \to P_1(x))$.

$$\cfrac{\cfrac{\cfrac{\forall xP_0(x)}{P_0(x)}\ \forall\mathbf{E} \qquad \cfrac{\forall x(P_0(x) \to P_1(x))}{P_0(x) \to P_1(x)}\ \forall\mathbf{E}}{P_1(x)}\ {\to}\mathbf{E}}{\forall xP_1(x)}\ \forall\mathbf{I}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[P_0(x) \wedge P_2(x)]}{P_0(x)}\wedge\mathbf{E} \quad \dfrac{\forall x(P_0(x) \to (P_1(x) \vee \neg P_2(x)))}{P_0(x) \to (P_1(x) \vee \neg P_2(x))}\forall\mathbf{E}}{P_1(x) \vee \neg P_2(x)}\to\mathbf{E} \quad [P_1(x)] \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[P_0(x) \wedge P_2(x)]}{P_2(x)}\wedge\mathbf{E} \quad [\neg P_2(x)]}{P_2(x) \wedge \neg P_2(x)}\wedge\mathbf{I}}{\neg\neg P_1(x)}\neg\mathbf{I}}{P_1(x)}\neg\mathbf{E}}{P_1(x)}\vee\mathbf{E}}{(P_0(x) \wedge P_2(x)) \to P_1(x))}\to\mathbf{I}}{\forall x((P_0(x) \wedge P_2(x)) \to P_1(x)))}\forall\mathbf{I}$$

Figure 2.15: Example.

□

2.11.5 Problems

Problem 239 *Give a natural deduction of "If everybody is amused and tired, then everybody is amused and everybody is tired."*

Problem 240 *Derive* $\forall x R_0(x,x)$ *from* $\forall x \forall y R_0(x,y)$.

Problem 241 *Derive* $\neg\forall x P_0(x)$ *from* $\forall x \neg P_0(x)$.

Problem 242 *Is this a correct deduction:*

$$\dfrac{\dfrac{\forall x R_0(x,y)}{R_0(z,y)}\forall\mathbf{E}}{\forall y R_0(z,y)}\forall\mathbf{I}$$

Problem 243 *Is this a correct deduction:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall x P_0(x)}{P_0(x)}\forall\mathbf{E} \quad \dfrac{\forall x P_1(x)}{P_1(y)}\forall\mathbf{E}}{P_0(x) \wedge P_1(y)}\wedge\mathbf{I}}{\forall x(P_0(x) \wedge P_1(y))}\forall\mathbf{I}}{\forall y \forall x(P_0(x) \wedge P_1(y))}\forall\mathbf{I}$$

2.12 More natural deduction

We learned above the rules for universal quantification. Now come rules for the existential quantifier, and that finishes the set of rules of natural deduction in predicate logic.

2.12.1 Rules for the existential quantifier

The elimination and introduction rules for the existential quantifier are:

∃-Elimination Rule: The variable x should not occur free neither in B nor in any (uneliminated) assumption in the deduction of B, except perhaps in A.

$$\dfrac{\exists x A \quad \begin{array}{c}[A]\\ \vdots \\ B\end{array}}{B}\exists\mathbf{E}$$

∃-Introduction Rule: The term t has to be free for x in A:

$$\dfrac{A(t/x)}{\exists x A}\exists\mathbf{I}$$

Let us discuss the motivation of these rules. The motivation for the ∃-Elimination Rule is the following contemplation: The intuitive meaning of $\exists x A$ is that some object a in the universe (of the model) satisfies A when used as a value for x. On the other hand we can derive B which does not mention x at all, from the assumption A. So, B is true whatever the value of x is as long as A is true. Well, A is true for the value a of x. Therefore B is true.

Let us take a very informal example: Assume that somebody likes jazz, and in addition let us take it as given that everybody who likes jazz likes music. Let us denote by x that person who likes jazz. When we combine this with the assumption that everybody who likes jazz likes music, we can conclude that x likes music. So, indeed, there is somebody who likes music.

$$\frac{\text{Somebody likes jazz} \qquad \dfrac{x \text{ likes jazz} \qquad \begin{array}{l}\text{Everybody who likes jazz likes music}\\ \ldots\end{array}}{\begin{array}{l}\ldots\\ \text{Somebody likes music}\end{array}}}{\text{Somebody likes music}}$$

More exactly, suppose s satisfies $\exists x A$ and the assumptions used in the derivation of B from A. We argue that s must satisfy B. There is an a such that $s(a/x)$ satisfies A. Hence $s(a/x)$ satisfies all the assumptions used in the derivation of B from A. Hence $s(a/x)$ satisfies B. Remember that x is not free in B. So since $s(a/x)$ satisfies B, also s does.

The motivation for the $\exists$-Introduction Rule is the following: We know $A(t/x)$. So we know at least one value of x that satisfies A, namely the value of the term t. Therefore we are justified in concluding $\exists x A$.

More exactly, If $\mathcal{M} \models_s A(t/x)$, then by the Substitution Lemma, $\mathcal{M} \models_{s(a/x)} A$ for $a = t^{\mathcal{M}}\langle s \rangle$. In particular, $\mathcal{M} \models_s \exists x A$.

Without the restriction in the rule $\exists$ E we get the following wrong deduction:

$$\cfrac{\cfrac{\exists x P_0(x) \quad [P_0(x)]}{P_0(x)}\ \exists\,\mathbf{E}}{\forall x P_0(x)}\ \forall\,\mathbf{I}$$

With this wrong deduction we would get completely false results, like: "If some tiles are red, then all tiles are red."

Without the restriction in the rule $\exists$ I we get the following wrong deduction:

$$\frac{\forall x \neg P_0(x,x)}{\exists z \forall x \neg P_0(z,x)}\ \exists\,\mathbf{I}$$

With this wrong deduction we would get completely false results, like: "If no node is a neighbor of itself, then some node has no neighbors."

2.12.2 Solved problems

Problem 244 *We derive $\exists x R_0(x,y)$ from $\exists z R_0(z,y)$.*

Solution:

$$\cfrac{\exists x R_0(x,y) \quad \cfrac{[R_0(x,y)]}{\exists z R_0(z,y)}\ \exists\,\mathbf{I}}{\exists z R_0(z,y)}\ \exists\,\mathbf{E}$$

□

Problem 245 *Derivation of $\exists y \exists x (R_0(x,y) \vee R_1(y,z))$ from $\forall x \forall y (R_0(x,y) \vee R_1(y,z))$.*

Solution: Below we can observe that x is always free for x and y is always free for y:

$$\cfrac{\cfrac{\cfrac{\cfrac{\forall x \forall y (R_0(x,y) \vee R_1(y,z))}{\forall y (R_0(x,y) \vee R_1(y,z))}\ \forall\,\mathbf{E}}{R_0(x,y) \vee R_1(y,z)}\ \forall\,\mathbf{E}}{\exists x (R_0(x,y) \vee R_1(y,z))}\ \exists\,\mathbf{I}}{\exists y \exists x (R_0(x,y) \vee R_1(y,z))}\ \exists\,\mathbf{I}$$

□

Problem 246 *Prove, that if someone is a friend of everybody, then everybody has a friend.*

Solution: The reason why the claim is true is the following: Take x and assume it is a friend of everybody. Let us take somebody y. By assumption x is a friend of y. So y has a friend, namely x.

Now we have to write this into a natural deduction. In other words, we have to derive $\forall y \exists x R_0(x,y)$ from $\exists x \forall y R_0(x,y)$.

$$\cfrac{\exists x \forall y R_0(x,y) \quad \cfrac{\cfrac{\cfrac{[\forall y R_0(x,y)]}{R_0(x,y)}\ \forall\,\mathbf{E}}{\exists x R_0(x,y)}\ \exists\,\mathbf{I}}{\forall y \exists x R_0(x,y)}\ \forall\,\mathbf{I}}{\forall y \exists x R_0(x,y)}\ \exists\,\mathbf{E}$$

□

2.12.3 Problems

Problem 247 *Prove by natural deduction the statement: If someone is a happy millionaire, then someone is happy.*

Problem 248 *Prove by natural deduction*

$$\forall x\exists y R_0(x,y) \wedge \forall x\exists y R_1(x,y)$$

from the assumption

$$\forall x\exists y(R_0(x,y) \wedge R_1(x,y)).$$

Problem 249 *Prove by natural deduction:*

$$\exists x\forall y R_0(x,y) \vee \forall x\exists y\neg R_0(x,y).$$

Problem 250 *Prove by natural deduction:*

$$\forall x\exists y R_0(x,y) \vee \exists x\neg\exists y R_0(x,y).$$

Problem 251 *Prove by natural deduction:*

$$\forall x\exists y(R_0(x,y)\wedge R_1(y,x)) \to \forall u\exists v(R_0(u,v)\vee R_1(v,u)).$$

Problem 252 *Prove by natural deduction: There is a person such that if that person is drunk then everybody is drunk. Hint: To be proved is the sentence*

$$\exists x(P_0(x) \to \forall x P_0(x)).$$

2.13 Natural deduction—Recap

We can collect now all the introduction and elimination rules of propositional logic to a table, see Figure 2.16.

2.13.1 Solved problems

Problem 253 *Derive $\forall x\neg A$ from $\neg\exists xA$.*

Solution: Idea: Suppose it is not true that there are some x satisfying A. So A says something contradictory about x, like x is small and at the same time large, red and not red, or something like that.

So how to prove that every x satisfies $\neg A$. We take an arbitrary x and prove $\neg A$. We use negation introduction rule. So we assume A and derive a contradiction.

The rest is easy. From A follows $\exists xA$ by the $\exists$-introduction rule. And now this is a contradiction with the assumption $\neg\exists xA$.

$$\dfrac{\dfrac{\dfrac{\neg\exists xA \qquad \dfrac{[A]}{\exists xA}\ \exists\,\mathbf{I}^{1)}}{\exists xA \wedge \neg\exists xA}\ \wedge\,\mathbf{I}}{\neg A}\ \neg\,\mathbf{I}}{\forall x\neg A}\ \forall\,\mathbf{I}^{2)}$$

1) x is free for x.

2) The variable x does not occur free in $\neg\exists xA$, the only (uneliminated) assumption in the deduction of $\neg A$. □

Problem 254 *Use the previous solved problem to derive $\exists xA$ from $\neg\forall x\neg A$.*

Solution: Idea: Suppose it is not true that every x fails to satisfy A. So it is not true that A says something so contradictory about x that every x fails to satisfy it. So A gives some hope for x.

So how to prove that some x satisfies A? We have to pull an x from the sleeve. There is no way in sight to use a direct deduction, since we do not know what A is, so we resort to an indirect proof. We assume $\neg\exists xA$ and derive a contradiction.

We use the previous solved problem. There we showed how to derive $\forall x\neg A$ from $\neg\exists xA$. This contradicts the assumption $\neg\forall x\neg A$. We are done!

$$\dfrac{\dfrac{\dfrac{\neg\forall x\neg A \qquad \begin{matrix}[\neg\exists xA]\\ \vdots\\ \forall x\neg A\end{matrix}}{\forall x\neg A \wedge \neg\forall x\neg A}\ \wedge\,\mathbf{I}}{\neg\neg\exists xA}\ \neg\,\mathbf{I}}{\exists xA}\ \neg\,\mathbf{E}$$

□

2.13.2 Problems

Problem 255 **(a)** *Is this a correct deduction:*

$$\dfrac{\dfrac{\forall x R_0(x,y)}{R_0(z,y)}\ \forall\,\mathbf{E}}{\forall y R_0(z,y)}\ \forall\,\mathbf{I}$$

Connective	Introduction	Elimination
Conjunction	$\dfrac{A \quad B}{A \wedge B}\ \wedge\mathbf{I}$	$\dfrac{A \wedge B}{A}\ \wedge\mathbf{E} \qquad \dfrac{A \wedge B}{B}\ \wedge\mathbf{E}$
Disjunction	$\dfrac{A}{A \vee B}\ \vee\mathbf{I} \qquad \dfrac{B}{A \vee B}\ \vee\mathbf{I}$	$\dfrac{A \vee B \quad \begin{matrix}[A] \\ \vdots \\ C\end{matrix} \quad \begin{matrix}[B] \\ \vdots \\ C\end{matrix}}{C}\ \vee\mathbf{E}$
Implication	$\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{A \to B}\ \to\mathbf{I}$	$\dfrac{A \to B \quad A}{B}\ \to\mathbf{E}$
Equivalence	$\dfrac{\begin{matrix}[A] \\ \vdots \\ B\end{matrix} \quad \begin{matrix}[B] \\ \vdots \\ A\end{matrix}}{A \leftrightarrow B}\ \leftrightarrow\mathbf{I}$	$\dfrac{A \leftrightarrow B \quad A}{B}\ \leftrightarrow\mathbf{E} \qquad \dfrac{A \leftrightarrow B \quad B}{A}\ \leftrightarrow\mathbf{E}$
Negation	$\dfrac{\begin{matrix}[A] \\ \vdots \\ B \wedge \neg B\end{matrix}}{\neg A}\ \neg\mathbf{I}$	$\dfrac{\neg\neg A}{A}\ \neg\mathbf{E}$
Universal quantifier	$\dfrac{A}{\forall x A}\ \forall\mathbf{I}$ x should not be free in any assumption of the deduction of A	$\dfrac{\forall x A}{A(t/x)}\ \forall\mathbf{E}$ t has to be free for x in A
Existential quantifier	$\dfrac{A(t/x)}{\exists x A}\ \exists\mathbf{I}$ t has to be free for x in A	$\dfrac{\exists x A \quad \begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{B}\ \exists\mathbf{E}$ x should not be free in B or any assumption of the deduction of B

Figure 2.16: The rules of natural deduction.

(b) *Is this a correct deduction:*

$$\cfrac{\cfrac{\cfrac{\cfrac{\forall x P_0(x)}{P_0(x)}\;\forall\mathbf{E} \qquad \cfrac{\forall x P_1(x)}{P_1(x)}\;\forall\mathbf{E}}{P_0(x) \wedge P_1(y)}\;\wedge\mathbf{I}}{\forall x(P_0(x) \wedge P_1(y))}\;\forall\mathbf{I}}{\forall y \forall x(P_0(x) \wedge P_1(y))}\;\forall\mathbf{I}$$

Problem 256 *Derive* $\neg\forall x P_0(x)$ *from* $\forall x \neg P_0(x)$.

Problem 257 *Give a natural deduction of the sentence* $\forall x P_1(x)$ *from the sentences* $\forall x P_0(x)$ *and* $\forall x(\neg P_1(x) \to \neg P_0(x))$.

Problem 258 *Prove by natural deduction*

$$\neg(\forall x P_0(x) \wedge \exists x \neg P_0(x)).$$

Problem 259 *Prove by natural deduction:*

$$\forall y \exists x \neg R_0(x, y) \to \exists x \exists y \neg R_0(x, y).$$

Problem 260 *Prove by natural deduction:*

$$\forall x \forall y(R_0(x, y) \wedge R_1(x, y)) \to \forall x R_0(x, x).$$

Problem 261 *Prove by natural deduction:*

$$\forall x \forall y(R_0(x, y) \wedge \neg R_1(x, y)) \to \exists x \forall y \neg R_1(x, y).$$

Problem 262 *Prove by natural deduction:*

$$\exists z(\exists x \neg(R_0(x, x) \vee R_1(x, x)) \to \exists y \neg R_0(z, y)).$$

Problem 263 *Prove by natural deduction:*

$$\forall x \neg P_0(x) \to \neg\exists x P_0(x).$$

Problem 264 *Prove by natural deduction:*

$$\exists x \neg P_0(x) \to \neg\forall x P_0(x).$$

Problem 265 *Prove by natural deduction:*

$$\forall x \forall y R_0(x, y) \to \exists x \forall y R_0(x, y).$$

Problem 266 *Prove by natural deduction:*

$$\forall x \forall y R_0(x, y) \to \exists x \exists y R_0(x, y).$$

Problem 267 *Prove by natural deduction:*

$$\forall x \forall y R_0(x, y) \to \forall y \forall x R_0(x, y).$$

Problem 268 *Prove by natural deduction:*

$$\exists x \exists y R_1(x, y) \to \exists x \exists y(R_0(x, y) \to R_1(x, y)).$$

Problem 269 *Prove by natural deduction:*

$$\exists x \exists y \neg R_0(x, y) \to \exists x \exists y(R_0(x, y) \to R_1(x, y)).$$

Problem 270 *Prove by natural deduction:*

$$\forall x \forall y R_1(x, y) \to \forall x \forall y(R_0(x, y) \to R_1(x, y)).$$

Problem 271 *Prove by natural deduction:*

$$\neg(\exists x \neg P_0(x) \wedge \forall x P_0(x)).$$

Problem 272 *Prove by natural deduction:*

$$\neg(\neg\exists x P_0(x) \wedge \neg\forall x \neg P_0(x)).$$

Problem 273 *Prove by natural deduction:*

$$\neg\exists x \neg P_0(x) \vee \neg\forall x P_0(x).$$

Problem 274 *Prove by natural deduction:*

$$\exists x P_0(x) \vee \forall x \neg P_0(x).$$

Problem 275 *Prove by natural deduction:*

$$\neg\forall x \forall y R_0(x, y) \vee \forall x R_0(x, x).$$

Problem 276 *Prove by natural deduction:*

$$\neg\exists x R_0(x, x) \vee \exists x \exists y R_0(x, y).$$

2.14 Soundness

The rules of natural deduction are not arbitrary, they are chosen to reflect the rules that we use when we make inferences in science and everyday life. The rules for the quantifiers are the way they are because that is how we use quantifiers—that is how we understand them. One may be convinced of this by merely looking at the rules. However, this can also be proved and the result is called the Soundness Theorem.

