Alexander Kasiukov

Suffolk County Community College
MAT 101
A Survey of Mathematical Reasoning

Logic — The Boring Chapter 0

January 25, 2025

Contents

1 Preface

1.1 The Purpose of these Notes

These notes were written as a supplement for an introductory course MAT101: *A Survey of Mathematical Reasoning* taught by the author in Fall 2022 at Suffolk County Community College. The bulk of the content was completed in July 2022. Insignificant updates have been made since that time.

1.2 What is Logic?

`Logic` is a discipline of human mind, instructing us how to build

- the vessels that give the proper form to our thoughts, and
- the conduits directing our reasoning towards the truth.

As we climb through this course, we will once in a while look back and re-examine the meaning of logic from progressively higher vantage points. But those snapshots, reflective of the historical development of the subject and distorted by its inevitable ideosyncrasies, will neither give us the ultimate definition, nor reveal the full scope of logic. Instead, each subsequent definition will gave us only an approximation of the true essence of the subject, their transient and incomplete nature calling upon us to further the the field of logic in the quest for its ultimate meaning.

Logic is both a theoretical and an experimental field, and the tension — sometimes catastrophic — between its foundations and applications is the main force shaping its continuing development. Let the general purpuse of finding the truth guide us in our studies, even if we inavitably fall short of that lofty goal.

1.3 What to Expect from this Course

The dualism of logic-as-an-object of study on the one hand and logic-as-a-tool for studying other things on the other, should be a part of any introduction into the subject. Indeed, no tool can be effectively used without at least some basic understanding of its mechanism, and no theory is meaningful until it is put to practice. Thus these notes will have two —

intertwined but distinct — narratives: an instruction on how to build those vessels and conduits for human thoughts, and a demonstration on how the tools we build can facilitate our thinking. For clarity, we will use the word `meta-logic` for the narrative describing logic-as-an-object of study.

Since it is logic itself that gives us the ability to be precise, the meta-logical part of a basic introduction into the subject must inadvertently be informal, intuitive and — as mathematicians often put it — "hand-waiving". Indeed, without becoming circular, such an introduction cannot employ the tools that would allow it to be rigorous. Thus, don't expect anything else from the these notes: when introducing logical concepts, we will aim at achieving *intuitive* clarity without the pretense of rigor and precision[1]. However, as we progress through the material, we will pick the tools along the way that will enable us to be more and more exacting in our discourse.

Finally, in a more advanced course, you will be able to apply the tools we develop here to the study of logic itself, making the study-of-logic-with-logic look like the ouroboros [2] symbol:

Figure 1: The Ouroboros

[1]This introduction should be taken in that spirit as well.

[2]from Ancient Greek οὐροβόρος, combining οὐρά [oura] meaning 'tail' and βορός [boros] — 'eating'

1.4 The Phenomenon of Self-reference

The ouroboros captures the idea of self-reference which is perhaps the most intriguing aspect of logic. No story of logic is complete without this topic.

From the first appearance of Liar's Paradox attributed to Epimenides[3] to the modern work on the foundations of mathematics, self-reference and the resulting paradoxes feed the drama of logical development[4] and make the discipline of logic as interesting as it is. Without the paradoxes causing periodic catastrophes and reshaping of the whole discipline, logic would be perhaps a bit more deep, but no more exciting than a washing machine owner's manual. More importantly, the world we live in would be much more regular and mechanical, and thus less humane. The paradoxes of logic capture the aspect of our existence that elevates the imperfect human mind to the level on which the universe itself operates.

Sadly, what we will have time for in this course is only the introductory basic part of logic that lays the foundations for the study of self-reference without following through on this promise. Thus it can be properly termed "the boring part" of the subject. I hope the glimpses and the shadows of the ouroboros you see — if only superficially — through these notes will inspire you to go further in your studies.

[3] Ἐπιμενίδης [Epimenides of Crete] was a semi-mythical Greek philosopher-poet who supposedly lived sometime around 7th or 6th century BC. The paradox stems from a poem Κρητικά [Cretica], attributed to him and quoted twice in the New Testament (Acts 17:28 and Titus 1:12-13). The relevant verse, "Cretans, always liars,…" appears in the Epistle to Titus, chapter 1, verse 12. There is no evidence Epimenides himself considered the verse paradoxical. The paradox can be rephrased as follows. Epimenides says: "All Cretans always lie". But Epimenides is a Cretan himself. Is his statement true or false?

[4] More on that in the section on the history of this discipline.

2 Propositional Logic

2.1 Ontology: Statements and Logical Gates

`Ontology` describes the *notions* used in a field of study[5]. The first notion of logic which is used to structure its field of study is that of a statement.

Definition (of `statement`): A `statement` expresses the general idea of such everyday concepts as fact, judgment, sentence, claim etc. The most important property of a statement is its `truth` or `falsehood`, referred to as its `truth value`. When determining whether something is or is not a statement, we are not concerned with deciding its truth value. We just need to make sure that the truth value is a property that makes sense when applied to the thing we are considering. ♢

Example (statement): This dog is big. ♢

Example (non-statements):
Is that dog big?
green
Give me that dog! ♢

The above examples also show one important aspect of statements: statements acquire their full meaning from their specific context. We will pay close attention to the subject of context later.

Statements can be constructed from other statements in various ways.