But how can we possibly prove that the rules reflect our everyday use of language? Well, we can prove it to a certain degree of accuracy. This accuracy is provided by the concept of satisfaction of a formula in a model. We can prove that what we can deduce from a formula A by means of natural deduction is satisfied in every model by every assignment that satisfies A in the model. So if our concept $\mathcal{M} \models_s A$ of satisfaction reflects faithfully our concept of truth, then the Soundness Theorem will show that natural deduction preserves truth and is therefore "sound".

The main application of the Soundness Theorem comes in a roundabout way. We can use it to show that certain deductions are impossible. Let us take a simple example: It is intuitively obvious that we cannot deduce $\forall x P_0(x)$ from $\exists x P_0(x)$. But how to prove it conclusively. When we have the Soundness Theorem in our hands, we can simply construct a model in which $\exists x P_0(x)$ is true but $\forall x P_0(x)$ is false. This does it.

2.14.1 Soundness

Soundness of natural deduction in predicate logic means that deductions respect truth in the following sense: If A can be derived from the assumptions $B_1, \ldots, B_n$, and the assignment s satisfies $B_1, \ldots, B_n$ in a structure $\mathcal{M}$, then s also satisfies A in $\mathcal{M}$. The derivability of A from $B_1, \ldots, B_n$ means the existence of a natural deduction that has $B_1, \ldots, B_n$ as the assumptions and A as the conclusion. So we have to look at all such deductions. Deductions are built up from individual steps which are rules, rules for connectives and quantifiers. The idea now is that we show that truth—or rather satisfaction by an assignment in a model—is preserved at each individual step of the deduction. Then truth—or satisfaction–"flows" from the assumptions steps by step to the conclusion. This is just like inductive proofs in arithmetic. If we know that $f(0)$ is even and then go on to show for all n that if $f(n)$ is even then $f(n+1)$ is even, then we know that $f(100)$ is even, because we can—in our mind—start from the evenness of $f(0)$ and proceed step by step all the way to $f(100)$ and then conclude that also $f(100)$ is even.

So we prove the claim: If A has a natural deduction from $B_1, \ldots, B_n$, and s satisfies $B_1, \ldots, B_n$ in a structure $\mathcal{M}$, then s also satisfies A in $\mathcal{M}$.

The proof is "by induction" on the structure of a natural deduction. The deductions consist of steps which are applications of the rules of natural deduction. We proceed from simpler deductions to more complex ones. We show that the conclusions of such applications of the rules are all satisfied by s in $\mathcal{M}$. We go through all the rules and in each case assume that the assumptions of the rule are already satisfied by s in $\mathcal{M}$, and show that the conclusion is likewise satisfies by s in $\mathcal{M}$. Many of the steps are completely trivial, and may have been discussed already when we motivated the rules, but this show not deter us. The point of this proof is the combined effect of all the small steps. Individual steps may be trivial, but the overall conclusion is not.

The shortest possible deduction has some formulas $B_1, \ldots, B_n$ as assumptions and one of them, say B_i, as the conclusion. The claim that any assignment s that satisfies $B_1, \ldots, B_n$ also satisfies B_i is automatically true.

We consider now a deduction with the assumptions $B_1, ..., B_n$ and make the Induction Hypothesis: All smaller deductions satisfy the claim that any assignment s that satisfies the assumptions of the deduction also satisfies the conclusion. What does "smaller" deduction mean. We rely on the intuitive concept that a smaller deduction has fewer applications of the rules.

2.14.2 Conjunction

$\wedge$-Introduction Rule:

$$\frac{A \quad B}{A \wedge B} \wedge\mathbf{I}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. We assume $\mathcal{M} \models_s A$ and $\mathcal{M} \models_s B$. We show $\mathcal{M} \models_s A \wedge B$. But this is trivial!

$\wedge$-Elimination Rules:

$$\frac{A \wedge B}{A} \wedge\mathbf{E} \qquad \frac{A \wedge B}{B} \wedge\mathbf{E}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. We assume $\mathcal{M} \models_s A \wedge B$. We show $\mathcal{M} \models_s A$ and $\mathcal{M} \models_s B$. But this is again trivial!

2.14.3 Disjunction

∨-Introduction Rules:

$$\frac{A}{A \vee B} \vee \mathbf{I} \qquad \frac{B}{A \vee B} \vee \mathbf{I}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. We assume $\mathcal{M} \models_s A$. We show $\mathcal{M} \models_s A \vee B$. But this is trivial! Also, if we assume $\mathcal{M} \models_s B$. Then trivially $\mathcal{M} \models_s A \vee B$.

∨-Elimination Rules:

$$\frac{A \vee B \quad \begin{array}{c}[A]\\ \vdots \\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ C\end{array}}{C} \vee E$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. We assume $\mathcal{M} \models_s A \vee B$. By Induction Hypothesis the derivation of C from A, as well as the derivation of C from B, are sound i.e. if s' is any assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$, and moreover $\mathcal{M} \models_{s'} A$, then $\mathcal{M} \models_{s'} C$, and if $\mathcal{M} \models_{s'} B$, then $\mathcal{M} \models_{s'} C$, whatever s' is. We show $\mathcal{M} \models_s C$. But $\mathcal{M} \models_s A \vee B$ implies $\mathcal{M} \models_s A$ or $\mathcal{M} \models_s B$. In either case we have $\mathcal{M} \models_s C$.

2.14.4 Implication

The →-Elimination Rule is:

$$\frac{A \to B \quad A}{B} \to \mathbf{E}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. Assume $\mathcal{M} \models_s A \to B$ and $\mathcal{M} \models_s A$. We show $\mathcal{M} \models_s B$. This is trivial!

The →-Introduction Rule is:

$$\frac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \to B} \to \mathbf{I}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. By Induction Hypothesis the derivation of B from A is sound, i.e. if s' is any assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$, and moreover $\mathcal{M} \models_{s'} A$, then $\mathcal{M} \models_{s'} B$, whatever s'. We prove $\mathcal{M} \models_s A \to B$. Case 1: Not $\mathcal{M} \models_s A$. Clear! Case 2: $\mathcal{M} \models_s A$. By assumption, in this case $\mathcal{M} \models_s B$, so $\mathcal{M} \models_s A \to B$.

2.14.5 Equivalence

We leave both the formulation of the claim, and the details of the proof as an exercise.

2.14.6 Negation

The ¬-Introduction Rule is:

$$\frac{\begin{array}{c}[A]\\ \vdots \\ B \wedge \neg B\end{array}}{\neg A} \neg \mathbf{I}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. By Induction Hypothesis the inference of $B \wedge \neg B$ from A is sound i.e. if s' is any assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$, and moreover $\mathcal{M} \models_{s'} A$, then $\mathcal{M} \models_{s'} B \wedge \neg B$. But $\mathcal{M} \models_{s'} B \wedge \neg B$ can never hold. So $\mathcal{M} \models_s A$ is false. Hence $\mathcal{M} \models_s \neg A$. Note that this does not mean $\mathcal{M} \not\models_s A$ for all s what-so-ever, only that $\mathcal{M} \not\models_s A$ for those s that satisfy $B_1, \ldots, B_n$ in $\mathcal{M}$.

The ¬-Elimination Rule is:

$$\frac{\neg\neg A}{A} \neg \mathbf{E}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. We assume $\mathcal{M} \models_s \neg\neg A$. We show $\mathcal{M} \models_s A$. Clear!

2.14.7 Quantifiers

∀-Elimination Rule: The term t has to be free for x in A:

$$\frac{\forall x A}{A(t/x)} \forall \mathbf{E}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. Suppose $\mathcal{M} \models_s \forall x A$. Then for any a in M, $\mathcal{M} \models_{s(a/x)} A$, in particular if $a = t^{\mathcal{M}}\langle s \rangle$. By the Substitution Lemma $\mathcal{M} \models_s A(t/x)$.

$\forall$-Introduction Rule: The variable x should not occur free in any (uneliminated) assumption in the deduction of A.

$$\frac{A}{\forall x A} \;\forall\,\mathbf{I}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. Suppose also that s satisfies in $\mathcal{M}$ all uneliminated temporary assumptions in the deduction of A. Now for all a in M, the modified assignment $s(a/x)$ satisfies those assumptions since x is not free in them (this needs a simple proof but we omit it, see Problem 211). By the Induction Hypothesis, $s(a/x)$ satisfies A. Since a was arbitrary, s satisfies $\forall x A$.

$\exists$-Elimination Rule: The variable x should not occur free neither in B nor in any (uneliminated) assumption in the deduction of B, except perhaps in A.

$$\frac{\exists x A \qquad \begin{matrix}[A] \\ \vdots \\ B\end{matrix}}{B} \;\exists\,\mathbf{E}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. Suppose s satisfies $\exists x A$ and the uneliminated temporary assumptions used in the derivation of B from A. We argue that s must satisfy B. There is an a such that $s(a/x)$ satisfies A. Hence $s(a/x)$ satisfies all the assumptions used in the derivation of B from A (we use again Problem 211). By the Induction Hypothesis $s(a/x)$ satisfies B. Remember that x is not free in B. So since $s(a/x)$ satisfies B, also s does.

$\exists$-Introduction Rule: The term t has to be free for x in A:

$$\frac{A(t/x)}{\exists x A} \;\exists\,\mathbf{I}$$

Suppose s is an assignment that satisfies in $\mathcal{M}$ the assumptions $B_1, \ldots, B_n$. If $\mathcal{M} \models_s A(t/x)$, then by the Substitution Lemma, $\mathcal{M} \models_{s(a/x)} A$ for $a = t^{\mathcal{M}}\langle s \rangle$. In particular, $\mathcal{M} \models_s \exists x A$.

2.14.8 Soundness Theorem

Theorem 2.18 *If a formula of predicate logic has a natural deduction, then it is valid. If a formula A of predicate logic has a natural deduction from assumptions which are satisfied by an assignment s in a structure $\mathcal{M}$, then $\mathcal{M} \models_s A$.*

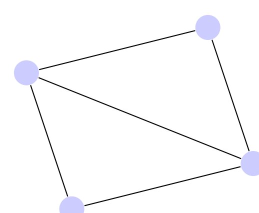
2.14.9 Applications

We can show that a formula B is not derivable by natural deduction from a formula A by finding an assignment s and a structure $\mathcal{M}$ such that $\mathcal{M} \models_s A$ and not $\mathcal{M} \models_s B$. For example, to show that $\exists y \forall x R_0(x, y)$ is not derivable from $\forall x \exists y R_0(x, y)$, we let $M = \{0, 1, 2, \ldots\}$ and $R_0^{\mathcal{M}} = \{(m, n) : m \leq n\}$. Then $\mathcal{M} \models \forall x \exists y R_0(x, y)$ but $\mathcal{M} \not\models \exists y \forall x R_0(x, y)$.

2.14.10 Solved problems

Problem 277 *Show that from the assumption that "every node has a neighbor which is a neighbor of every other node" we cannot derive that "every node is the neighbor of every other node".*

Solution: The assumption is $\forall x \exists y (xEy \land \forall z(\neg y = z \to yEz))$. The conclusion is $\forall y \forall z(\neg y = z \to yEz)$. We have to show that there is no natural deduction of $\forall y \forall z(\neg y = z \to yEz))$ from $\forall x \exists y (xEy \land \forall z(\neg y = z \to yEz))$. Let $\mathcal{G}$ be the below graph. It is easy to show that the given assumption is true in $\mathcal{G}$ but the given conclusion is false in $\mathcal{G}$.



□

Problem 278 *Suppose every red tile is left of every blue tile. Suppose additionally that every blue tile is left of every yellow tile. Show that we cannot derive from these that every red tile is left of every yellow tile.*

Solution: The assumptions are $\forall x(R(x) \to \forall y(B(y) \to x < y))$ and $\forall x(B(x) \to \forall y(Y(y) \to x < y))$. The conclusion is $\forall x(R(x) \to \forall y(Y(y) \to x < y))$.

The problem is to show that there is no natural deduction of $\forall x(R(x) \to \forall y(Y(y) \to x < y))$ from $\forall x(R(x) \to \forall y(B(y) \to x < y))$ and $\forall x(B(x) \to \forall y(Y(y) \to x < y))$.

Let $\mathcal{M}$ be a tile model which has no blue tiles, but has a yellow tile left of a red tile. Then the assumptions are true but the conclusion is false. □

Problem 279 *The following are called the axioms of equivalence relation:*

1. $\forall x \forall y(x \equiv y \to y \equiv x)$ *(Symmetry axiom)*
2. $\forall x(x \equiv x)$ *(Reflexivity axiom)*
3. $\forall x \forall y \forall z((x \equiv y \wedge y \equiv z) \to x \equiv z)$ *(Transitivity axiom)*

Deduce $\forall x \forall y \forall z((x \equiv y \wedge z \equiv y) \to x \equiv z)$ *from the axioms of equivalence relation.*

Solution: See Figure 2.17.
□

2.14.11 Problems

Problem 280 *Show that the following inference is not correct: Suppose every day that is not rainy is not windy, and some day is windy. Then every day is rainy.*

Problem 281 *Show that the following sentence is not derivable by natural deduction:*

$$\exists x \neg P_0(x) \to \neg \exists x P_0(x)$$

Problem 282 *Show that the following sentence is not derivable by natural deduction:*

$$\neg \forall x P_0(x) \to \forall x \neg P_0(x)$$

Problem 283 *Show that the following sentence is not derivable by natural deduction:*

$$\forall x(P_0(x) \vee P_1(x)) \to \forall x P_0(x) \vee \forall x P_1(x)$$

Problem 284 *Show that the following sentence is not derivable by natural deduction:*

$$(\exists x P_0(x) \wedge \exists x P_1(x)) \to \exists x(P_0(x) \wedge P_1(x))$$

Problem 285 *Show that the following sentence is not derivable by natural deduction:*

$$\exists x(P_0(x) \vee P_1(x)) \to \exists x P_0(x)$$

Problem 286 *Show that the following sentence is not derivable by natural deduction:*

$$\forall z(\forall x R_0(x,x) \to \forall y R_0(z,y))$$

Problem 287 *Show that the following sentence is not derivable by natural deduction:*

$$\exists x \forall y R_0(x,y) \to \forall x \exists y R_0(x,y)$$

Problem 288 *Show that the following sentence is not derivable by natural deduction:*

$$\forall x \exists y R_0(x,y) \to \exists x \forall y R_0(x,y)$$

Problem 289 *Show that the following sentence is not derivable by natural deduction:*

$$\forall x R_0(x,x) \to \forall x \forall y R_0(x,y)$$

Problem 290 *Show that the following sentence is not derivable by natural deduction:*

$$\exists x \exists y R_0(x,y) \to \exists x R_0(x,x)$$

Problem 291 *Show that the following sentence is not derivable by natural deduction:*

$$\forall x(P_0(x) \to \forall y P_0(y))$$

Problem 292 *Show that the following sentence is not derivable by natural deduction:*

$$\forall x(\exists y R_0(x,y) \to \forall x \exists y R_0(x,y))$$

Problem 293 *Show that the following inference is not possible in natural deduction:*

$$\{\forall x(P_0(x) \to P_1(x)), \forall x P_1(x)\} \vdash \forall x P_0(x)$$

Problem 294 *Show that the following inference is not possible in natural deduction:*

$$\{\forall x(\neg P_1(x) \to \neg P_0(x)), \forall x P_1(x)\} \vdash \forall x P_0(x)$$

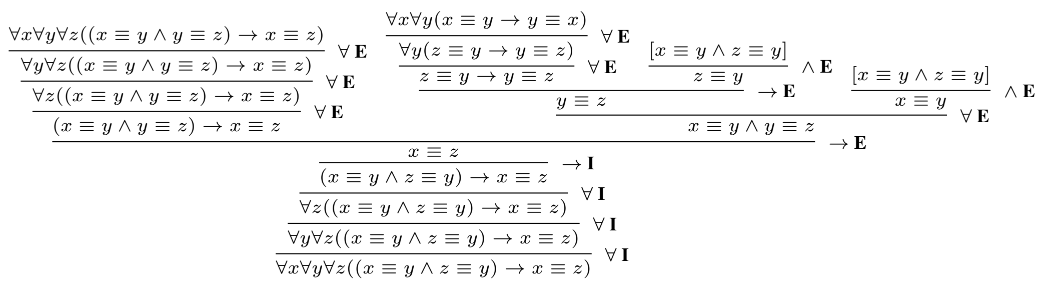


Figure 2.17: A derivation

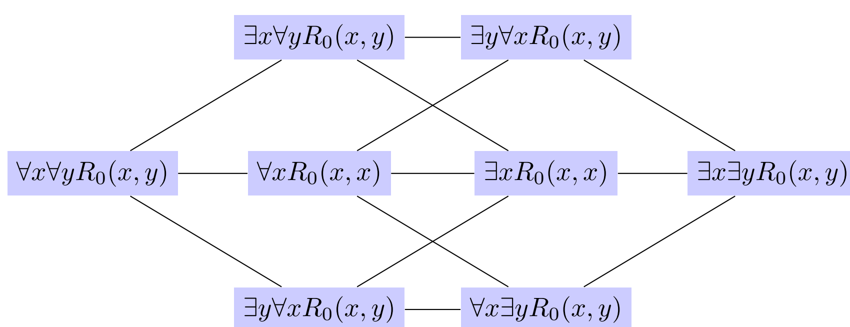


Figure 2.18: Two-quantifier implications.