Example (combining statements): Two statements, "This dog is big." and "You should take it outside." can be combined into one: "This dog is big and you should take it outside." ♢

Definition (of `logical gate`): A `logical gate` is a *particular way* of constructing a new statement from one or more other statements, independent from the specifics of the statements being used in the construction. Each gate is defined by the truth value of the resulting statement for each possible

[5]from Ancient Greek ὄντος [óntos] — "being" and λόγος [lógos] – a description; in other words, a description of things that exist.

combination of truth values of the original statements. It is convenient to represent that information in the form of so called "truth tables". ♢

For instance, the "combining statements" example on page 6 used the *conjunction* of two statements in question.

Definition (of `logical gate ‘‘conjunction’’`): The `conjunction` of two statements A and B, denoted $A \wedge B$, is a statement whose truth value is defined by the following table:

A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

In natural language, conjunction is usually expressed by the word "and". For example, "this dog is big *and* friendly" is logically the same as

"(this dog is big) $\wedge$ (this dog is friendly)".

However, many different constructions have the same *logical* meaning, with their differences expressing additional (non-logical) meaning variations, as in "this dog is big *but* friendly". ♢

Example (combining statements using specific gate): Using conjunction, we can rewrite the "combining statements" example on page 6 as:

(This dog is big and you should take it outside.) =
(This dog is big.) $\wedge$ (You should take it outside.)

♢

Definition (of `logical gate ‘‘negation’’`): The negation of a statement A, denoted $\neg A$ is a statement whose truth value is defined by the following table:

A	$\neg A$
T	F
F	T

$\diamondsuit$

Since the list of combinations of truth values of two statements does not depend on the logical gate being considered, we can combine several truth tables into one, with that list occupying the first two columns, and each new logical gate represented by every subsequent column. This is done in the following table, where we define a few more standard logical gates. The labels "Operation Title" and "Possible Meaning" refer to their rows, rather than the first column.

Definition (of `disjunction, XOR, implication, equivalence`):

Operation Title:		disjunction	XOR	implication	equivalence
Possible Meaning:		or	either…or	if…then	if and only if
A	B	$A \vee B$	$A \veebar B$	$A \Rightarrow B$	$A \Leftrightarrow B$
T	T	T	F	T	T
T	F	T	T	F	F
F	T	T	T	T	F
F	F	F	F	T	T

XOR stands for "exclusive OR". $\diamondsuit$

Even though there are other logical gates which are even occasionally useful in modeling natural language, these are the main logical gates of propositional logic.

HOMEWORK: How many logical gates combining two statements are there in the total?

Importantly, negation, conjunction and disjunction are sufficient for constructing all other logical gates, no matter the number of their constituent statements. We will study that question later.

2.1.1 Special Role of Implication — What is Logic?

Accepting certain statements as true may require acceptance of certain other statements as true, purely because of the *structure* of those statements. This idea[6] — the idea of formal `inference` — is the very foundation of logic

[6] Explicitly stated already by Aristotle, and possibly understood even earlier.

itself. Implication is the formal model of the inference relation between statements. Thus, the concept of implication is the center of logic.

Example (inference): Believing that a black cat is hidden in a particular box necessitates believing that a cat (of any color) is in the box. Using implication, we can express this idea as

$$\text{(a black cat is in the box)} \Rightarrow \text{(a cat is in the box)}.$$

◇

Definition (of `argument`): An `argument` is a statement which has the form of an implication. In an argument $A \Rightarrow B$, statement A can be called the `assumption`, the `hypothesis`, the `premise`, or the `antecedent` of the argument; statement B can be called the `consequence`, the `conclusion`, or the `consequent` of the argument. ◇

Definition (of `validity`): An argument is `valid` if and only if it is a true statement, as prescribed by the truth table that defines implication (see page 8).

As we can read from the truth table of implication on page 8, the statement A implies the statement B if and only if whenever A is true, the B must be true as well. (To put it differently, it must be impossible for A to be true and for B to be false.) Thus, an argument is valid if and only if in any circumstances when the assumption A is true, the conclusion B is true as well. ◇

Now we can formulate the notion of logic in more precise terms. Logic is the technology for

- giving our thoughts the right structural form;
- determining their truth value when it can be done based on their structure alone; and
- using the structure of our hypotheses[7] for making valid inferences.

[7] i.e. the statements that we are willing to accept, if only provisionally

This falls short of our ultimate goal — that of finding the truth. Logic — at least in its present form — guarantees neither the truth nor the meaningfulness of what it offers. It is merely the *grammar* of rational thinking that gives our thoughts the structure making them unambiguous and precise, and provides the formal rules of inference. It is the meter and rhyme of rational thinking, enabling it to express the poetry of truth. Logic by itself cannot *give* us the truth, it can merely help us *formulate* our beliefs and *explicate* the truth already contained in our assumptions. We need to venture outside of logic to come up with reasonable assumptions. Logical conclusions are formal and relative — relative to the truth of our assumptions. Absolute truth of the conclusion is the focus of the following concept:

Definition (of `soundness`): An argument is `sound` if and only if it is valid and starts with a true hypothesis. ♢

Definition (of `modus ponens`): Modus ponens (Latin for "method of affirming") is a fundamental principle of logic affirming the truth of the conclusion of a sound argument:

$$\Bigg(\Big(A \Rightarrow C\Big) \wedge A\Bigg) \Rightarrow C.$$

If the assumption A implies the conclusion C (meaning that the argument $A \Rightarrow C$ is valid) and the assumption A is true (meaning, together with the previous, that the argument is sound), then the conclusion C is true. ♢

Before we move on to the next section, let's introduce some additional terminology related to implications.

Definition (of `converse, inverse, counter-positive`): Suppose we have a statement in the form of an implication

$$A \Rightarrow B$$

where A and B are some other statements. Then

- statement $B \Rightarrow A$ is called "the `converse`" of the original;

- statement $(\neg A) \Rightarrow (\neg B)$ is called "the `inverse`" of the original;
- statement $(\neg B) \Rightarrow (\neg A)$ is called "the `counter-positive`" of the original.

$\diamondsuit$

> **HOMEWORK:** Use the truth tables on pages 7, 8 (that define negation, implication and equivalence) to verify the equivalence of
>
> - the original implication and its counter-positive;
> - the converse and the inverse of an implication.

One additional piece of intuition related to implication is expressed in terms of relative "strength" of statements. When an implication $A \Rightarrow B$ holds, one may refer to A as the `stronger`, and to B — as the `weaker` of the two. Similarly, finding a consequence B for a given statement A is called `weakening` the A. Likewise, finding an assumption A from which the statement B follows is called `strenthening` the B.

2.2 Apologia: Truth Tables

`Apologia` is the formal defense of certain position, conduct or actor[8]. In these notes, we will use this term to describe how arguments are *validated* in a particular logical theory.

Propositional logic gives us the tools for only the most basic analysis of reasoning. Such analysis can only give the lowest resolution picture, based on breaking down real life narrative into statements and logical gates. The smallest units in this breakdown process, besides the gates, are the so-called `atomic` statements, namely those that cannot be represented as combinations of smaller statements[9]. Apologia of propositional logic will be based on

[8] from the Greek word απολογία [apologia] coming from από [apo] — "of", and λόγος [logos] — "speech", literally "about the". However, in homage to Ἀπολογία Σωκράτους [Apología Sokratous] written by Plato, it is used as meaning "the defence of".

[9] Even though in the later sections, those "atomic" statements will be — sometimes — broken further into smaller pieces of information, those pieces will not be statements.

propositional analysis, namely breaking an argument into atomic statements and gates and applying the truth tables to the resulting implication formula.

Example (valid argument): This dog is always happy after a meal. However, it seems to be troubled and restless. Probably it is hungry. $\diamondsuit$

Example (invalid argument): This dog is always happy after a meal. If you don't feed it, it will be very angry. $\diamondsuit$

What makes one argument valid and another one invalid? How can we effectively decide these questions in general? This is the subject of this section.

The first step in determining validity of an argument is its *propositional analysis*, namely breaking the whole of the argument as a statement into smaller statements combined together by logical gates.

In natural communications, we rely on the context when omitting implicit assumptions, and use the flexibility of our language to convey shades of meaning and to avoid rigid repetitiveness. These features make our conversations more succinct and lively, but obscure the structure. To "correct" these shortcomings, we need to make all omitted assumptions explicit and adjust the wording — without change in meaning — in order to reveal the argument's ultimate structure. Some authors even introduce additional terminology to stress the latter point, using the term `proposition` to describe the underlying meaning that may be expressed in various ways by different statements.

Take the first example, "This dog is always happy after a meal. However, it seems to be troubled and restless. Probably it is hungry." Consider the whole argument as one statement. First, we can break that statement along the boundaries of the *sentences*, explicating the logical gates and the grouping that holds it together. At the cost of adding some redundancy, we will also make the individual sentences a bit more self-sufficient, so that one sentence does not depend on the context introduced by another. Some (non-logical) shades of meaning will be lost in this analysis.

$$\Big((\text{this dog is always happy after a meal}) \wedge \\ (\text{this dog is troubled and restless})\Big) \Rightarrow (\text{this dog is hungry}).$$

If we replace different statements with different letters, we get

$$(A \wedge B) \Rightarrow C,$$

which cannot possibly be a valid argument. Indeed, take A and B true and C false. This choice of the truth values will make the whole implication false demonstrating that this argument is invalid.

On the other hand, it should be intuitively clear that what we had before this replacement of statements with letters was a valid argument. How can we reconcile these two conclusions?

The problem here is the insufficient depth of our analysis. To demonstrate validity of this argument, we need to break it down further to reveal more of its propositional structure:

$$\Bigg(\Big((\text{the dog has eaten}) \Rightarrow (\text{the dog is happy}) \Big) \wedge \Big(\neg\, (\text{the dog is happy}) \Big) \Bigg) \Rightarrow \Big(\neg\, (\text{the dog has eaten}) \Big)$$

This looks pretty cumbersome. To make it a bit more readable, a different notation[10] is usually favored in situations such as this:

$$\begin{array}{l} (\text{the dog has eaten}) \Rightarrow (\text{the dog is happy}) \\ \neg\, (\text{the dog is happy}) \\ \hline \neg\, (\text{the dog has eaten}) \end{array}$$

In the above, the horizontal line stands for the main implication of the argument and can be read as "therefore". The assumption of the argument is above the horizontal line, and the conclusion is below that line. The assumption is broken — as much as possible — into a conjunction of smaller statements, written individually one per line. These conjunctions are implicit in this form of writing.