Problem 295 *Show that the following inference is not possible in natural deduction:*

$$\{\forall x(P_0(x) \to P_1(x)), \exists x P_0(x)\} \vdash \forall x P_1(x)$$

Problem 296 *This is a project rather than a problem. Prove by natural deductions all implications from left to right in Figure 2.18, and show that none of these implications can be reversed and no other implications hold.*

2.15 Axioms and theories

Logic, whether it is propositional logic or the stronger predicate logic, tries to capture what it means to be true in every conceivable situation, or in other words, in every conceivable valuation or model. In most cases, however, we are not interested in truth in every model but only in truth in every model of a certain kind. Sentences that describe what kind of models we are in a particular case interested in are called axioms. A collection of axioms is called a theory. Examples of theories are:

- Graph theory
- Group theory
- Theory of order

Axioms and theories specify what we are interested in, and declare some basic assumptions that we can use in deductions. Mathematics has many kinds of axioms e.g. in algebra (groups, rings, fields, vector spaces, etc) and topology (topological spaces, metric spaces, Hausdorff spaces, etc). Also number theory has its own axioms. Geometry has axioms, laid down by Euclid over 2000 years ago. Newton introduced Axioms of Motion. Axioms occur everywhere in science, but they have special role in mathematics, because mathematics does not have experiments—or does it?

2.15.1 Identity axioms

Identity is so basic that it is almost difficult to see as a particular theory. However, to make inferences about identity we do need some special assumptions—axioms. The following formulas involving identity are valid:

I1 $\forall x\; x = x$

I2 $\forall x \forall y(x = y \to y = x)$

I3 $\forall x \forall y \forall z((x = y \wedge y = z) \to x = z)$

I4 $\forall x \forall y((x = y \wedge P_n(x)) \to P_n(y))$

I5 $\forall x \forall x' \forall y \forall y'((x = y \wedge x' = y' \wedge R_n(x, x')) \to R_n(y, y'))$

These are the identity axioms. They form the theory of identity.

Here are some other formulas about identity that can be derived from the identity axioms:

- $\forall x \exists y(x = y)$ (Everyone is someone)
- $\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n (x_1 = y_1 \wedge \ldots \wedge x_n = y_n \wedge A) \to A(y_1/x_1, \ldots, y_n/x_n)$, when y_i is free for x_i for i=1,...,n. (Identical elements can be substituted for each other)
- $\exists x \forall y(x = y) \to \forall x \forall y(x = y)$ (If there is just one element then any two elements are equal)
- $\exists x \exists y \forall z(z = x \vee z = y) \to \forall x \forall y \forall z(x = y \vee y = z \vee x = z)$ (If there are just two elements then of any three elements two are equal)

Note that the Identity Axioms have been written with certain bound variables. One can change the bound variable with a small trivial deduction, for example:

$$\dfrac{\dfrac{\forall x x = x}{y = y}\;\forall\mathbf{E}}{\forall y y = y}\;\forall\mathbf{I}$$

Identity axioms have the special role that they are always assumed when "=" is part of the formula to be proved.

2.15.2 Finiteness of the universe

The sentence $\exists x_1 \ldots \exists x_n \forall y(y = x_1 \vee \ldots \vee y = x_n)$ is true in a structure if and only if the structure has at most n elements. Note that from this one can deduct $\forall x_1 \ldots \forall x_{n+1}(x_1 = x_2 \vee x_1 = x_3 \vee \ldots \vee x_n = x_{n+1})$. Intuitive reason: if there are only at most n elements then any $n + 1$ elements must have some repetition.

I1	$\forall x\, x = x$
I2	$\forall x \forall y (x = y \rightarrow y = x)$
I3	$\forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$
I4	$\forall x \forall y ((x = y \wedge P_n(x)) \rightarrow P_n(y))$
I5	$\forall x \forall x' \forall y \forall y' ((x = y \wedge x' = y' \wedge R_n(x, x')) \rightarrow R_n(y, y'))$

Figure 2.19: Identity axioms.

2.15.3 Axioms of order

The following are the axioms of order, sometimes also called the axioms of linear order, or of total order.

O1 $\forall x \neg x < x$ (antireflexivity)

O2 $\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$ (transitivity)

O3 $\forall x \forall y (x < y \vee y < x \vee x = y)$ (connectivity)

These axioms are satisfied by the orders of natural numbers with their natural "less than" order, by the real numbers with their natural order, and the rational numbers with their usual order.

The following sentences are provable from the axioms of order

- $\forall x \forall y (x < y \rightarrow \neg y < x)$
- $\forall x \forall y \forall z (x < y \rightarrow (z < y \vee x < z))$
- $\exists x_1 \ldots \exists x_n \forall y (y = x_1 \vee \ldots \vee y = x_n) \rightarrow \exists x \forall y (x < y \vee x = y)$ (If the order is finite, it has a smallest element).
- $\forall x \forall x' ((\forall y (x < y \vee x = y) \wedge \forall y (x' < y \vee x' = y)) \rightarrow x = x')$ (the smallest element is unique, if it exists).

2.15.4 Axioms of tile models

We have discussed many examples of tile models. To make derivations about tile models we need axioms which describe what kind of structures the tile models are. Every tile model has order as part of the structure, but in addition we have the colors. So the axioms of tile models have to include the axioms of order and in addition some extra axioms about the colors. Here they are

T1 $<$ is an order (Tiles are in a certain order side by side from left to right)

T2 $\forall x \neg (B(x) \wedge R(x))$ (No tile is both red and blue)

T3 $\forall x \neg (B(x) \wedge Y(x))$ (No tile is both blue and yellow)

T4 $\forall x \neg (R(x) \wedge Y(x))$ (No tile is both red and yellow)

T5 $\forall x (R(x) \vee B(x) \vee Y(x))$ (Every tile is red, blue or yellow)

Of course, all tile models satisfy the tile axioms. The following sentences are provable from tile axioms

- $\forall x \forall y ((R(x) \wedge B(y)) \rightarrow (x < y \vee y < x))$ (If x is red and y is blue, then one is left of the other.)
- $\forall x (\forall y (R(y) \rightarrow y < x) \rightarrow (B(x) \vee Y(x)))$ (If every red tile is left of x then x must be blue or yellow)
- $\forall x \forall y ((R(x) \wedge B(y)) \rightarrow x < y) \rightarrow \forall x \forall y ((B(x) \wedge R(y)) \rightarrow y < x)$ (If every red tile is left of every blue tile, then every blue tile is right of every red tile)

We think of tile models as finite, but we cannot express this in predicate logic.

2.15.5 Axioms of graph theory

Graphs are binary predicates which are anti reflexive and symmetric. Thus the axioms of graph theory are:

O1	$\forall x \neg x < x$	antireflexivity
O2	$\forall x \forall y \forall z((x < y \land y < z) \to x < z)$	transitivity
O3	$\forall x \forall y(x < y \lor y < x \lor x = y)$	connectivity

Figure 2.20: Axioms of order.

T1	$<$ is an order	Tiles are in a certain order
T2	$\forall x \neg(B(x) \land R(x))$	No tile is both red and blue
T3	$\forall x \neg(B(x) \land Y(x))$	No tile is both blue and yellow
T4	$\forall x \neg(R(x) \land Y(x))$	No tile is both red and yellow
T5	$\forall x(R(x) \lor B(x) \lor Y(x))$	Every tile is red, blue or yellow

Figure 2.21: Axioms of tile models.

G1 $\forall x \neg xEx$ (antireflexivity)

G2 $\forall x \forall y(xEy \to yEx)$ (symmetry)

Of course, all graphs satisfy the graph axioms. Examples of sentences that can be derived fro the axioms of graph theory are:

- $\exists x \forall y(\neg y = x \to yEx) \leftrightarrow \exists x \forall y(\neg x = y \to xEy)$ (This equivalence says "Some node is the neighbor every other node" in two different ways.)
- $\forall x \forall y(\neg y = x \to yEx) \leftrightarrow \forall x \forall y(\neg x = y \to xEy)$ (This equivalence says "Any two disctinct nodes are neighbors" in two different ways.)
- $\forall x \forall y(xEy \to \neg x = y)$ (This is the axiom of antireflexivity in an equivalent form.)

2.15.6 Solved problems

Problem 297 *Prove* $\forall x \exists y(x = y)$ *(Everyone is someone)*

Solution: We want to show that whatever x is given, there is some y that is identical with x. Of course, we choose x itself as y.

$$\dfrac{\dfrac{\dfrac{\forall x(x = x)}{x = x}\ \forall\mathbf{E}}{\exists y(x = y)}\ \exists\mathbf{I}}{\forall x \exists y(x = y)}\ \forall\mathbf{I}$$

□

Problem 298 *Prove* $\forall x_0(P_0(x_0) \to \forall x_1(\neg P_0(x_1) \to \neg x_0 = x_1))$ *(Whoever sings is not identical to anyone who doesn't.)*

Solution: We want to show that whenever a singing a is given and a non-singing b, then $a \neq b$. We use the fact that identity preserves all properties. So $a = b$ leads to a contradiction.

See Figure 2.23 for the derivation.

□

Problem 299 *Prove* $\exists x_0 \forall x_1(x_0 = x_1) \to \forall x_2 \forall x_3(x_2 = x_3)$ *(If there is just one element then any two elements are equal).*

Solution: Let us think about this: We assume that there is some a such that every element is equal to a. Then we

G1	$\forall x \neg xEx$	antireflexivity
G2	$\forall x \forall y (xEy \to yEx)$	symmetry

Figure 2.22: Axims of graph theory.

$$
\dfrac{\dfrac{\dfrac{[\neg P_0(x_1)] \qquad \dfrac{\dfrac{[x_0 = x_1] \quad [P_0(x_0)]}{x_0 = x_1 \wedge P_0(x_0)}\,\wedge\mathbf{I} \qquad \dfrac{\dfrac{\forall x \forall y((x_0 = y \wedge P_0(x_0)) \to P_0(y))}{\forall y((x_0 = y \wedge P_0(x_0)) \to P_0(y))}\,\forall\mathbf{E}}{(x_0 = x_1 \wedge P_0(x_0)) \to P_0(x_1)}\,\forall\mathbf{E}}{P_0(x_1)}\,\to\mathbf{E}}{\dfrac{P_0(x_1) \wedge \neg P_0(x_1)}{\neg x_0 = x_1}\,\neg\mathbf{I}}\,\wedge\mathbf{I}}{\dfrac{\dfrac{\neg P_0(x_1) \to \neg x_0 = x_1}{\forall x_1(\neg P_0(x_1) \to \neg x_0 = x_1)}\,\forall\mathbf{I}}{P_0(x_0) \to \forall x_1(\neg P_0(x_1) \to \neg x_0 = x_1)}\,\to\mathbf{I}}\,\to\mathbf{I}}{\forall x_0(P_0(x_0) \to \forall x_1(\neg P_0(x_1) \to \neg x_0 = x_1))}\,\forall\mathbf{I}
$$

Figure 2.23: Whoever sings is not identical to anyone who doesn't.

take two elements x_2 and x_3. So we know that $a = x_2$ and that $a = x_3$. By the symmetry axiom we get $x_2 = a$ and $a = x_3$. By the transitivity axiom we get $x_2 = x_3$, as desired. See Figure 2.24 for the derivation.
□

Problem 300 *Problem: Prove from the axioms of order:* $x < y \to \neg y < x$

Solution: Suppose $x < y$. Why is it not true that $y < x$? Well, if $y < x$, then by transitivity, $x < x$. But by antireflexivity $x < x$ is false. So $y < x$ leads to a contradiction and we may conclude that $y < x$ is false. See Figure 2.24 for the derivation.
□

2.15.7 Problems

Problem 301 *Prove*

$$\forall x(P(x) \to \exists y(x = y \wedge P(y))).$$

(Everyone who sings is identical to someone who sings.)

Problem 302 *Prove*

$$\forall x(\exists z R(x,z) \to \forall y(x = y \to \exists z R(y,z))).$$

Problem 303 *Prove*

$$(x_1 = y_1 \wedge \ldots \wedge x_n = y_n \wedge A) \to A(y_1/x_1, \ldots, y_n/x_n),$$

when y_i is free for x_i for $i = 1, \ldots, n$. (Identical elements can be substituted for each other)

Problem 304 *Prove*

$$\exists x \exists y \forall z(z = x \vee z = y) \to \forall x \forall y \forall z(x = y \vee y = z \vee x = z)$$

(If there are just two elements then of any three elements two are equal)

Problem 305 *Prove that the sentence*

$$\exists x_1 \ldots \exists x_n \forall y(y = x_1 \vee \ldots \vee y = x_n)$$

is true in a structure if and only if the structure has at most n elements.

Problem 306 *Prove that from*

$$\exists x_1 \ldots \exists x_n \forall y(y = x_1 \vee \ldots \vee y = x_n)$$

one can deduct

$$\forall x_1 \ldots \forall x_{n+1}(x_1 = x_2 \vee x_1 = x_3 \vee \ldots \vee x_n = x_{n+1}).$$

Intuitive reason: if there are only at most n elements then any n+1 elements must have some repetition.

Problem 307 *Show that the axioms of order are satisfied by the orders of*

- *natural numbers* $(\mathbb{N}, <)$
- *real numbers* $(\mathbb{R}, <)$
- *rational numbers* $(\mathbb{Q}, <)$
- $(\{0, 1, \ldots, n\}, <)$

Problem 308 *Prove from the axioms of order*

$$\forall x \forall y \forall z(x < y \to (z < y \vee x < z))$$

Problem 309 *Prove from the axioms of order*

$$\exists x_1 \ldots \exists x_n \forall y(y = x_1 \vee \ldots \vee y = x_n)$$
$$\to \exists x \forall y(x < y \vee x = y)$$

(If the order is finite, it has a least element).

Problem 310 *Prove from the axioms of order*

$$\forall x \forall x'((\forall y(y < x \vee y = x)$$
$$\wedge \forall y(y < x' \vee y = x')) \to x = x')$$

(The largest element is unique, if it exits).

Problem 311 *Prove from tile axioms*

$$\forall x \forall y((R(x) \wedge B(y)) \to (x < y \vee y < x))$$

(If x is red and y is blue, then one is left of the other.)

Problem 312 *Prove from tile axioms*

$$\forall x \forall y((R(y) \to y < x) \to (B(x) \vee Y(x)))$$

(If every red tile is left of x then x must be blue of yellow)

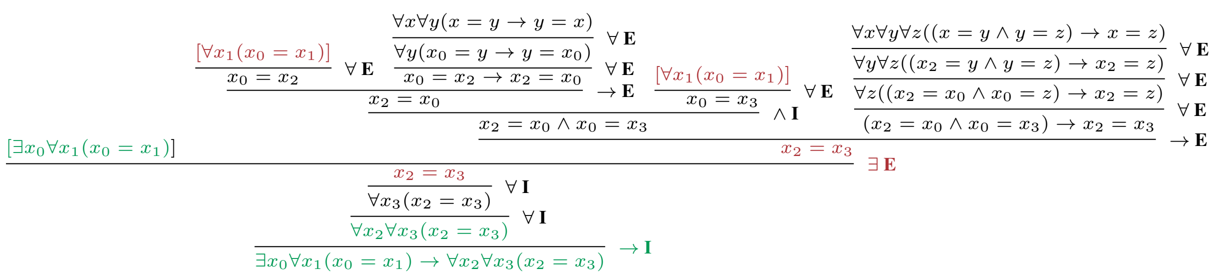


Figure 2.24: If there is just one element then any two elements are equal.

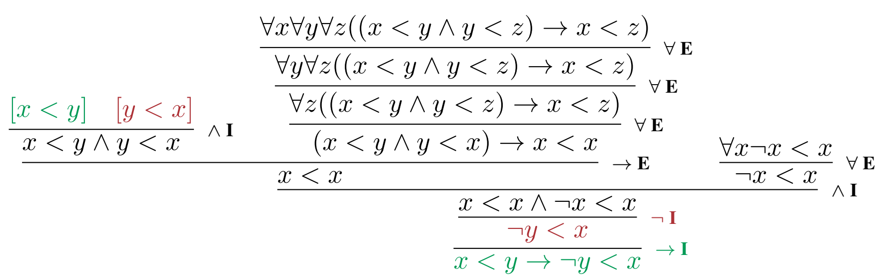


Figure 2.25: Less than is not greater than.

Problem 313 *Prove from tile axioms*

$$\forall x\forall y((R(x) \land B(y)) \to x < y) \to$$
$$\forall x\forall y((B(x) \land R(y)) \to y < x)$$

(If every red tile is left of every blue tile, then every blue tile is right of every red tile.)

Problem 314 *Prove from graph axioms*

$$\exists x\forall y(\neg y = x \to yEx) \leftrightarrow \exists x\forall y(\neg x = y \to xEy)$$

(This equivalence says "Some node is the neighbor every other node" in two different ways.)

Problem 315 *Prove from graph axioms*

$$\forall x\forall y(\neg y = x \to yEx) \leftrightarrow \forall x\forall y(\neg x = y \to xEy)$$

(This equivalence says "Any two distinct nodes are neighbors" in two different ways.)

Problem 316 *Prove from graph axioms*

$$\forall x\forall y(xEy \to \neg x = y)$$

(This is the axiom of antireflexivity in an equivalent form.)

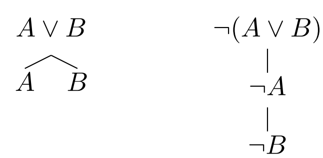
2.16 Semantic trees

As in propositional logic, semantic trees are a powerful alternative proof method—alternative to natural deduction. Semantic proofs are among the most effective and easiest to use, also in predicate logic. In a sense, the semantic tree method proves a sentence by trying to sketch what a model would be like in which the sentence is not true. You may recall that in propositional logic a semantic proof of A shows that the negation $\neg A$ of A is not satisfiable. This implies that A must be valid.

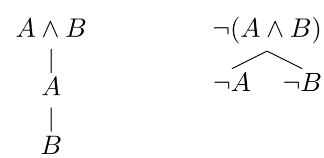
The basic thinking in forming a semantic tree of a formula is the following: On the top of the tree we write a formula and we imagine that we are in the possession of a model and an assignment that satisfies the formula in the model. Then depending on the logical form of the formula—whether it is a conjunction, a disjunction, etc—we write other formulas below the top always following the basic thinking that we write only true formulas, true meaning a formula satisfied by our imagined assignment in the model at hand. When we apply this thinking to quantifiers we may have to change the assignment. In reality there is no assignment and no model, they are just guidelines, a manner of thinking. But if the tree reveals that a formula and its negation are true in the imagined model we know that no such model can exist and the negation of the formula we started with has to be satisfied by every assignment in every model. This is the conclusion we must draw from our honest attempt to defend the idea that the formula is satisfied by some assignment in some model. This sort of thinking is the spirit of the semantic tree method.

Let us recap the rules of forming semantic trees.

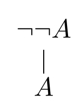
- Disjunction:



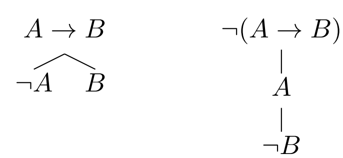
- Conjunction:



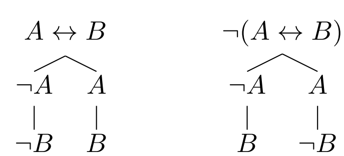
- Negation:



- Implication:



- Equivalence:



2.16.1 Semantic trees for predicate logic

A semantic tree can contain both sentences and formulas. If A is a formula, then $A(t/x)$ is the formula obtained from A by replacing x in its free occurrences by t, and it is assumed that t is free for x in A. Here are some examples:

A	$A(t/x)$
xEy	tEy
$x = y \to \exists x(xEy)$	$t = y \to \exists x(xEy)$
$\forall y(x = y \to \exists x(xEy))$	$\forall y(t = y \to \exists x(xEy))$

2.16.2 Quantifier rules

The rules for the universal quantifier are as follows:

- Universal:

  $$\begin{array}{c} \forall x A \\ | \\ A(t/x) \end{array} \qquad \begin{array}{c} \neg\forall x A \\ | \\ \neg A(d/x) \end{array}$$

  In the first case t is any term on the branch leading to the node where this rule is applied, but t has to be free for x in A. This first rule can be applied several times (see Section 2.16.3). In the second case d is a new constant.

- Existential:

  $$\begin{array}{c} \exists x A \\ | \\ A(d/x) \end{array} \qquad \begin{array}{c} \neg\exists x A \\ | \\ \neg A(t/x) \end{array}$$

  In the first case d is a new constant. In the second case t is any term on the branch leading to the node where this rule is applied, but t has to be free for x in A. This second rule can be applied several times (see Section 2.16.3).

2.16.3 A special feature of the quantifier rules

The rule

$$\begin{array}{c} \forall x A \\ | \\ A(t/x) \end{array}$$

can be used whenever new terms emerge into a branch. The same applies to

$$\begin{array}{c} \neg\exists x A \\ | \\ \neg A(t/x) \end{array}$$

If the branch has no constants, the term x can be used, or one can use the constant symbol c_0.

2.16.4 Closed branch

A branch of a semantic tree is closed if it contains both B and $\neg B$ for some B.

A semantic proof of A is a finite semantic tree for $\neg A$ in which all branches are closed.

For an example, let us look at a semantic proof of

$$\exists x(A \wedge B) \to (\exists x A \wedge \exists x B)$$

$$\begin{array}{c} \neg(\exists x(A \wedge B) \to (\exists x A \wedge \exists x B)) \\ | \\ \exists x(A \wedge B) \\ | \\ \neg(\exists x A \wedge \exists x B) \\ | \\ A(c/x) \wedge B(c/x) \\ | \\ A(c/x) \\ | \\ B(c/x) \\ \diagup \qquad \diagdown \\ \begin{array}{cc} \neg\exists x A & \neg\exists x B \\ | & | \\ \neg A(c/x) & \neg B(c/x) \end{array} \end{array}$$

Sometimes the semantic tree does not close:

Let us build a semantic tree for

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

$$\begin{array}{c} \forall x \exists y R(x, y) \\ | \\ \exists y R(c_0, y) \\ | \\ R(c_0, c_1) \\ | \\ \exists y R(c_1, y) \\ | \\ R(c_1, c_2) \\ | \\ \vdots \end{array}$$

Connective	Rule	Rule for the negation
Disjunction	$A \vee B$ $A \quad\quad B$	$\neg(A \vee B)$ $\neg A$ $\neg B$
Conjunction	$A \wedge B$ A B	$\neg(A \wedge B)$ $\neg A \quad\quad \neg B$
Negation	$\neg\neg A$ A	
Implication	$A \to B$ $\neg A \quad\quad B$	$\neg(A \to B)$ A $\neg B$
Equivalence	$A \leftrightarrow B$ $\neg A \quad\quad A$ $\neg B \quad\quad B$	$\neg(A \leftrightarrow B)$ $\neg A \quad\quad A$ $B \quad\quad \neg B$
Universal	$\forall x A \;\; (\star)$ $A(t/x)$	$\neg \forall x A$ $\neg A(d/x)$
Existential	$\exists x A$ $A(d/x)$	$\neg \exists x A \;\; (\star\star)$ $\neg A(t/x)$

Figure 2.26: The rules of semantic trees. Rules $(\star)$ and $(\star\star)$ can be used whenever new terms appear into a branch.

An attempt to form a semantic tree may fail. This is because not all sentences have a semantic proof; not all sentences are valid. It can be proved that a formula is valid if and only if it has a semantic proof, so this method is complete.

2.16.5 Solved problems

Problem 317 *Build a semantic tree for:*

$$\forall x(P_0(x) \vee P_1(x)) \wedge \neg\exists y P_0(y) \wedge \forall x \neg P_1(x)$$

Solution: We are asked to build the semantic tree for this formula, not a semantic proof. So we start with the formula and follow the rules:

$$\forall x(P_0(x) \vee P_1(x)) \wedge \neg\exists y P_0(y) \wedge \forall x \neg P_1(x)$$
$$\forall x(P_0(x) \vee P_1(x))$$
$$\neg\exists y P_0(y)$$
$$\forall x \neg P_1(x)$$
$$P_0(c_0) \vee P_1(c_0)$$
$$\neg P_0(c_0)$$
$$\neg P_1(c_0)$$
$$P_0(c_0) \qquad P_1(c_0)$$

We can observe that both branches of this tree are closed. This fact has special importance as we shall see later. It can be used to conclude that the formula we start with cannot be satisfied by any assignment in any model.
□

Problem 318 *Build a semantic tree for:*

$$\forall x A \wedge \forall x B \wedge \neg\forall x(A \wedge B)$$

Solution:

$$\forall x A \wedge \forall x B \wedge \neg\forall x(A \wedge B)$$
$$\forall x A$$
$$\forall x B$$
$$\neg\forall x(A \wedge B)$$
$$\neg(A(c_0/x) \wedge B(c_0/x))$$
$$\neg A(c_0/x) \qquad \neg B(c_0/x)$$
$$A(c_0/x) \qquad B(c_0/x)$$

□

2.16.6 Problems

Problem 319 *Give a semantic proof of*

$$\exists x \forall y R(x, y) \to \forall y \exists x R(x, y)$$

Problem 320 *Give a semantic tree of*

$$\exists x(P_0(x) \wedge P_1(x)) \wedge \neg\exists y P_0(y) \wedge \neg\exists x P_1(x)$$

Problem 321 *Give a semantic tree of*

$$\exists x(A \wedge \neg B) \wedge \forall x(A \to B)$$

2.17 More about semantic trees

2.17.1 Soundness of semantic proofs

We will argue below that whenever A has a semantic proof A is valid. If A has a semantic proof, then $\neg A$ has a semantic tree P in which all branches close. To prove that A is valid we assume that $\neg A$ is satisfied by an assignment s in a model $\mathcal{M}$ and derive a contradiction. The assignment s and the model $\mathcal{M}$ help us to construct a branch in P which is not closed. This is a contradiction.

Theorem 2.19 *If A has a semantic proof, then A is valid.*

Suppose P is a semantic tree for $\neg A_0$, but A_0 is not valid. Indeed, let $\mathcal{M}$ be a structure in which A_0 is not satisfied by some assignment s. Then $\neg A_0$ is satisfied by

s in in $\mathcal{M}$. Now we construct a branch of the semantic tree such that every sentence on the branch is satisfied by s in $\mathcal{M}$. The branch is constructed inductively step by step, and as the Induction Hypothesis we assume at each step that s satisfies all formulas on the branch before this step. In the beginning this holds because $\neg A_0$ is satisfied by s in $\mathcal{M}$.

Disjunction:

$A \vee B$ splits into A and B.

$\neg(A \vee B)$ continues with $\neg A$, then $\neg B$.

Suppose we have progressed to a node K, the branch that leads to K has $A \vee B$, and the tree splits into A and B because of the disjunction rule applied to this $A \vee B$. By Induction Hypothesis $A \vee B$ is satisfied by s in $\mathcal{M}$. Then either A or B is satisfied by s in $\mathcal{M}$. If A is satisfied by s, we go left, and if B is satisfied by s, we go right. In either case we maintain truth in $\mathcal{M}$.

Suppose we have progressed to a node K, the branch that leads to K has $\neg(A \vee B)$, and the tree continues with $\neg A$ and $\neg B$ because of the negated disjunction rule. By Induction Hypothesis $\neg(A \vee B)$ is satisfied by s in $\mathcal{M}$. Then both $\neg A$ and $\neg B$ are satisfied by s in $\mathcal{M}$. Thus if we follow the only branch ahead ourselves, we maintain truth in $\mathcal{M}$.

Conjunction:

$A \wedge B$ continues with A, then B.

$\neg(A \wedge B)$ splits into $\neg A$ and $\neg B$.

Suppose we have progressed to $\neg(A \wedge B)$ and this is still satisfied by s in $\mathcal{M}$. Then either $\neg A$ or $\neg B$ is satisfied by s in $\mathcal{M}$. If $\neg A$ is satisfied by s, we go left, and if $\neg B$ is satisfied by s, we go right. In either case we maintain truth in $\mathcal{M}$.

Suppose we have progressed to $A \wedge B$ and this is still satisfied by s in $\mathcal{M}$. Then both A and B are satisfied by s in $\mathcal{M}$. Thus if we follow the only branch ahead ourselves, we maintain truth in $\mathcal{M}$.

Negation:

$\neg\neg A$ continues with A.

Suppose we have progressed to $\neg\neg A$ and this is still satisfied by s in $\mathcal{M}$. Then A is satisfied by s in $\mathcal{M}$. Thus if we follow the only branch ahead ourselves, we maintain truth in $\mathcal{M}$.

Implication:

$A \to B$ splits into $\neg A$ and B.

$\neg(A \to B)$ continues with A, then $\neg B$.

Equivalence:

$A \leftrightarrow B$ splits into $\neg A$, $\neg B$ (left) and A, B (right).

$\neg(A \leftrightarrow B)$ splits into $\neg A$, B (left) and A, $\neg B$ (right).

Universal:

$\forall x A$ continues with $A(t/x)$.

$\neg\forall x A$ continues with $\neg A(d/x)$.

In the first case t is any term on the branch leading to $\forall x A$, which is free for x in A. In the second case d is a new constant.

Suppose we have progressed to $\forall x A$ and this is still satisfied by s in $\mathcal{M}$. Then $A(t/x)$ is satisfied by s whatever the term t, interpreted in M, we have. If t is not yet interpreted in $\mathcal{M}$ or by s (if it is a variable), we can interpret it in an arbitrary way and $A(t/x)$ is still true in $\mathcal{M}$. Thus we maintain truth in $\mathcal{M}$ on the branch that we follow.

Suppose we have progressed to $\neg\forall x A$ and this is still satisfied by s in $\mathcal{M}$. Then there is an element a in M such that the assignment $s(a/x)$ satisfies $\neg A$. Let d be a new constant. Let us interpret d in $\mathcal{M}$ by letting $d^{\mathcal{M}} = a$. Now with this new $\mathcal{M}$, i.e. $\mathcal{M}$ with the new constant, we maintain the condition that $\neg A(d/x)$ is satisfied by s in $\mathcal{M}$.

Existential:

$\exists x A$ continues with $A(d/x)$.

$\neg\exists x A$ continues with $\neg A(t/x)$.

End of the proof: The above process either ends at an atomic formula of else the branch goes on for ever, i.e. is infinite. In either case it is impossible that some sentence and its negation are satisfied by s in $\mathcal{M}$, so the branch cannot be closed.

If we make a semantic tree for $\neg A$ trying all possible rules that apply and still the tree has a branch that is not closed, then we can actually build a structure $\mathcal{M}$ and an assignment s such that $\neg A$ is satisfied by s in $\mathcal{M}$. So in this case A cannot be valid because it is false in $\mathcal{M}$. This is called the Completeness Theorem for the semantic proof method.

2.17.2 Solved problems

Problem 322 *Use the method of semantic trees to construct a model for*

$$\exists x \exists y R_0(x,y) \wedge \neg \forall x R_0(x,x).$$

Solution: Note that there are also other methods to construct models for given sentences, like the blatantly ad hoc method of trying out familiar structures and hoping for the best. However, we use the semantic tree method:

$$\exists x \exists y R_0(x,y) \wedge \neg \forall x R_0(x,x)$$
$$|$$
$$\exists x \exists y R_0(x,y)$$
$$|$$
$$\neg \forall x R_0(x,x)$$
$$|$$
$$\neg R_0(c_0,c_0)$$
$$|$$
$$\exists y R_0(c_1,y)$$
$$|$$
$$R_0(c_1,c_2)$$

The semantic tree suggests the model $\mathcal{M}$ such that the universe of $\mathcal{M}$ is $\{0,1,2\}$, $R_0^{\mathcal{M}} = \{(1,2)\}$, $c_0^{\mathcal{M}} = 0, c_1^{\mathcal{M}} = 1, c_2^{\mathcal{M}} = 2$. It is easy to verify that $\mathcal{M}$ is indeed a model of the given sentence. □

Problem 323 *Use the method of semantic trees to construct a model for*

$$\forall x \exists y R_0(x,y) \wedge \neg \forall x R_0(x,x).$$

Solution: Note that there are also other methods to construct models for given sentences, like the blatantly ad hoc method of trying out familiar structures and hoping for the best. However, we use the semantic tree method:

$$\forall x \exists y R_0(x,y) \wedge \neg \forall x R_0(x,x)$$
$$|$$
$$\forall x \exists y R_0(x,y)$$
$$|$$
$$\neg \forall x R_0(x,x)$$
$$|$$
$$\neg R_0(c_0,c_0)$$
$$|$$
$$\exists y R_0(c_0,y)$$
$$|$$
$$R_0(c_0,c_1)$$
$$|$$
$$\exists y R_0(c_1,y)$$
$$|$$
$$R_0(c_1,c_2)$$
$$|$$
$$\exists y R_0(c_2,y)$$
$$|$$
$$R_0(c_2,c_3)$$
$$|$$
etc.

The semantic tree suggests the model $\mathcal{M}$ such that the universe of $\mathcal{M}$ is $\{0,1,2,\ldots\}$, $R_0^{\mathcal{M}} = \{(n,n+1) : n = 0,1,2,\ldots\}$, $c_0^{\mathcal{M}} = 0, c_1^{\mathcal{M}} = 1, c_2^{\mathcal{M}} = 2, \ldots$. It is easy to verify that $\mathcal{M}$ is indeed a model of the given sentence. □

2.17.3 Problems

Problem 324 *Build a semantic tree for*

$$\exists x(P_0(x) \wedge P_1(x)) \wedge \forall y(P_0(y) \to \neg P_1(y))$$

Problem 325 *Build a semantic tree for*

$$\forall x(A \vee B) \wedge \exists x(\neg A \wedge \neg B)$$

Problem 326 *Build a semantic tree for*

$$\exists x(P_0(x) \wedge \neg P_1(x)) \wedge \forall y(P_0(y) \to P_1(y))$$

Problem 327 *Give a semantic proof of*

$$\exists x \forall y \neg R(x,y) \to \exists x \neg \exists y R(x,y)$$

Problem 328 *Use the method of semantic trees to construct a model for*

$$\forall x \exists y R_0(x,y) \wedge \neg \forall x \exists y R_0(y,x).$$

Note that there are also other methods to construct models for given sentences, like the blatantly ad hoc method of trying out familiar structures and hoping for the best.

Problem 329 *Use the method of semantic trees to construct a model for*

$$\forall x \exists y \forall z (R_0(x, y) \wedge R_0(y, z) \wedge \neg \forall x \forall y R_0(x, y)).$$

Note that there are also other methods to construct models for given sentences, like the blatantly ad hoc method of trying out familiar structures and hoping for the best.

2.18 Validity revisited

In everyday language a person utters a validity if he or she says something which is true but only because of its form, like "every day is rainy or else some days are not rainy".

A formula of predicate logic is valid, or a validity, if it is satisfied by every assignment in every structure.