To focus on the statement-gate structure of this argument, substitute the atomic statement "the dog has eaten" with E, and "the dog is happy" with H. Then it becomes:

[10] in these notes, we will call it "the Gentzen's notation"

$$\begin{array}{l} E \Rightarrow H \\ \neg H \\ \hline \neg E \end{array}$$

or, going back to the original form (which is less cumbersome now):

$$\Bigg(\Big(E \Rightarrow H \Big) \wedge \Big(\neg H \Big) \Bigg) \Rightarrow \Big(\neg E \Big),$$

Consider the truth table of this formula. To complete it, we used the truth tables around page 8 defining implication, conjunction and negation:

E	H	$\Bigg(\Big(E \Rightarrow H \Big) \wedge \Big(\neg H \Big) \Bigg) \Rightarrow \Big(\neg E \Big)$
T	T	T
T	F	T
F	T	T
F	F	T

Its last column shows that the argument in question is true for every possible combination of the truth values of the ingredient statements E and H. This is exactly the indicator we have been looking for.

Definition (of `tautology`): A propositional formula is called a `tautology` if and only if it is true for any combination of truth values of its ingredient statements. $\diamondsuit$

Theorem (*Valid Propositional Argument is a Tautology*). *An argument of propositional logic is valid if and only if that argument is a tautology.* $\diamondsuit$

Proof. [11] For an argument, being a tautology means having true conclusion whenever the assumptions of the argument are true. This is exactly the definition of validity of an argument. □

[11] Right now, we use the word "proof" informally, as a substitute for "a (hopefully) convincing explanation". The concept of proof will be the center of our attention later, when we will give it a precise definition.

The argument whose validity we verified in this example is called `modus tollens`, which is Latin for “method of removing” — meaning removing the assumption whenever it leads to a false conclusion.

> **HOMEWORK:** Verify, using the truth tables of implication and conjunction, that the logical formula expressing modus ponens (on page 10) is a tautology. Thus modus ponens itself is a valid argument.

***

Let’s analyze the second argument “This dog is always happy after a meal. If you don’t feed it, it will be very angry.” — the same way we did the first one. Explicating the logical gates and rephrasing some parts of the original argument to match the instances of the same atomic statement across different sentences, we get:

$$\Big(\text{(the dog has eaten)} \Rightarrow \text{(the dog is happy)}\Big) \Rightarrow \left(\Big(\neg\,\text{(the dog has eaten)}\Big) \Rightarrow \Big(\neg\,\text{(the dog is happy)}\Big)\right),$$

or, in a more concise form:

$$\frac{\text{(the dog has eaten)} \Rightarrow \text{(the dog is happy)}}{\neg\,\text{(the dog has eaten)} \Rightarrow \neg\,\text{(the dog is happy)}}$$

Using the same abbreviations as before, the statement-gate structure of this argument can be written as:

$$\frac{E \Rightarrow H}{(\neg E) \Rightarrow (\neg H)}$$

or, in the formula form:

$$\Big(E \Rightarrow H\Big) \Rightarrow \Bigg(\Big(\neg E\Big) \Rightarrow \Big(\neg H\Big)\Bigg).$$

The truth table of this formula can be computed like before:

E	H	$\Big(E \Rightarrow H\Big) \Rightarrow \Bigg(\Big(\neg E\Big) \Rightarrow \Big(\neg H\Big)\Bigg)$
T	T	T
T	F	T
F	T	F
F	F	T

The F in the last column signals that the argument is invalid. It corresponds to the case when E is false and H is true. In that case, the assumption of the argument, namely $E \Rightarrow H$, is true, but the conclusion $\Big(\neg E\Big) \Rightarrow \Big(\neg H\Big)$ is false.

Definition (of `counterexample`): For any given argument, a situation making its assumptions true and conclusion false is called a `counterexample` to that argument. For instance, the combination of false E and true H is a counterexample to the argument we are considering. ♢

A counterexample to an argument shows that the conclusion of that argument is not supported by the assumption. Thus, an argument is invalid if and only if it has at least one counterexample.

Note that an invalid argument can full well have a true conclusion. Validity of an argument has nothing to do with the truth or falsehood of that argument's assumption or conclusion. It merely concerns itself with whether or not the conclusion is supported by the assumption.

Furthermore, within logical discourse, we are not concerned whether our counterexamples are feasible in the real world. Any combination of truth values of the constituent atomic statements can serve as a (counter)example.

If the counterexample is indeed impossible in the real world, it means that our assumptions don't capture full relevant details of the situation we want to model in our argument. When you encounter an intuitively correct argument with true conclusion that is formally incorrect, most likely there is a problem with the assumptions not providing an accurate model of the world, or the analysis of the argument not going deep enough to reveal the structure that makes the argument valid.

2.3 Digression: Languages, Grammars and the Backus Notation

Speaking about any subject requires the use of a *language* capable of carrying the intended meaning. For this reason, the language is as important to study of logic as logical inference itself. However, in this section we merely scratch the surface of the language theory, illustrating some of its basic ideas by applications to Propositional Logic. For a more thorough introduction, lee section FIXME.

Language carries meaning in its form, and the form of a language is described by the grammar. A language is defined by its alphabet and expressions.

Definition (of `alphabet and letters; language and expressions`): Suppose A is a finite set. Denote as $A^{\mathbb{N}}$ the set of all finite sequences of the elements of A. Example: if we take $A = \{0, 1\}$, then 10010001000101100110010011001001111 is an example of such a sequence. By definition, a `language` $\mathscr{L}$ over an `alphabet` A is a particular subset of $A^{\mathbb{N}}$. The individual elements of A are called `letters` of the alphabet A, and the individual elements of $\mathscr{L}$ are called `expressions` of the language $\mathscr{L}$. Thus, the letters of an alphabet are the atomic building blocks of the expressions in the language over that alphabet. `expressions` $\diamondsuit$

Definition (of `grammar`): A grammar of a language $\mathscr{L}$ is a set of rules permitting to decide effectively, for any finite sequence of letters, whether or not that sequence is an expression of the language. $\diamondsuit$

Not every language can be described by a grammar. However, all the languages of the immediate interest to us will not only be describable by a grammar, but will permit a description of a very particular kind, namely as a finite list of production rules. A production rule is a recepie for a replacing one string of symbols with another.

Definition (of `symbols`): `Symbols` of a grammar are the letters of an extended alphabet, made of the original alphabet of the language in question (those are called the `terminal` symbols) and additional letters that are used by the grammar to conceptualize certain types of terminal symbol sequences that act as building blocks of the language's expressions. ♢

Example (Backus notation for describing comma-separated lists): Suppose we want to give a precise description for "a list of one or more digits, separated by commas followed by space"[12]. We can describe the grammar of the language of such lists using what is called `Backus Notation`[13]. This notation defines the grammar of a language incrementally, by listing the `production rules` for its symbols:

<list> ::= <digit> | <digit> , <list>

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

where

- the symbol "::=" is a part of the Backus notation, and means "can be replaced with". Each line with this symbol defines a new `production rule`.
- the symbol "|", also a part of the Backus notation, means "or". In principle, we could go without it, replacing the above with:

  <list> ::= <digit>
  <list> ::= <digit> , <list>
  <digit> ::= 0
  <digit> ::= 1
  <digit> ::= 2
  <digit> ::= 3

[12]One may ask why would we need a separate notation for what we just described using natural language. It turns out that in a more complicated situation — for example, when specifying a programming language — the notation we are about to introduce would be much more precise and succinct, which makes it worth the effort.

[13]It is often called `Backus-Naur Form` or `Backus Normal Form`. Both of these terms are incorrect and misleading, so we settle for `Backus Notation` instead of the more standard terminology.

$$\begin{array}{lcl}
\text{<digit>} & ::= & 4 \\
\text{<digit>} & ::= & 5 \\
\text{<digit>} & ::= & 6 \\
\text{<digit>} & ::= & 7 \\
\text{<digit>} & ::= & 8 \\
\text{<digit>} & ::= & 9
\end{array}$$

- the expressions between angle brackets, namely "`<digit>`" and "`<list>`", are called the `non-terminal` symbols. Those pertain to a particular grammar of a language, so that different grammars with different non-terminals may describe the same language. Non-terminals can help structure the grammar by classifying fragments of the expressions.

- the non-terminal symbol "`<list>`" is called the `start symbol` of the grammar. It gives the name to well-formed *expressions* of the language in question. Every grammar should have exactly one start symbol.

- the expressions without angle brackets, namely the ten digits, the comma and the space, are called the `terminal` symbols of the language. Their combinations make up the expressions of the language.

♢

In all the grammars we will consider in these notes, the space symbol will not convey any meaning, permitting us to add it liberally to our expressions just to make them more readable or to stress something in particular. From the formal point of view, all spaces in such situations should be ignored as if they were not there.

How does a grammar help us decide if a sequence of letters constitutes an expression of the language defined by that grammar? The following definition introduces the concept answering this question.

Definition (of `parse tree`): When a language is defined by a grammar, each expression of that language must come from a tree. The root of that tree must be the start symbol of the grammar, each node must correspond to a particular production rule, and each leaf must be a terminal symbol. All leafs taken together should give the expression itself. Such a tree is called the `parse tree` of the expression. ♢

Example (parse tree):
For example, the list “1, 5, 7” has the parse tree

```
<list> ::=
          <digit> ::=
                     "1"
          ","
          " "
          <list>  ::=
                     <digit> ::=
                                "5"
                     ","
                     " "
                     <list>  ::=
                                <digit> ::=
                                           "7"
```

◊

> **HOMEWORK:** Use the Backus notation to specify the grammar for all possible (signed or unsigned) decimals.

The parse tree of an expression gives the precise meaning to that expression. Thus each comprehension task involves parsing stage.

2.4 Disjunctive Normal Form

It turns out that any logical gate (no matter how many statements it combines) can be expressed as a disjunction of elementary conjunctions. More precisely, disjunctive normal form is defined by the following Backus notation.

Definition (of `disjunctive normal form`):

<DNF> ::= <elementary_conjunction>
| <elementary_conjunction> ∨ <DNF>

<elementary_conjunction> ::=
<term> | (<term> ∧ <elementary_conjunction>)

<term> ::= <variable> | (¬ <variable>)

<variable> ::= A | B | C | D | ...

♢

(Notice also that we started to use spaces for readability.)

HOMEWORK: What are the start symbol, the terminal symbols, the non-terminal symbols of this grammar?

Example (disjunctive normal form):

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

is a disjunctive normal form. ♢

HOMEWORK: Construct the parse tree of the disjunctive normal form for the above DNF.