Here are some examples

- $\exists x A \leftrightarrow \neg \forall x \neg A$
- $\forall x A \leftrightarrow \neg \exists x \neg A$
- $\exists x (A \vee B) \leftrightarrow (\exists x A \vee \exists x B)$
- $\forall x (A \wedge B) \leftrightarrow (\forall x A \wedge \forall x B)$
- $\forall x x = x$

2.18.1 Satisfiable

A formula is satisfiable if it is satisfied by some assignment in some structure. A statement in everyday conversation would be considered satisfiable—"plausible"—if things *could* be like the statement says, and maybe even are. If I say "There is intelligent life outside the Earth in our galaxy", this would probably be considered plausible, possibly a true statement, although we really do not know, and there is a chance that I am wrong. This is what satisfiable means. But now we are talking about predicate logic so satisfiability has a technical meaning, namely being satisfied by some assignment in some model.

Here are some examples

- $\forall x \exists y R_0(x, y) \wedge \neg \exists y \forall x R_0(x, y)$
- $\forall x (P_0(x) \vee P_1(x)) \wedge (\neg \forall x P_0(x) \vee \neg \forall x P_1(x))$
- $\neg(\exists x P_0(x) \rightarrow \forall x P_0(x))$

2.18.2 Refutable

A formula is refutable if there are an assignment s and a structure $\mathcal{M}$ that refute it, i.e. the assignment s does not satisfy the formula in $\mathcal{M}$. It is as if someone says "Every day in August is rainy" and you refute it by pointing out that in the year 1996 there was a day in August when it did not rain.

Here are some examples from predicate logic:

- $\exists x P_0(x) \rightarrow \forall x P_0(x)$
- $\forall x \exists y R_0(x, y) \rightarrow \exists y \forall x R_0(x, y)$
- $x = y$

2.18.3 Contingent

A formula is contingent if it is both satisfiable and refutable. A person utters a contingency, like "It is raining", "Someone is vacuuming upstairs", or "Bach is the greatest composer of all", if what he or she says can be true but can also be false. So contingencies are satisfiable but not conversely, as validities are satisfiable but not refutable.

Here are some examples from predicate logic.

- $\forall x P_0(x)$
- $\exists x \forall y R_0(x, y)$
- $\exists x (R_0(x, y) \wedge P_0(x))$
- $x = y$ (depending on the assignment this can be true or false)

2.18.4 Contradiction

In everyday language a person utters a contradiction if he or she says something which is false merely because of its form, like "I am a vegetarian and I am not a vegetarian". Contradicting oneself in a social situation is problematic, and usually leads to being ignored or else asked to clarify what one means. If one utters only validities in a conversation, one does not make much progress. It is the same with contradictions. Therefore people usually utter contingencies, statements that could be right or wrong. In everyday life as also in science contingencies are judged in the light of empirical data. If I say “It rained yesterday in New York”, I could be—a priori—right or wrong, so it is a contingency, but of course we can check the meteorological data and make an a posteriori judgement that my statement was right, or wrong.

A formula of predicate logic is a contradiction if it is not satisfied by any assignment in any structure.

Here are some examples from predicate logic:

- $\forall x A \wedge \exists x \neg A$
- $\exists x A \wedge \forall x \neg A$
- $\forall x(A \rightarrow B) \wedge \exists x(A \wedge \neg B)$
- $x = y \wedge x = z \wedge \neg y = z$

2.18.5 Categories of formulas of predicate logic

Every formula of predicate logic is either a validity, a contradiction or a contingency.

Every satisfiable formula is either valid or contingent.

Every refutable formula is either a contradiction or a contingency.

Recognizing what the type of a given formula is, whether it is a validity, a contradiction or a contingency, is the bread and butter of a logician. However, this may be a difficult taks.

2.18.6 Hard question

Given a formula, can we decide mechanically whether it is a validity, a contradiction or a contingency?

It can be proved that this is not possible, if mechanically is interpreted as is now common. The usual current definition of being decidable mechanically, due to the British mathematician Alan Turing (1912—1954), is roughly the same as being decidable in finite time by a computer program which has a finite but endless amount of time and memory.

2.18.7 The method of deductions

Given a potential deduction for a formula A it is not difficult to check whether the deduction is correct or not.

This can be done mechanically. Such computer programs are called proof checkers (see e.g. http://coq.inria.fr/). One can make a list of all possible deductions and check them one by one. The hard case is when there is no deduction, because then it takes infinite time to find it out.

2.18.8 Equivalence of formulas

Two formulas of predicate logic, A and B, are called (logically) equivalent if $A \leftrightarrow B$ is a validity. Equivalence of formulas is used in everyday language and in science all the time, often without paying much attention to it.

2.18.9 Equivalent formulas of predicate logic

Here are some important logical equivalences in predicate logic:

Formula	Equivalent formula	Condition
$\forall x A$	$\neg\exists x \neg A$	
$\exists x A$	$\neg\forall x \neg A$	
$\forall x(A \wedge B)$	$\forall x A \wedge \forall x B$	
$\exists x(A \vee B)$	$\exists x A \vee \exists x B$	
$\exists x \exists y A$	$\exists y \exists x A$	
$\forall x \forall y A$	$\forall y \forall x A$	
$\forall x(A \rightarrow B)$	$\exists x A \rightarrow B$	x not free in B
$\forall x(A \rightarrow B)$	$A \rightarrow \forall x B$	x not free in A
$\exists x(A \wedge B)$	$\exists x A \wedge B$	x not free in B

2.18.10 Method

Suppose we have to decide whether A is satisfiable or a contradiction. We build a semantic tree for A.

If all branches end up closed, then we have a semantic proof of $\neg A$. Then A is a contradiction.

On the other hand, if we apply all the rules to their fullest extent and still have a branch which is not closed, then we can use it to build a model for A. Then A is satisfiable. Note that the branch may be infinite.

2.18.11 Solved problems

Problem 330 *Prove the following logical equivalence in predicate logic using natural deduction:* $\forall x(A \rightarrow B) \leftrightarrow$

$(\exists xA \to B)$, *assuming that x is not free in B.*

Solution:

$$\dfrac{\dfrac{\exists xA \qquad \dfrac{\dfrac{\forall x(A \to B)}{A \to B}\,\forall\mathbf{E} \qquad [A]}{B}\,\to\mathbf{E}}{B}\,\exists\mathbf{E}^{1)}}{\exists xA \to B}\,\to\mathbf{I}$$

1) x is not free in B.

We could do the same with a semantic tree:

$$\begin{array}{c}
\neg(\forall x(A \to B) \to (\exists xA \to B)) \\
| \\
\forall x(A \to B) \\
| \\
\neg(\exists xA \to B) \\
| \\
\exists xA \\
| \\
\neg B \\
| \\
A(c_0/x) \\
| \\
(A \to B)(c_0/x) \\
| \\
A(c_0/x) \to B \\
\swarrow \quad \searrow \\
\neg A(c_0/x) \qquad B
\end{array}$$

Both branches close, so this is a semantic proof of $\forall x(A \to B) \to (\exists xA \to B)$.

Now the other direction:

$$\dfrac{\dfrac{\dfrac{\exists xA \to B \qquad \dfrac{[A]}{\exists xA}\,\exists\mathbf{I}}{B}\,\to\mathbf{E}}{A \to B}\,\to\mathbf{I}}{\forall x(A \to B)}\,\forall\mathbf{I}^{1)}$$

1) x is not free in B. □

2.18.12 Problems

Problem 331 *Use the method of semantic trees to construct a model for*

$$\exists x\forall yR_0(x, y) \wedge \neg\forall xR_0(x, x).$$

Problem 332 *Prove using the semantic tree method: The formula*

$$\exists x(A \wedge B) \to (A \wedge \exists xB)$$

is valid, assuming that x is not free in A.

Problem 333 *Consider the formula*

$$(R(x, y) \vee \forall xP(x)) \to \forall x(R(x, y) \vee P(x)).$$

Is the formula valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 334 *Decide whether the following sentence is a validity, a contingency or a contradiction.*

$$\exists xP_0(x) \to \forall xP_0(x)$$

Problem 335 *Decide whether the following sentence is a validity, a contingency or a contradiction.*

$$\neg(\neg\forall xP_0(x) \wedge \neg\exists x\neg P_0(x))$$

Problem 336 *Decide whether the following sentence is a validity, a contingency or a contradiction.*

$$\forall xP_0(x) \vee \exists x\neg P_0(x)$$

Problem 337 *Decide whether the following sentence is a validity, a contingency or a contradiction.*

$$\forall x\forall y(R_0(x, y) \wedge \neg R_1(x, y)) \to \exists x\forall y\neg R_1(x, y)$$

2.19 n-ary predicates

Up to now we have dealt with unary predicates (relations) and binary ones. Indeed, binary relations are much more common than the higher dimensional relation we shall now introduce.

2.19.1 Ternary predicates

A ternary (3-place) predicate binds three elements just as a binary predicate binds two elements. Predicates are also called relations. The relation

"x, y and z are on the same line"

is an example of a ternary relation on the plane.

A ternary relation on a set M is any subset of M^3, i.e. any set of triples (a, b, c), where a,b,c are from M.

In order to be able to deal with ternary relations by means of predicate logic we henceforth allow vocabularies to have

- Unary predicate symbols $P_0, P_1, \ldots$
- Binary predicate symbols $R_0, R_1, \ldots$
- Ternary predicate symbols $R^3_0, R^3_1, \ldots$

In a structure $\mathcal{M}$ a ternary predicate symbol R^3_n is interpreted as a ternary relation $(R^3_n)^{\mathcal{M}} \subseteq M^3$.

Note that new symbols in the vocabulary mean also new atomic formulas, such as $R^3_n(x, y, z)$.

2.19.2 n-ary relations

An n-ary (n-place) relation (or predicate) binds n elements just as a ternary relation binds three elements.

$$\text{"}x - y = z - u\text{"}$$

is a 4-ary relation on the reals. (Equidistance relation)

"Student x in course y got z credit points in exam u in the year z"

is a 5-ary relation. n-ary relations resemble in many ways what are called databases in computer science. More technically speaking, an n-ary relation on a set M is any set of n-tuples $(a_1, \ldots, a_n)$ of elements of M.

In order to be able to deal with n-ary relations by means of predicate logic we henceforth allow vocabularies to have

- Unary predicate symbols $P_0, P_1, \ldots$
- Binary predicate symbols $R_0, R_1, \ldots$
- n-ary predicate symbols $R^n_0, R^n_1, \ldots$

In a structure $\mathcal{M}$ an n-predicate symbol R^n_i is interpreted as an n-ary relation $(R^n_i)^{\mathcal{M}} \subseteq M^n$.

Note that new symbols in the vocabulary mean also new atomic formulas, such as $R^n_i(y_1, \ldots, y_n)$.

2.19.3 Predicate logic with n-ary predicate symbols

There is no change to the rules of natural deduction and semantic trees. Deductions are made just as before. $R^3_0(x, y, z)$ is just like any other formula with three free variables, e.g. one of the following:

$$xEy \wedge yEz$$

$$P_0(x) \wedge P_1(y) \wedge P_2(z)$$

$$R_0(x, y) \vee R_0(x, z) \vee R_0(y, z)$$

except that it is atomic, that is, it has no internal logical structure, like the above there formulas.

Since we have new atomic formulas we have to also add new identity axioms:

I1 $\forall x x = x$

I2 $\forall x \forall y (x = y \rightarrow y = x)$

I3 $\forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$

I4 $\forall x \forall y ((x = y \wedge P_n(x)) \rightarrow P_n(y))$

I5 $\forall x \forall x' \forall y \forall y' ((x = y \wedge x' = y' \wedge R_n(x, x')) \rightarrow R_n(y, y'))$

I6 $\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n ((x_1 = y_1 \wedge \ldots \wedge x_n = y_n \wedge R^n_i(x_1, \ldots, x_n)) \rightarrow R^n_i(y_1, \ldots, y_n))$

The new axiom I6 is just an elaboration of I4 and I5 to the new situation of n-ary predicate symbols.

2.19.4 Solved problems

Problem 338 *Consider the ternary relation*

$$\text{"}x \text{ times } y \text{ equals } z\text{"}$$

on natural numbers. Let us use this relation to interpret R^3_0 in the structure $\mathcal{N}$, the universe of which is the set of natural numbers, that is $(a, b, c) \in (R^3_0)^{\mathcal{N}}$ if and only if $a \cdot b = c$.

Which of the following sentences are true in $\mathcal{N}$:

1. $\forall x \forall y \exists z R^3_0(x, y, z)$
2. $\forall x \forall z \exists y R^3_0(x, y, z)$

3. $\forall x\forall y\forall z\forall u((R_0^3(x,y,z)\wedge R_0^3(x,y,u))\rightarrow z=u)$.

Solution: Let us take an arbitrary assignment s.

1. $\mathcal{N}\models_s \forall x\forall y\exists z R_0^3(x,y,z)$, as can be seen as follows: First note that $(a,b,ab)\in(R_0^3)^{\mathcal{N}}$ for all $a,b\in N$. Hence

$$N\models_{s(a/x,b/y,ab/z)} R_0^3(x,y,z)$$

for all $a,b\in N$. Hence

$$N\models_{s(a/x,b/y)} \exists z R_0^3(x,y,z)$$

for all $a,b\in N$. Hence

$$N\models_s \forall x\forall y\exists z R_0^3(x,y,z).$$

2. $N\not\models_s \forall x\forall z\exists y R_0^3(x,y,z)$, as can be seen as follows: First note that $(2,a,3)\notin(R_0^3)^{\mathcal{N}}$ for all $a\in N$. Hence

$$N\not\models_{s(2/x,3/z,a/Y)} R_0^3(x,y,z)$$

for all $a\in N$. Hence

$$N\not\models_{s(2/x,3/z)} \exists y R_0^3(x,y,z).$$

Hence

$$\not\models_s \forall x\forall y\exists z R_0^3(x,y,z).$$

3. $\mathcal{N}\models_s \forall x\forall y\forall z\forall u((R_0^3(x,y,z)\wedge R_0^3(x,y,u))\rightarrow z=u)$, as can be seen as follows: First note that if $(a,b,c)\in(R_0^3)^{\mathcal{N}}$ and $(a,b,d)\in(R_0^3)^{\mathcal{N}}$ then $c=ab=d$, i.e. $c=d$ Hence

$$\mathcal{N}\models_{s(a/x,b/y,c/z,d/u)} (R_0^3(x,y,z)\wedge R_0^3(x,y,u)) \rightarrow z=u$$

for all $a,b,c,d\in N$. Hence

$$\mathcal{N}\models_s \forall x\forall y\forall z\forall u((R_0^3(x,y,z)\wedge R_0^3(x,y,u)) \rightarrow z=u).$$

□

2.19.5 Problems

Problem 339 *Use the method of semantic trees to construct a model for*

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

Problem 340 *Prove using the semantic tree method: The sentence*

$$\forall y\forall z(\exists x(R'(y)\wedge R(x,y,z))\rightarrow(R'(y)\wedge\exists x R(x,y,z)))$$

is valid.

Problem 341 *Consider the sentence*

$$\forall x(P(c,d)\rightarrow R(x,c,d))\rightarrow(P(c,d)\rightarrow\forall x R(x,c,d)).$$

Is the sentence valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 342 *Consider the sentence*

$$(R(c,d,e)\vee\forall x P(x))\rightarrow\forall x(R(x,d,e)\vee P(x)).$$

Is the sentence valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 343 *Consider the sentence*

$$\forall x(P(c,d)\wedge R(x,c,d))\wedge\exists x(P(c,d)\rightarrow\neg R(x,c,d)).$$

Is the sentence valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 344 *Consider the structure $\mathcal{M}$ with the three unary predicates and the one 3-place relation R of the below table in the set* {*Anna, Joonas, Minna, Tero, Harri, Logic, Algebra, 4,5*}.

P_0	P_1	P_2
Anna	*Logic*	*5*
Joonas	*Algebra*	*4*
Minna	*Algebra*	*4*
Tero	*Logic*	*4*
Harri	*Algebra*	*5*

Which of the following sentences are true in $\mathcal{M}$?

1. $\exists x \exists y((P_0(x) \wedge P_0(y) \wedge \exists z \exists u(R(x,z,u) \wedge R(y,z,u) \wedge \neg x = y))$
2. $\exists x \exists y((P_1(x) \wedge P_1(y) \wedge \exists z \exists u(R(z,x,u) \wedge R(z,y,u) \wedge \neg x = y))$
3. $\exists x \exists y((P_2(x) \wedge P_2(y) \wedge \exists z \exists u(R(u,z,x) \wedge R(u,z,y) \wedge \neg x = y))$

2.20 Functions

The concept of a function is familiar from

- Calculus: $\sin(x)$, $\sqrt{1+x^2}$
- Algebra: x^3,, x^{-1},
- Set theory: $\{(x,y) \in R^2 : x^3 = y\}$.

To be able to use logic in the study of functions we introduce function symbols.

Also functions of several variables are familiar from

- Calculus xy, $\sin(x+y)$, $sin(x)cos(y)$
- Algebra $x \cdot y$, $x^{-1}y$
- Linear algebra $2x + 5y$, $10x - y + 2z$
- Set theory $\{(x,y,z) \in R^2 : x + y = z\}$

The introduction of function symbols extends the applicability of logic but at the same time makes logic more complicated, as we shall see below.

2.20.1 Function symbols

To be able to use logic in the study of functions of one or several variables we introduce n-ary function symbols for all $n > 0$.

We allow henceforth vocabularies to contain also function symbols

- F_0^n, F_1^n ,...

Here F_i^n is called an n-ary function symbol. We use F, F', G, G', etc as shorthands for function symbols.