Given the truth table of a logical gate, it is extremely easy to determine its DNF. One just needs to write a term for each row with T output listing all inputs equal to T as themselves, and all inputs equal to F as their negations.

Example (DNF of implication): Take the implication

A	B	$A \Rightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

The first line gives the elementary conjunction $A \wedge B$; the second line results in F and thus does not give any elementary conjunction; the third gives $(\neg A) \wedge B$, and the fourth results in $(\neg A) \wedge (\neg B)$. Combining all these elementary conjunctions together, we get the disjunctive normal form of implication:

$$\Big(A \Rightarrow B\Big) \Leftrightarrow \Big(\big(A \wedge B\big) \vee \big((\neg A) \wedge B\big) \vee \big((\neg A) \wedge (\neg B)\big)\Big) .$$

$\diamondsuit$

2.5 Standard Identities of Propositional Logic

There are many identities connecting different propositional formulas, similar in kind to the familiar associativity, commutativity and other identities of arithmetic.

Some of these identities are discussed in this section.

2.5.1 Identities with Implication

The implication $A \Rightarrow C$ is equivalent to stating that either the assumption A is false — and we are not responsible for any conclusions we make — or the conclusion C must be true:

$$\Big(A \Rightarrow C\Big) \Leftrightarrow \Big((\neg A) \vee C\Big) .$$

Also worth mentioning is the fact that the equivalence of two statements means that each of these statements implies the other:

$$\Big(A \Leftrightarrow B\Big) \Leftrightarrow \Big((A \Rightarrow B) \wedge (B \Rightarrow A)\Big) .$$

HOMEWORK: Verify these two equivalences using truth tables.

2.5.2 Arithmetic Properties of Conjunction and Disjunction

There is a lot of similarity between addition and multiplication on one hand, and conjunction and disjunction — on the other. Conjunction and disjunction share the properties of commutativity and associativity with the two arithmetic operations.

> **HOMEWORK:** Formulate and verify using the truth tables the properties of commutativity and associativity for disjunction and conjunction.

A more subtle point is the existence of a neutral element. If we consider logical gates as operations on *truth values* of the constituent statements, rather than the statements themselves, then conjunction and disjunction share the property of a neutral element — in this case neutral truth value — with the arithmetic operations. Recall, that a neutral element n of a binary operation $\bullet$ is the element with the property $n \bullet x = x \bullet n = x$ for any x that can be used with that operation. For example, the neutral element of addition is zero, because $0 + x = x + 0 = x$ for any number x.

> **HOMEWORK:** Which one of the two truth values, "true" and "false", is the neutral one for disjunction? Which one is neutral element for conjunction?

2.5.3 De Morgan Laws

Theorem (*negation of disjunction*). *Negation of a disjunction is the conjunction of individual negations:*

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

♢

Theorem (*negation of conjunction*). *Negation of a conjunction is the disjunction of individual negations:*

$$\neg(A \wedge B \ldots) \quad \Leftrightarrow \quad (\neg A) \vee (\neg B) \ldots \,.$$

$\diamondsuit$

2.5.4 Distributive Laws

Theorem (*distributivity of conjunction with respect to disjunction*).

$$A \wedge (B \vee C \ldots) \quad \Leftrightarrow \quad (A \wedge B) \vee (A \wedge C) \ldots \,.$$

$\diamondsuit$

Theorem (*distributivity of disjunction with respect to conjunction*).

$$A \vee (B \wedge C \ldots) \quad \Leftrightarrow \quad (A \vee B) \wedge (A \vee C) \ldots \,.$$

$\diamondsuit$

HOMEWORK: Verify these four theorems using truth tables.

Example (using logical gates and identities in solving equations): Suppose we want to solve the equation

$$\frac{x^2-4}{x-2} = 2.$$

One possible way to go about it is to find common denominator and get everything on one side:

$$\frac{x^2-4}{x-2} = 2 \quad \Leftrightarrow \quad \frac{x^2-4}{x-2} = \frac{2x-4}{x-2} \quad \Leftrightarrow \quad \frac{x^2-4-2x+4}{x-2} = 0 \quad \Leftrightarrow \quad \frac{x^2-2x}{x-2} = 0 \quad \Leftrightarrow \quad \frac{x(x-2)}{x-2} = 0.$$

Since a fraction is zero if and only if the numerator is, and the denominator isn't zero, the last equation is equivalent to the system:

$$\begin{cases} x(x-2) = 0 \\ x - 2 \neq 0. \end{cases}$$

This simultaneous system is just a conjunction of two statements written in a different form. The first equation states that a product is zero. That means that one of the factors is zero. Thus

$$\begin{cases} x(x-2) = 0 \\ x - 2 \neq 0 \end{cases} \quad \Leftrightarrow \quad \begin{cases} \left[\begin{array}{l} x = 0 \\ x - 2 = 0 \end{array} \right. \\ x - 2 \neq 0. \end{cases}$$

where the square bracket is just another way of expressing disjunction.

Now we can use the distributivity of conjunction with respect to disjunction which then leads to the solution in one step:

$$\begin{cases} \left[\begin{array}{l} x = 0 \\ x - 2 = 0 \end{array} \right. \\ x - 2 \neq 0 \end{cases} \quad \Leftrightarrow \quad \left[\begin{array}{l} \begin{cases} x = 0 \\ x - 2 \neq 0 \end{cases} \\ \\ \begin{cases} x - 2 = 0 \\ x - 2 \neq 0 \end{cases} \end{array} \right. \quad \Leftrightarrow \quad x = 0.$$

♢

This example illustrates one particularly good way of presenting a solution of an equation, inequality, or a system thereof. It is called the `method of equivalence transformations`. In the context of this method, equivalence means *preservation of the the solution set* as we move from one step to the next. In our specific example, it means that the original equation

$$\frac{x^2 - 4}{x - 2} = 2$$

has the same solutions as the (trivial) equation $x = 0$, meaning that 0 is the solution of the original equation. When using this notation, one warning about order of operations is necessary.

In arithmetic, we usually drop parentheses when the same operation is repeated over and over. It can be done without harm in $x + y + z$ because of associativity of addition, which makes grouping insignificant. We also do it in non-associative situations, like $x \div y \div z$, by conventionally interpreted repeated division by "grouping left", i.e. as $(x \div y) \div z$. Less frequently "grouping right" convention is employed, as in $x^{y^z} = x^{(y^z)}$. However, when dealing with the logical operations of equivalence and implication, the meaning is different. When $A, B, C, \ldots$ are equations, or — more generally — statements, the notation $A \Leftrightarrow B \Leftrightarrow C \ldots$ stands for $(A \Leftrightarrow B) \wedge (B \Leftrightarrow C) \ldots$ which would correspond to an interlocking pattern of parentheses that is too easy to confuse for something else to use in practice:

$$(A \Leftrightarrow [B) \Leftrightarrow (C] \ldots.$$

In a sense, the implication and equivalence signs behave more like an equal sign than an operation symbol.

2.6 History: Stoics

As we undertake our first brief incursion into the subject of history, I want to make a general remark that will pertain to all historical sections in these notes.

Deep ideas are like rivers. When they gather enough strength to get noticed and recognized, their full content is rarely the result of an output from a single source. More typically, they manifest a confluence of many streams of thought, sometimes even contributed at the same time by independent thinkers, and often running underground completely hidden from an observer, only to reappear later as a crucial admixture in a bigger stream. This makes it difficult to definitively attribute any profound theory or an idea to a single author or a point in time. The best I hope to accomplish in my historical notes is to mark the development of logic not by building monuments at the symbolic but often meaningless origins, but by taking scenic shots in places where the confluence and synergy of already full-bodied ideas created new depth evident enough to be recognized and admired.

While the ideas of propositional logic can be traced at least as far back as Aristotle[14] and Tyrtamus Theophrastus[15], they reached the level of a developed system of reasoning in the works of another Greek philosopher, Chrysippus of Soli.

[14]more about him later

[15]c. 371 – c. 287 BC, Greek philosopher of Peripatetic School who succeeded Aristotle as the school's head.

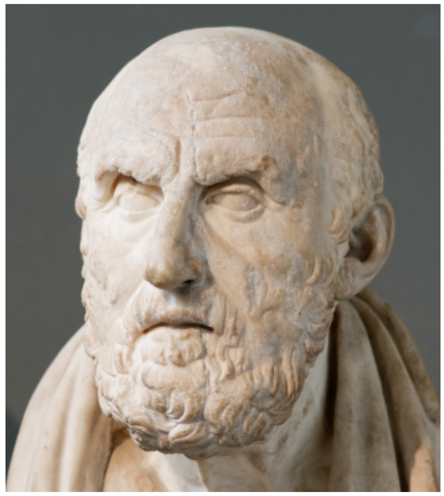


Figure 2: Χρύσιππος [Chrysippos] (c.279–c.204 BC)

Regarded as the leading logician during his own life time, Chrysippus was overshadowed by Aristotle in subsequent history. Chrysippus already had the notions of propositions, logical gates, and argument forms. Roughly speaking, propositional logic is synonymous with "Chrysippus logic".

3 Syllogistic Logic

3.1 Ontology: Categories and Quantifiers

Syllogistic logic shifts the focus away from logical gates (even though it cannot completely dispose of them), and adds the notions of `category`[16] and `quantifier` to propositional analysis of arguments. Consider the following argument:

Example (Syllogistic Argument): All cats are mammals. All mammals are vertebrates. Thus all cats are vertebrates. ♢

Propositional analysis yields:

$$\Big(\text{(all cats are mammals)} \wedge \text{(all mammals are vertebrates)}\Big) \Rightarrow$$
$$\text{(all cats are vertebrates)}\,.$$

If we replace different statements with different letters, we will get an invalid argument:

$$\Big(C \wedge M\Big) \Rightarrow V$$

HOMEWORK: Why is this argument invalid?

3.2 Apologia: Euler-Venn Diagrams

We have faced a similar situation earlier when a valid propositional argument appeared invalid because we did not go deep enough in its analysis (see page 13). Here, however, deeper *propositional* analysis is impossible: there are no parts in the statements (that we can break away as *full statements*) matching similar parts in other statements.

[16] Note that the word "category" came to mean something entirely different in modern mathematics. The mathematical concept which is the most accurate representation of a category in the sense we use here is the concept of a `set`.