If the vocabulary contains function symbols, we interpret these symbols as functions in the domain of the structure:

The function symbol F_i^n is interpreted in a structure $\mathcal{M}$ with domain M as an n-ary function on M i.e.

$$(F_i^n)^{\mathcal{M}} : M^n \to M$$

Note: these functions are total i.e. defined everywhere. So if we want to take a symbol F_0^1 for $1/x$ in the domain of all real numbers, we have to define the interpretation of F_0^1 also at $x = 0$. We can define e.g. $1/0 = 0$, but normally in mathematics the function $1/x$ on the reals is considered as a partial function, that is, as a function that is not defined everywhere.

Now that we have function symbols we can build new terms:

- $F(x), G(x,c)$
- $F(F(x)), G(F(x),c)$
- Etc

Note: we allow nesting of function symbols, as in:

$$G(F(x),c).$$

This gives new life to the world of terms. Terms are not anymore mere constants or variables, they can be very long nested expressions, a bit like polynomials.

Note: Some terms do not have variables, e.g. $G(F(c),c)$. They are called constant terms.

With the extended concept of a term we get also new atomic formulas

$$t = t'$$

from our new terms. For example:

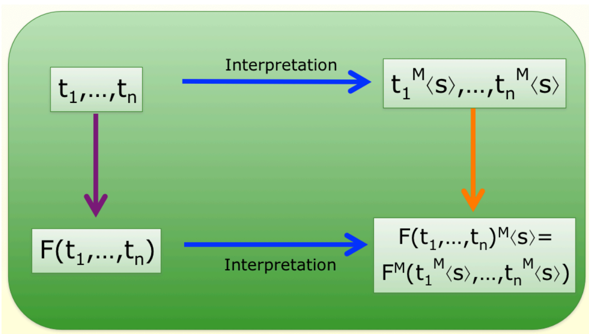


Figure 2.27: The value of a term.

- $y = F(x)$.
- $F(x) = G(y, y)$.
- $G(x, y) = G(F(x), x)$.
- $F(G(x, y)) = G(F(x), F(y))$.

2.20.2 Values of the new terms

Like the old terms x_n and c, the new terms t have a value $t^{\mathcal{M}}\langle s\rangle$ in a structure under any assignment.

- $x^{\mathcal{M}}\langle s\rangle = s(x)$
- $c^{\mathcal{M}}\langle s\rangle = c^{\mathcal{M}}$
- $F(t_1, \ldots, t_n)^{\mathcal{M}}\langle s\rangle = F^{\mathcal{M}}(t_1^{\mathcal{M}}\langle s\rangle, \ldots, t_n^{\mathcal{M}}\langle s\rangle)$

In the third case, when we define the value of $F(t_1, \ldots, t_n)^{\mathcal{M}}\langle s\rangle$, we first determine the values of the arguments $t_1, \ldots, t_n$, obtaining $t_1^{\mathcal{M}}\langle s\rangle, \ldots, t_n^{\mathcal{M}}\langle s\rangle$, and then we plug these values to the function $F^{\mathcal{M}}$ interpreting the function symbol F in $\mathcal{M}$ (see Picture 2.27).

Definition 2.20 *Satisfaction for the new identities (equations) is defined as follows:* $\mathcal{M} \models_s t = t' \iff t^{\mathcal{M}}\langle s\rangle = t'^{\mathcal{M}}\langle s\rangle$.

For example:

$$\mathcal{M} \models_s F(x) = G(F(x))$$

$$\iff F^{\mathcal{M}}(s(x)) = G^{\mathcal{M}}(F^{\mathcal{M}}(s(x))).$$

2.20.3 The ring of integers

Consider the structure $(\mathbb{Z}, +, \cdot)$ the universe of which consists of positive and negative integers endowed with two functions, the addition and the multiplication of integers. This is arguably the most important structure of mathematics! It is called a ring in algebra. One often adds here constants 0 and 1.

Suppose $n > 2$. Are there integers x,y and z (all > 0) such that $x^n + y^n = z^n$? No! (This is called Fermat's Last Theorem.)

2.20.4 The successor function on natural numbers

The structure $(\mathbb{N}, S)$ of natural numbers with the successor function $S(n) = n + 1$ is arguably the most *fundamental* structure in mathematics. One often adds 0 as a constant.

Important properties of this structure are:

- $S(n) = S(m) \to n = m$.
- If $n > 0$, there is m such that $S(m) = n$.

This structure satisfies also the so called Induction Schema: Suppose A is a formula with constant 0 and successor function S. Then

$$(A(0/x_0) \wedge \forall x_0(A \to A(S(x_0)/x_0))) \to \forall x_0 A$$

is an Induction Axiom. The Induction Schema is the collection of all Induction Axioms.

2.20.5 The effect of terms in deductions

Terms with function symbols impose new restrictions to substitution and to the quantifier rules. The new restrictions arise naturally from the fact that a term may contain many variables and we have to pay attention to all of them. Without function symbols a term, which has variables, may only be a lone variable and we already paid attention to this lone variable - easier!

A term t is free for a variable x in a formula A if no variable y occurring in t becomes a bound occurrence of y after t is substituted to free occurrences of x in A. Notation $A(t/x)$ is used only when t is free for x in A.

We can always change A to a logically equivalent formula A', by changing bound variables, such that t is free for x in A.

Note that $F(x, y)$ is not free for x in $\forall y(R(x, y) \to P(y))$, but it is free for x in the logically equivalent formula $\forall z(R(x, z) \to P(z))$.

2.20.6 Quantifier rules in deduction and semantic trees

Natural deduction

When using the elimination of universal quantifier and introduction of existential quantifier rules in natural deduction one has to obey the restriction that the term t that is substituted for a variable x must be free for x in the formula A in question. Since, after introducing function symbols, we have more terms, we have to be more careful with this rule.

Semantic trees

When using semantic trees to prove things in predicate logic with function symbols we have to take all the terms into account, also the terms containing function symbols:

- Universal:

$$\begin{array}{c} \forall x A \\ | \\ A(t/x) \end{array} \qquad \begin{array}{c} \neg\forall x A \\ | \\ \neg A(d/x) \end{array}$$

  In the first case t is any term on the branch leading[3] to $\forall x A$, but t has to be free for x in A. In the second case d is a new constant.

- Existential:

$$\begin{array}{c} \exists x A \\ | \\ A(d/x) \end{array} \qquad \begin{array}{c} \neg\exists x A \\ | \\ \neg A(t/x) \end{array}$$

  In the first case d is a new constant. In the second case t is any term on the branch leading to $\forall x A$, but t has to be free for x in A.

[3] It is not wrong to choose a term t **not** occurring on the branch, but it will not help either, since this rule is used repeatedly, always whenever new terms occur.

Example

Let us give a semantic proof of $\forall x P(x) \to P(F(c))$?

$$\begin{array}{c} \neg(\forall x P(x) \to P(F(c))) \\ | \\ \forall x P(x) \\ | \\ \neg P(F(c)) \\ | \\ P(F(c)) \end{array}$$

The only branch of this tree closes, so this is a semantic proof of the given sentence.

2.20.7 Identity axioms for function symbols

Function symbols call also for new identity axioms. We want to say that for any function, if the arguments are identical, then so are the values.

New Identity Axiom:

I7 $\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n((x_1 = y_1 \wedge \ldots \wedge x_n = y_n) \to F_i^n(x_1, \ldots, x_n) = F_i^n(y_1, \ldots, y_n))$

2.20.8 Solved problems

Problem 345 *Consider the natural numbers with the successor function $S(n) = n + 1$ and constant 0. Prove*

$$\forall x_0(\neg x_0 = 0 \to \exists x_1(x_0 = S(x_1)))$$

by using the Induction Schema.

Solution: Let A be the formula $\neg x_0 = 0 \to \exists x_1(x_0 = S(x_1))$. Let us first prove $A(0/x_0)$ i.e. $\neg 0 = 0 \to \exists x_1(0 = S(x_1))$. This follows from the Identity Axiom $\forall x(x = x)$ easily. Let us then assume A and derive $A(S(x_0)/x_0)$ i.e. $\neg S(x_0) = 0 \to \exists x_1(S(x_0) = S(x_1))$. But this is easy even without assuming A. Thus we have proved $\forall x_0(\neg x_0 = 0 \to \exists x_1(x_0 = S(x_1))$. See Figure 2.28 for the natural deduction based on the above argumentation.
□

Problem 346 *Prove the validity of the following stronger form of the Identity Axiom I7:*

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x\, x=x}{0=0}\,\forall\mathbf{E} \qquad [\neg 0=0]}{0=0\wedge\neg 0=0}\,\wedge\mathbf{I}}{\neg\neg\exists x_1(0=S(x_1))}\,\neg\mathbf{I}}{\exists x_1(0=S(x_1))}\,\neg\mathbf{E}}{\neg 0=0\to\exists x_1(0=S(x_1))}\,\to\mathbf{I}}{A(0/x_0)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x(x=x)}{S(x_0)=S(x_0)}\,\forall\mathbf{E}}{\exists x_1(S(x_0)=S(x_1))}\,\exists\mathbf{I}}{A(S(x_0)/x_0)}}{A\to A(S(x_0)/x_0)}\,\to\mathbf{I}}{\forall x_0(A\to A(S(x_0)/x_0))}\,\forall\mathbf{I}}{A(0/x_0)\wedge\forall x_0(A\to A(S(x_0)/x_0))}\,\wedge\mathbf{I} \qquad B}{\forall x_0(\neg x_0=0\to\exists x_1(x_0=S(x_1)))}\,\to\mathbf{E}
$$

In this deduction B is the Induction Axiom $(A(0/x_0)\wedge\forall x_0(A\to A(S(x_0)/x_0)))\to\forall x_0 A$

Figure 2.28: An example of the use of induction.

$$\forall x_1\ldots\forall x_n\forall y_1\ldots\forall y_n((x_1=y_1\wedge\ldots\wedge x_n=y_n)$$
$$\to t=t'),$$

where t' is obtained from t by replacing x_1 by y_1, x_2 by $y_2,\ldots,x_n$ by y_n.

Solution: If t is a constant, then t' is t and the claim is trivial. If t is a variable other than $x_1,\ldots,x_n$, then again t' is t and the claim follows. If t is the variable x_i, then t' is y_i, and the claim is essentially of the form $x_i=y_i\to x_i=y_i$, and therefore trivial. We are left with the case: t is $F(t_1,\ldots,t_m)$.

Note that $t'=F(t'_1,\ldots,t'_m)$ where t'_i is obtained from t_i by replacing x_1 by y_1, x_2 by $y_2,\ldots,x_n$ by y_n ($i=1,\ldots,m$). We assume as an Induction Hypothesis that

$$\forall x_1\ldots\forall x_n\forall y_1\ldots\forall y_n((x_1=y_1\wedge\ldots\wedge x_n=y_n)$$
$$\to t_i=t'_i)$$

for all $i=1,\ldots,m$. Then we prove the validity of

$$\forall x_1\ldots\forall x_n\forall y_1\ldots\forall y_n((x_1=y_1\wedge\ldots\wedge x_n=y_n)$$
$$\to F(t_1,\ldots,t_m)=F(t'_1,\ldots,t'_m)).$$

Let a model $\mathcal{M}$ and an assignment s be given. Suppose $a_1,\ldots,a_n,b_1,\ldots,b_n$ are such that

$$\mathcal{M}\models_{s'}(x_1=y_1\wedge\ldots\wedge x_n=y_n)$$

for

$$s'=s(a_1\ldots a_n b_1\ldots b_n/x_1\ldots x_n y_1\ldots y_n).$$

Then $a_i=b_i$ for $i=1,\ldots,n$. By induction hypothesis $t_j^{\mathcal{M}}\langle s'\rangle = {t'}_j^{\mathcal{M}}\langle s'\rangle$. Hence $\mathcal{M}\models_{s'} F(t_1,\ldots,t_m)=F(t'_1,\ldots,t'_m)$.

Note: The validity of the strong form of I7 can be also proved by natural deduction or by the method of semantic trees. □

2.20.9 Problems

Problem 347 *Use the method of semantic trees to construct a model for*

$$\forall xR(x,F(x))\vee\neg\forall xR(x,x).$$

Problem 348 *Prove using the semantic tree method: The sentence*

$$\exists x(P(c)\vee P(F(x)))\to(P(c)\vee\exists yP(y))$$

is valid.

Problem 349 *Consider the sentence*

$$\forall x(P(c,d)\to R(x,c,d))$$
$$\to(P(c,d)\to\forall xR(F(x),c,d)).$$

Is the sentence valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 350 *Consider the sentence*

$$(R(c,d) \lor \forall x P(F(x))) \to \forall x(R(c,d) \lor P(x)).$$

Is the sentence valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 351 *Consider the sentence*

$$\forall x P(F(x), x) \lor \exists x \neg P(x, F(x)).$$

Is the sentence valid, contingent or a contradiction? If it is valid or a contradiction, demonstrate this with a natural deduction or a semantic tree (See Section 2.18). If it is contingent, demonstrate this with models obtained by means of semantic trees (See Section 2.17).

Problem 352 *Suppose L is the vocabulary consisting of a constant symbol c and a 1-ary (unary) function symbol F. Suppose $\mathcal{M}$ is an L-structure with universe $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ i.e. the set of integers, positive and negative, $c^{\mathcal{M}} = 0$ and $F^{\mathcal{M}}(a) = a+1$ for all a in $\mathbb{Z}$. Which elements of $\mathbb{Z}$ are values of constant terms (such as e.g. c and $F(c)$) which contain no variables?*

Problem 353 *Suppose L is the vocabulary consisting of a constant symbols c and d and a 2-ary function symbol F. Suppose $\mathcal{M}$ is an L-structure with all the real numbers as the universe $c^{\mathcal{M}} = 0$, $d^{\mathcal{M}} = 1$ and $F^{\mathcal{M}}(a,b) = a+b$ for all a and b. Which elements are values of constant terms (such as e.g. c and $F(c,c)$) which contain no variables?*

Problem 354 *Which of the terms*

1. $F(y)$
2. $F(z)$
3. $F(x)$

are free for the variable y in the following formula $\exists x \forall z(R_0(x,y) \lor \neg \forall y R_0(y,x))$?

Problem 355 *The following "natural deduction" of $\forall y \exists z \forall x R(z,y,x)$ from $\forall y \forall x R(F(x),y,x)$ has an error. What is the error:*

$$\dfrac{\dfrac{\dfrac{\forall y \forall x R(F(x),y,x)}{\forall x R(F(x),y,x)}\ \forall\mathbf{E}}{\exists z \forall x R(z,y,x)}\ \exists\mathbf{I}}{\forall y \exists z \forall x R(z,y,x)}\ \forall\mathbf{I}$$

Problem 356 *Prove by means of the semantic tree method*

$$\forall x \exists y(R(F(x),x) \to R(y,x)).$$

Problem 357 *Give a natural deduction of*

$$\forall x \exists y(R_0(F(x),x) \lor R_1(y,x))$$

from the assumption

$$\forall x \forall y(R_0(y,x) \lor R_1(y,x)).$$

Problem 358 *Prove*

$$\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n((x_1 = y_1 \land \ldots \land x_n = y_n) \to t = t'),$$

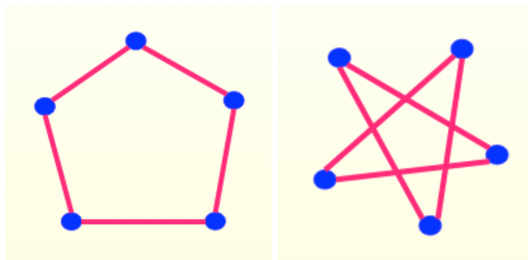
where t' is obtained from t by replacing everywhere x_i by y_i. Use the Identity Axioms I1-I7 and natural deduction or, alternatively, a semantic proof.

2.21 Isomorphism

In logic the slogan is: "Structure is everything—elements are nothing." This is as in chess: whether the pieces are made of marble or plastic makes no difference to the game itself. It also hardly makes sense to ask what natural numbers $0, 1, 2, \ldots$ are "made of". What matters is that 0 is the smallest natural number, 1 is the next, then comes 2, etc.

Isomorphism captures this phenomenon of focusing on structure rather than elements.

A related fact is that we may draw a picture of a structure in many (isomorphic) ways. For example, a graph may have two completely different (but isomorphic) appearances:



The below figure demonstrates an example of an isomorphism between the above two graphs:

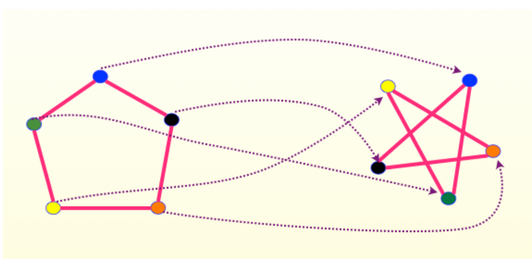


Figure 2.29: Isomorphism

In Figure 2.30 there are four isomorphic unary structures and an indication what the isomorphisms are like.

2.21.1 Exact definition for graphs

We have not defined exactly what is it that we call an isomorphism. We now present an exact definition for graphs. Later we shall treat other structures.

Definition 2.21 *Let L be the vocabulary $\{E\}$ of graphs. Let $\mathcal{M}$ and $\mathcal{M}'$ be two graphs. We say that a mapping f is an isomorphism $\mathcal{M} \to \mathcal{M}'$ if*

ISO1 *f maps elements of the universe of $\mathcal{M}$ to elements of the universe of $\mathcal{M}'$.*

ISO2 *Every element of the universe of $\mathcal{M}'$ is the image of exactly one element of the universe of $\mathcal{M}$.*

ISO3 *If a and b are in the universe of $\mathcal{M}$, then $aE^{\mathcal{M}}b$ if and only if $f(a)E^{\mathcal{M}'}f(b)$.*

Condition ISO2 says, in other words, that f is a bijection $M \to M'$. Figure 2.29 is an example of a bijection satisfying ISO1—ISO3.

We now show that the same sentences are true in isomorphic graphs. Since truth of a sentence is defined in terms of the concept of an assignment satisfying a formula, we use induction on formulas, rather than induction on sentences.

We introduce the concept of conjugacy as a helpful auxiliary concept, in order that the proof that is coming goes through smoothly.

Definition 2.22 *Suppose f is an isomorphism from a graph $\mathcal{M}$ to a graph $\mathcal{M}'$. Suppose s is an assignment for $\mathcal{M}$ and s' an assignment for $\mathcal{M}'$. Then s and s' are conjugates with respect to f if for all variables x $s'(x) = f(s(x))$. See Figure 2.31.*

Proposition 2.23 *If s and s' are conjugate, then for all formulas A:*

$$\mathcal{M} \models_s A \text{ if and only if } \mathcal{M}' \models_{s'} A.$$

Proof:
Case 1: A is an equation $x = y$. $\mathcal{M} \models_s A$ implies $s(x) = s(y)$, which implies $f(s(x)) = f(s(y))$ by ISO1, and this implies $s'(x) = s'(y)$ by conjugacy, which finally implies $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies $s'(x) = s'(y)$, which implies $f(s(x)) = f(s(y))$ by conjugacy, and this implies $s(x) = s(y)$ by condition ISO2, which finally implies $\mathcal{M} \models_s A$.
Case 2: A is an atomic formula xEy. $\mathcal{M} \models_s A$ implies $s(x)E^{\mathcal{M}}s(y)$, which implies $f(s(x))E^{\mathcal{M}'}f(s(y))$ by ISO3, and this implies $s'(x)E^{\mathcal{M}'}s'(y)$ by conjugacy, which finally implies $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies $s'(x)E^{\mathcal{M}'}s'(y)$, which implies $f(s(x))E^{\mathcal{M}'}f(s(y))$ by conjugacy, and this implies $s(x)E^{\mathcal{M}}s(y)$ by condition ISO3, which finally implies $\mathcal{M} \models_s A$.
Case 3: A is $\neg B$ and the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). $\mathcal{M} \models_s A$ implies $\mathcal{M} \not\models_s B$, which implies $M' \not\models_{s'} B$, by Induction Hypothesis, and this finally gives $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies $\mathcal{M}' \not\models_{s'} B$, which implies $M \not\models_s B$, by Induction Hypothesis, and this finally gives $\mathcal{M} \models_s A$.

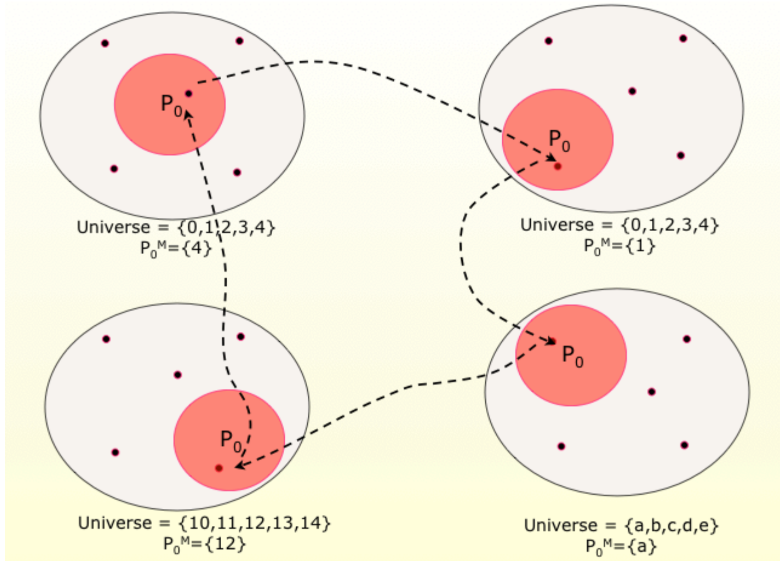


Figure 2.30: Four isomorphic unary structures.

Case 4: A is $B \wedge C$, $B \vee C$, $B \rightarrow C$ or $B \leftrightarrow C$. We assume as the Induction Hypothesis that the claim has already been proved for B and C and for all conjugate s and s'. (Exercise)

Case 5: A is $\exists x B$. Suppose the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). Suppose first $\mathcal{M} \models_s A$. This implies $\mathcal{M} \models_{s(a/x)} B$ for some a. Note that $s(a/x)$ and $s'(f(a)/x)$ are conjugate! By Induction Hypothesis, $\mathcal{M}' \models_{s'(f(a)/x)} B$. Thus $\mathcal{M}' \models_{s'(b/x)} B$ for some b. This implies $\mathcal{M}' \models_{s'} A$.

Suppose then $\mathcal{M}' \models_{s'} A$. This implies $\mathcal{M}' \models_{s'(b/x)} B$ for some b. By ISO2 there is a such that $f(a) = b$. Note that $s(a/x)$ and $s'(b/x)$ are conjugate! By Induction Hypothesis, $\mathcal{M} \models_{s(a/x)} B$. Thus $\mathcal{M} \models_{s(a/x)} B$ for some a. This implies $\mathcal{M} \models_s A$.

Case 6: A is $\forall x B$. Assume as the Induction Hypothesis that the claim has already been proved for B and for all conjugate s and s'. (Exercise)

We have proved that $\mathcal{M} \models_s A$ if and only if $\mathcal{M}' \models_{s'} A$ when $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic graphs and s and s' are conjugate. □

When we assume that A is a sentence we can drop the assignment and conclude $\mathcal{M} \models A$ if and only if $\mathcal{M}' \models A$. In particular, one cannot separate isomorphic graphs by means of a sentence of predicate logic.

We cannot say anything about the vertices of a graph, except whether they are identical or not, and whether they are neighbors or not. Suppose for example the universe of $\mathcal{M}$ is $\{0, 1, 2, 3, 4\}$ and $E^{\mathcal{M}} = \{(0, 1)\}$. Suppose also the universe of $\mathcal{M}'$ is $\{10, 11, 12, 13, 14\}$ and $E^{\mathcal{M}'} = \{(12, 13)\}$. No sentence A can "say" that the vertices of $\mathcal{M}$ are different from the vertices of $\mathcal{M}'$, or that the only edge is in a "different" place in $\mathcal{M}$ than in $\mathcal{M}'$. To be more exact, the statement "the only edge is in a different place in $\mathcal{M}$ than in $\mathcal{M}'$" is meaningless.

All the above can be done in an arbitrary vocabulary—it is not in any way restricted to graphs.

2.21.2 Solved problems

Problem 359 *Which of the following graphs are isomorphic?*

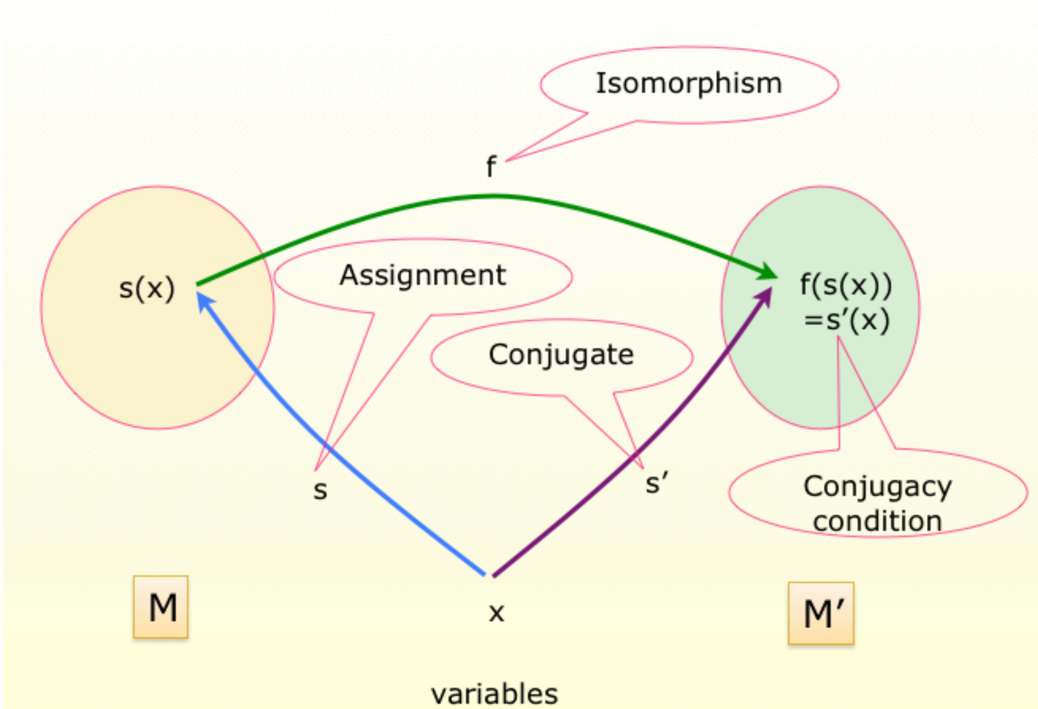
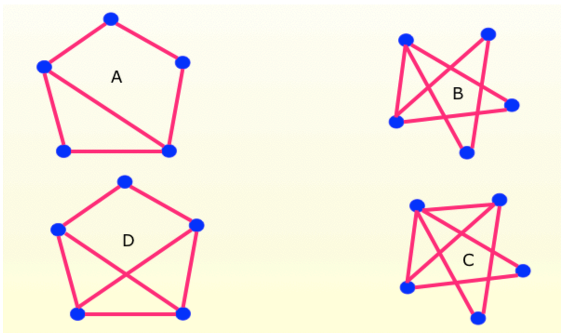
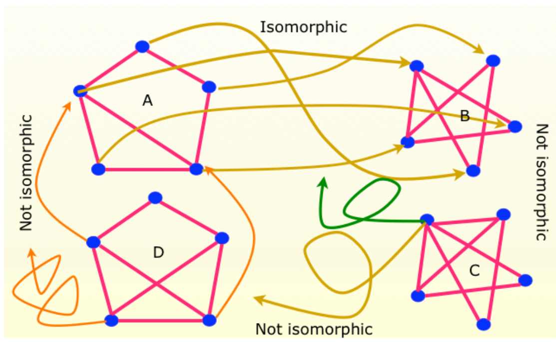


Figure 2.31: Conjugacy

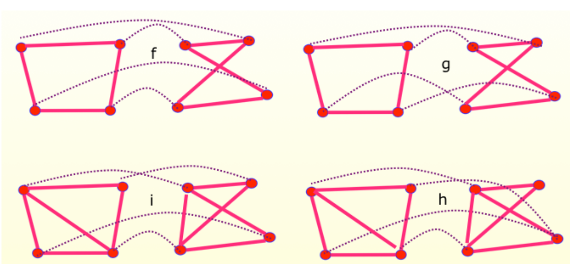


Solution: The top two are isomorphic, the rest are not, as the below picture demonstrates.



□

Problem 360 *Which of the functions is an isomorphism*



Solution: Function f is not an isomorphism because it maps the end points of the leftmost edge to vertices which are not neighbors. Function g is an isomorphism which one can see by going through all the edges. Function h is not an isomorphism because one vertices in the right hand side graph is the image of two vertices from the left. Finally, function i is an isomorphism which one can see by going through all the edges. □

Problem 361 *Prove that isomorphism preserves truth in unary structures, using the following definition: Let L be the vocabulary $\{P_1, \ldots, P_n\}$, where each P_i is unary. Let $\mathcal{M}$ and $\mathcal{M}'$ be two L-structures. We say that a mapping f is an* **isomorphism** *$\mathcal{M} \to \mathcal{M}'$ if*

ISO1 *f maps elements of the universe of $\mathcal{M}$ to elements of the universe of $\mathcal{M}'$.*

ISO2 *Every element of the universe of $\mathcal{M}'$ is the image of exactly one element of the universe of $\mathcal{M}$.*

ISO3 *If a is in the universe of $\mathcal{M}$, then $P_i^{\mathcal{M}}(a)$ if and only if $P_i^{\mathcal{M}'}(f(a))$.*

Solution: We should show that the same sentences are true in isomorphic unary structures. Since truth of a sentence is defined in terms of the concept of an assignment satisfying a formula, we use induction on formulas. The concept of conjugacy turns out to be again useful.

We prove: If $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic and s and s' are conjugate, then for all formulas A: $\mathcal{M} \models_s A$ if and only if $\mathcal{M}' \models_{s'} A$.

Case 1: A is an equation $x = y$. As before: $\mathcal{M} \models_s A$ implies $s(x) = s(y)$, which implies $f(s(x)) = f(s(y))$ by ISO1, and this implies $s'(x) = s'(y)$ by conjugacy, which finally implies $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies $s'(x) = s'(y)$, which implies $f(s(x)) = f(s(y))$ by conjugacy, and this implies $s(x) = s(y)$ by condition ISO2, which finally implies $\mathcal{M} \models_s A$.

Case 2: A is an atomic formula $P_i(x)$. $\mathcal{M} \models_s A$ implies $s(x) \in P_i^M$, which implies $f(s(x)) \in P_i^{\mathcal{M}'}$ by ISO3, and this implies $s'(x) \in P_i^{\mathcal{M}'}$ by conjugacy, which finally implies $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies $s'(x) \in P_i^{\mathcal{M}'}$, which implies $f(s(x)) \in P_i^{\mathcal{M}'}$ by conjugacy, and this implies $s(x) \in P_i^{\mathcal{M}}$ by condition ISO3, which finally implies $\mathcal{M} \models_s A$.

Case 3: A is $\neg B$ and the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). As before: $\mathcal{M} \models_s A$ implies $M \not\models_s B$, which implies $\mathcal{M}' \not\models_{s'} B$, by Induction Hypothesis, and this finally gives $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M} \models_{s'} A$ implies $\mathcal{M}' \not\models_{s'} B$, which implies $\mathcal{M} \not\models_s B$, by Induction Hypothesis, and this finally gives $\mathcal{M} \models_s A$.

Case 4: A is $B \wedge C$, $B \vee C$, $B \to C$ or $B \leftrightarrow C$. We assume as the Induction Hypothesis that the claim has already been proved for B and C and for all conjugate s and s'. (Exercise)

Case 5: A is $\exists x B$. Suppose the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). As before: Suppose first $\mathcal{M} \models_s A$. This implies $\mathcal{M} \models_{s(a/x)} B$ for some a. Note that $s(a/x)$ and $s'(f(a)/x)$ are conjugate! By Induction Hypothesis, $\mathcal{M}' \models_{s'(f(a)/x)} B$. Thus $\mathcal{M}' \models_{s'(b/x)} B$ for some b. This implies $\mathcal{M}' \models_{s'} A$.

Suppose then $\mathcal{M}' \models_{s'} A$. This implies $\mathcal{M}' \models_{s'(b/x)} B$ for some b. By ISO2 there is a such that $f(a) = b$. Note that $s(a/x)$ and $s'(b/x)$ are conjugate! By Induction Hypothesis, $\mathcal{M} \models_{s(a/x)} B$. Thus $\mathcal{M} \models_{s(a/x)} B$ for some a. This implies $\mathcal{M} \models_s A$.

Case 6: A is $\forall x B$. Suppose the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). Exercise.

We have proved that $\mathcal{M} \models_s A$ if and only if $\mathcal{M}' \models_{s'} A$ when $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic unary structures and s and s' are conjugate. When we assume that A is a sentence we can drop the assignment and conclude $\mathcal{M} \models A$ if and only if $\mathcal{M}' \models A$.

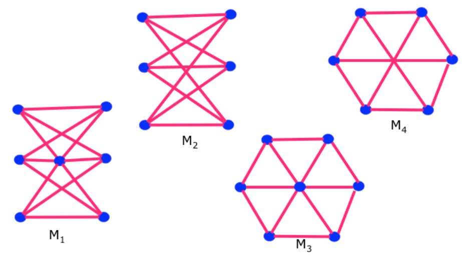
In particular, one cannot separate isomorphic unary structures by means of a sentence of predicate logic.

If the vocabulary L is $\{P\}$, where P is a unary predicate symbol, we cannot say with a sentence which elements are in P, only how many they are, at least if they are only finitely many. This can be seen as follow. Suppose the universe of $\mathcal{M}$ is $\{0, 1, 2, 3, 4\}$, $P^{\mathcal{M}} = \{0\}$, and the universe of $\mathcal{M}'$ is $\{0, 1, 2, 3, 4\}$, $P^{\mathcal{M}} = \{1\}$. Now $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic. No sentence A "says" that there is a difference between $\mathcal{M}$ and $\mathcal{M}'$.

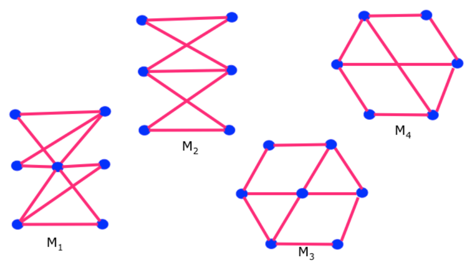
□

2.21.3 Problems

Problem 362 *Which of the below graphs are isomorphic and which are not?*



Problem 363 *Problem: Which of the below graphs are isomorphic and which are not?*



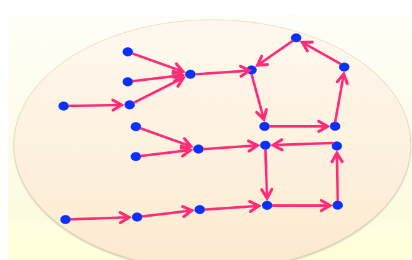
Problem 364 *Prove that if the graph $\mathcal{M}$ is isomorphic to the graph $\mathcal{M}'$ and $\mathcal{M}'$ is isomorphic to the graph $\mathcal{M}''$, then $\mathcal{M}$ and $\mathcal{M}''$ are isomorphic.*

Problem 365 *Prove that if a graph $\mathcal{M}$ is isomorphic to a structure $\mathcal{M}'$ then $\mathcal{M}'$ is also a graph.*

Problem 366 *Prove that every graph with 10 vertices is isomorphic to a graph with $\{1, 2, \ldots, 10\}$ as the universe.*

2.22 More about isomorphism

Up to now we have discussed the concept of isomorphism only in the case of graphs and in the case of unary predicates. Let us now look at the situation where we have a vocabulary consisting of one unary function. Here is a picture of a structure with a unary function:



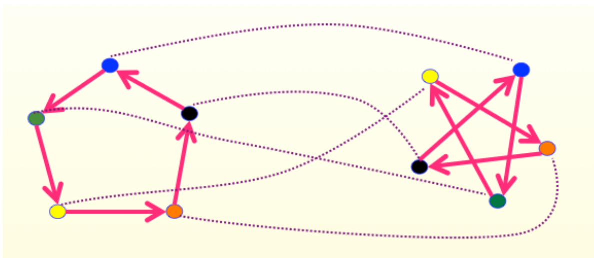
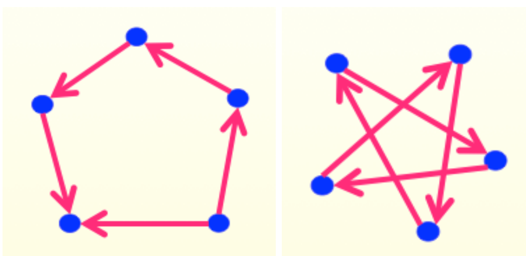


Figure 2.32: Isomorphism of unary functions.

As with graphs, one and the same unary function may have different appearances:



Picture 2.32 shows that these two renditions of a unary function represent indeed isomorphic unary functions.

2.22.1 Exact definition for structures with functions

After the preliminary discussion on isomorphism of unary structures, let us try to give an exact definition:

Definition 2.24 *Let L be the vocabulary $\{G\}$, where G is a unary function symbol. Let $\mathcal{M}$ and $\mathcal{M}'$ be two L-structures. We say that a mapping f is an isomorphism $\mathcal{M} \to \mathcal{M}'$ if*

ISO1 *f maps elements of the universe of $\mathcal{M}$ to elements of the universe of $\mathcal{M}'$.*

ISO2 *Every element of the universe of $\mathcal{M}'$ is the image of exactly one element of the universe of $\mathcal{M}$.*

ISO3 *If a and b are in the universe of $\mathcal{M}$, then $b = G^{\mathcal{M}}(a)$ if and only if $f(b) = G^{\mathcal{M}'}(f(a))$.*

Note: ISO3 says $f(G^{\mathcal{M}}(a)) = G^{\mathcal{M}'}(f(a))$ for all a.

An example of an isomorphism of structures with a unary function was in Figure 2.32.

2.22.2 Isomorphism preserves truth

We show that the same sentences are true in isomorphic L-structures, also in the case of unary functions. Since truth of a sentence is defined in terms of the concept of an assignment satisfying a formula, we use induction on formulas.

Recall the concept of conjugacy of assignments from Definition 2.22 and Figure 2.31: Suppose f is an isomorphism from $\mathcal{M}$ to $\mathcal{M}'$. Suppose s is an assignment for $\mathcal{M}$ and s' an assignment for $\mathcal{M}'$. Then s and s' are conjugates with respect to f if for all variables x, $s'(x) = f(s(x))$.

We prove first a preliminary Lemma:

Lemma 2.25 *If s and s' are conjugate, then for all terms t:*

$$f(t^{\mathcal{M}}\langle s \rangle) = t^{\mathcal{M}'}\langle s' \rangle.$$

Proof:
Case 1: t is a constant c. This is not possible because our vocabulary L does not contain constant symbols.
Case 2: t is a variable x. Now by conjugacy

$$f(t^{\mathcal{M}}\langle s \rangle) = f(x^{\mathcal{M}}\langle s \rangle)$$

$$= f(s(x)) = s'(x) = t^{\mathcal{M}'}\langle s' \rangle.$$

Case 3: t is $G(t')$. We assume

$$f(t'^{\mathcal{M}}\langle s \rangle) = t'^{\mathcal{M}'}\langle s' \rangle$$

as the Induction Hypothesis. Now by ISO3

$$f(t^{\mathcal{M}}\langle s \rangle) = f(G(t')^{\mathcal{M}}\langle s \rangle)$$

$$= f(G^{\mathcal{M}}(t'^{\mathcal{M}}\langle s \rangle)) = G^{\mathcal{M}'}(f(t'^{\mathcal{M}}\langle s \rangle)) = G^{\mathcal{M}'}(t'^{\mathcal{M}'}\langle s' \rangle)$$

$$= G(t')^{\mathcal{M}'}\langle s' \rangle = t^{\mathcal{M}'}\langle s' \rangle.$$

□

Now the preservation of truth:

Proposition 2.26 *If s and s' are conjugate, then for all formulas A: $\mathcal{M} \models_s A$ if and only if $\mathcal{M}' \models_{s'} A$.*

Proof: Case 1: A is an equation $t = t'$. $\mathcal{M} \models_s A$ implies

$$t^{\mathcal{M}}\langle s\rangle = t'^{\mathcal{M}}\langle s\rangle,$$

which implies

$$f(t^{\mathcal{M}}\langle s\rangle) = f(t'^{\mathcal{M}}\langle s\rangle)$$

by ISO1, and this implies

$$t^{\mathcal{M}'}\langle s'\rangle = t'^{\mathcal{M}'}\langle s'\rangle$$

by the Lemma 2.25, which finally implies $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies

$$t^{\mathcal{M}'}\langle s'\rangle = t'^{\mathcal{M}'}\langle s'\rangle,$$

which implies

$$f(t^{\mathcal{M}}\langle s\rangle) = f(t'^{\mathcal{M}}\langle s\rangle)$$

by the Lemma 2.25, and this implies

$$t^{\mathcal{M}}\langle s\rangle = t'^{\mathcal{M}}\langle s\rangle$$

by condition ISO2, which finally implies $\mathcal{M} \models_s A$.

Case 2: A is $\neg B$ and the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). As before: $\mathcal{M} \models_s A$ implies $M \not\models_s B$, which implies $M' \not\models_{s'} B$, by Induction Hypothesis, and this finally gives $\mathcal{M}' \models_{s'} A$. Conversely, $\mathcal{M}' \models_{s'} A$ implies $\mathcal{M}' \not\models_{s'} B$, which implies $M \not\models_s B$, by Induction Hypothesis, and this finally gives $\mathcal{M} \models_s A$.

Case 3: Suppose the claim has already been proved for B and C and for all conjugate s and s' (Induction Hypothesis). Then the claim holds for $B \wedge C$, $B \vee C$, $B \to C$ and for $B \leftrightarrow C$. (Exercise)

Case 4: A is $\exists x B$. Suppose the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis).

As before: Suppose first $\mathcal{M} \models_s A$. This implies $\mathcal{M} \models_{s(a/x)} B$ for some a. Note that $s(a/x)$ and $s'(f(a)/x)$ are conjugate! By Induction Hypothesis, $\mathcal{M}' \models_{s'(f(a)/x)} B$. Thus $\mathcal{M}' \models_{s'(b/x)} B$ for some b. This implies $\mathcal{M}' \models_{s'} A$.

Suppose then $\mathcal{M}' \models_{s'} A$. This implies $\mathcal{M}' \models_{s'(b/x)} B$ for some b. By ISO2 there is a such that $f(a) = b$. Note that $s(a/x)$ and $s'(b/x)$ are conjugate! By Induction Hypothesis, $\mathcal{M} \models_{s(a/x)} B$. Thus $\mathcal{M} \models_{s(a/x)} B$ for some a. This implies $\mathcal{M} \models_s A$.

Case 5: A is $\forall x B$. Suppose the claim has already been proved for B and for all conjugate s and s' (Induction Hypothesis). Exercise.

□

We have proved that $\mathcal{M} \models_s A$ if and only if $\mathcal{M}' \models_{s'} A$ when $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic L-structures and s and s' are conjugate. When we assume that A is a sentence we can drop the assignment and conclude $M \models A$ if and only if $\mathcal{M}' \models A$.

In particular, one cannot separate isomorphic L-structures by means of a sentence of predicate logic.

2.22.3 Example

We cannot say anything about the elements of L-structures, $L = \{G\}$, except what their mutual relationships are in terms of the one function. Consider the models:

$\mathcal{M}$: Universe of $\mathcal{M}$ is $\{0, 1, 2, 3, 4\}$, $G^{\mathcal{M}}(a) = 0$ for all a.

$\mathcal{M}'$: Universe of $\mathcal{M}'$ is $\{0, 1, 2, 3, 4\}$, $G^{\mathcal{M}'}(a) = 1$ for all a.

No sentence A can "say" that the constant functions $\mathcal{G}^{\mathcal{M}}$ and $\mathcal{G}^{\mathcal{M}'}$ are different, because the function $f(0) = 1, f(1) = 0, f(a) = a$ if $a \in \{2, 3, 4\}$, is an isomorphism $\mathcal{M} \to \mathcal{M}'$.

2.22.4 Now the general case

All the above can be done in an arbitrary vocabulary - it is not in any way restricted to graphs, unary predicates or unary functions. The exact definition of isomorphism in the general case is as follows:

Definition 2.27 *Let L be an arbitrary vocabulary. Let $\mathcal{M}$ and $\mathcal{M}'$ be two L-structures. We say that a mapping f is an isomorphism from $\mathcal{M}$ to $\mathcal{M}'$ if*

ISO1 *f maps elements of the universe of $\mathcal{M}$ to elements of the universe of $\mathcal{M}'$.*

ISO2 *Every element of the universe of $\mathcal{M}'$ is the image of exactly one element of the universe of $\mathcal{M}$.*

ISO3 *There are three cases:*

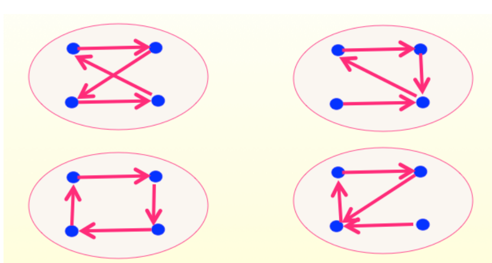
1. *If c is a constant symbol in L, then $f(c^{\mathcal{M}}) = c^{\mathcal{M}'}$.*
2. *If R_i^n is in L, then $(a_1, \ldots, a_n) \in (R_i^n)^{\mathcal{M}}$ iff $(f(a_1), \ldots, f(a_n)) \in (R_i^n)^{\mathcal{M}'}$ for all $a_1, \ldots, a_n$ in the universe of $\mathcal{M}$.*
3. *If F_i^n is in L then $f((F_i^n)^{\mathcal{M}}(a_1, \ldots, a_n)) = (F_i^n)^{\mathcal{M}'}(f(a_1), \ldots, f(a_n))$ for all $a_1, \ldots, a_n$ in the universe of $\mathcal{M}$.*

2.22.5 Ordered sets

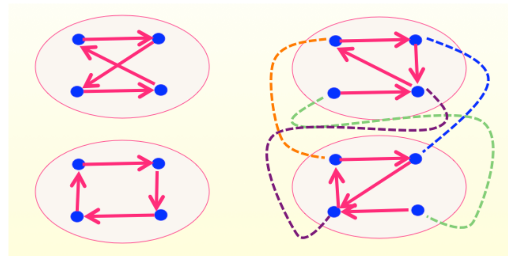
Let us consider the vocabulary $L = \{<\}$. L-structures which satisfy the axioms of order are called ordered sets. Finite ordered sets with the same number of elements are isomorphic. One maps the first element of Infinite ordered sets need not be isomorphic. The following four ordered sets are all non-isomorphic to each other: Integers, positive integers, rationals, reals.

2.22.6 Solved problems

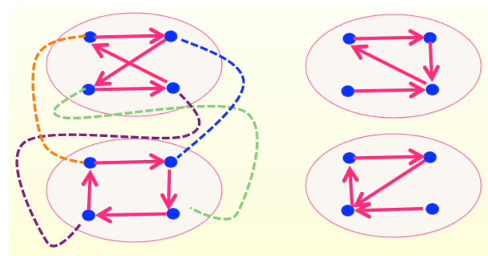
Problem 367 *Which of the following structures with a unary function are isomorphic? Which sentences separates the non-isomorphic ones?*



Solution: The two structures on the right are isomorphic:



Likewise, the two structures on the left are isomorphic:



Let us then think what would be a sentence that separates the left-hand structures from the right-hand ones, making it impossible for all the structures to be isomorphic. The point is that on the left every element is the image of one elements only while on the right there are elements that are images of two different element. Thus a sentence which is true in both structures on the right but in neither structure on the left is:

$$\exists x \exists y (F(x) = F(y) \land \neg x = y).$$

□

Problem 368 *Which of the following structures with a unary function are isomorphic. Which sentences separates the non-isomorphic ones?*

$\mathcal{M}$: *The natural numbers with the function $g(a) = a$, if a even and $g(a) = 1$ otherwise.*

$\mathcal{M}'$: *The natural numbers with the function $g'(a) = 0$, if a even and $g'(a) = a$ otherwise.*

$\mathcal{M}''$: *The natural numbers with the function $g''(a) = 0$, if a even and $g''(a) = 1$ otherwise.*

Solution: The first two models are isomorphic as the function $f(2n) = 2n+1$, $f(2n+1) = 2n$, shows. Let us check this: Suppose a is a natural number. Then $a = 2n$ or $a = 2n+1$ for some n. In the first case we obtain:

$$f(G(a)^{\mathcal{M}}) = f(G(2n)^{\mathcal{M}}) = f(g(2n)) =$$

$$= f(2n) = 2n+1$$

and

$$G(f(a))^{\mathcal{M}'} = G(f(2n))^{\mathcal{M}'} = G(2n+1)^{\mathcal{M}'} =$$

$$= g'(2n+1) = 2n+1.$$

On the other hand, if $a = 2n+1$ we obtain:

$$f(G(a)^{\mathcal{M}}) = f(G(2n+1)^{\mathcal{M}}) = f(g(2n+1)) =$$

$$= f(1) = 0$$

and

$$G(f(a))^{\mathcal{M}'} = G(f(2n+1))^{\mathcal{M}'} = G(2n)^{\mathcal{M}'} =$$

$$= g'(2n) = 0.$$

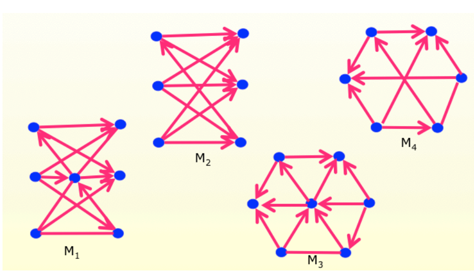
The model $\mathcal{M}''$ is not isomorphic to the first two models. An example of a sentence that separates them is

$$\exists x \exists y \forall z (G(z) = x \lor G(z) = y).$$

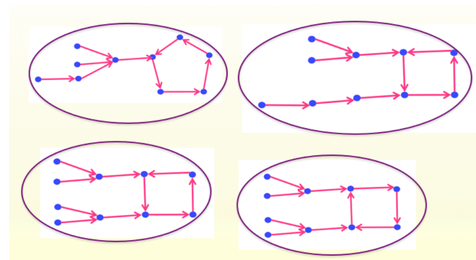
This sentence is clearly true on $\mathcal{M}''$ but false in both $\mathcal{M}$ and $\mathcal{M}'$. □

2.22.7 Problems

Problem 369 *Which of the below structures with a unary function are isomorphic and which are not?*



Problem 370 *Which of the below structures with a unary function are isomorphic and which are not?*



Problem 371 *Suppose $\mathcal{M}$ is a structure and s is an assignment for $\mathcal{M}$. Suppose f is an isomorphism from $\mathcal{M}$ to $\mathcal{M}'$. Show that there is one and only one assignment s' for $\mathcal{M}'$ such that s and s' are conjugates with respect to f.*

Problem 372 *Show that the following ordered sets are all non-isomorphic to each other:*

1. *Integers.*
2. *Positive integers.*
3. *Rational numbers.*
4. *Real numbers.*

Problem 373 *Prove for an arbitrary vocabulary that isomorphic structures satisfy the same sentences.*

Problem 374 *Let the vocabulary L be $\{R\}$, where R is a binary predicate symbol. Consider the L-structure $\mathcal{M}$ whose universe is the set of all rational numbers (i.e. numbers of the form m/n, where m and n are integers (without common divisors) and $n \neq 0$) and in which $R^{\mathcal{M}}$ is the set of pairs (a, b) such that $a < b$. Let $\mathcal{M}'$ be the L-structure whose universe is the set of all rational numbers except 0 and in which $R^{\mathcal{M}}$ is again the set of pairs (a, b) such that $a < b$. Are the structures $\mathcal{M}$ and $\mathcal{M}'$ isomorphic or not? (This problem is a bit harder than the other problems.)*

Index