Intuitively, the original argument should be valid — it is the propositional logic that is too crude of a tool to reveal enough of the structure of this argument to show its validity. To capture the connections between individual statements in our analysis, we need to consider building blocks smaller than full statements. Noticing that the words "cats", "mammals" and "vertebrates" are each shared by two of the three statements, we can use those words to pin down the interlocking relation among the statements. Thinking about these words as `categories` of objects in some universe, we can represent objects as points on the plane and those categories — as the domains on that plane. This way we can represent the argument in a graphic form:

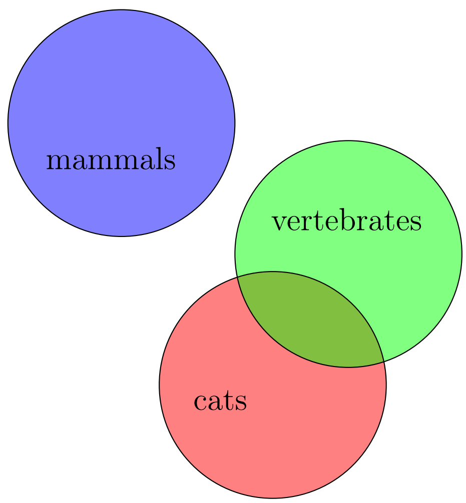


Figure 3: Sketch of the Categories — Too Presumptuous

But wait... This picture seems to imply that no mammal can be either a vertebrate or a cat. When making an initial sketch of the categories involved in an argument, we must avoid the possibility of imparting our picture with any assumptions not postulated in that argument. In other words, we need to draw the disks in *common position*. This idea leads to the following definition:

Definition (of `Venn diagram`): An arrangement of sets on the plane that makes any combination of membership status possible is called a `Venn diagram` of those sets. ♢

HOMEWORK: How many membership possibilities are there for three sets? For n sets, where $n \in \mathbb{N}$?

The following sketch is an example of a Venn diagram for the three categories considered in our argument:

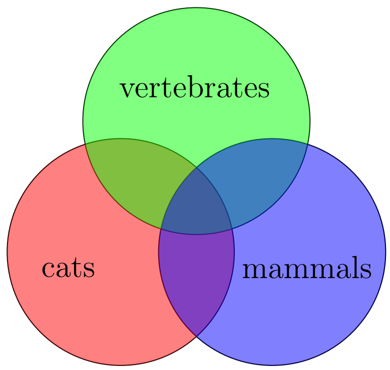


Figure 4: Venn Diagram of the Categories

We can encode the fact "all cats are mammals" by horizontally shading the part of "cats" category lying outside of the "mammals" category:

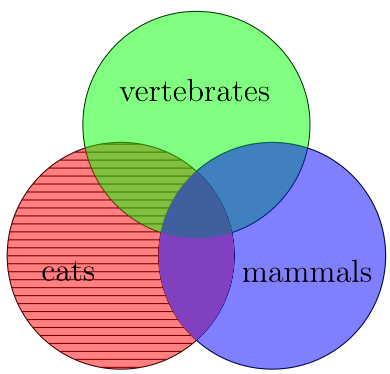


Figure 5: Venn Diagram with One Premise Marked

The shading indicates that no object is permitted to be in the shaded area. Continuing with our analysis, we can encode the fact "all mammals are vertebrates" by vertically shading the part of "mammals" category lying outside of the "vertebrates" category:

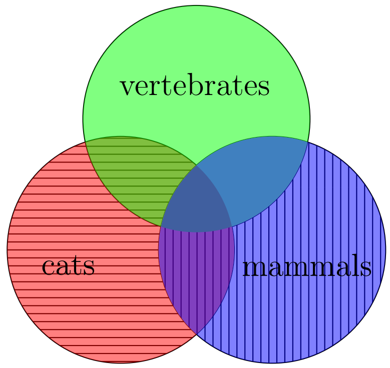


Figure 6: Venn Diagram with Both Premises Marked

This analysis shows that the real picture of categories mentioned in this particular argument looks like this:

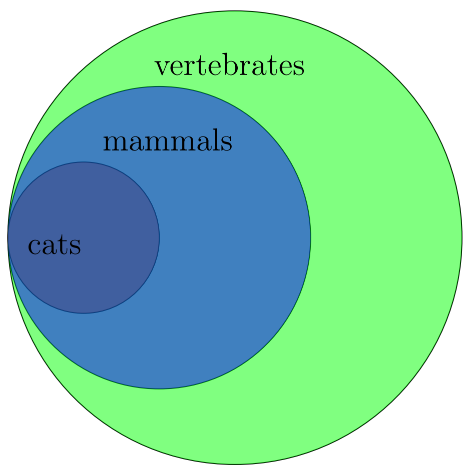


Figure 7: Euler Diagram of the Syllogism

This way of representing the information at hand is called an `Euler diagram`. In contrast with Venn diagrams, which start with an assumption-free depiction of the categories, Euler diagrams summarize the results of Venn diagram analysis by showing the actual configuration of categories reflecting the assumptions of the argument being analyzed.

The above Euler diagram demonstrates the validity of our argument by translating the statements "everybody of this kind is of that kind" into geometric statements "this category is *inside* of that category".

The conclusion that "cats" are inside "vertebrates", is obviously supported by the assumptions that "mammals" are inside "vertebrates" and "cats" are inside "mammals".

Example (syllogistic argument that depends on existential presupposition): Since all unicorns are mammals, and all mammals are animals, we can conclude that some unicorns are animals. ♢

This example corresponds to the following Venn diagram, where the vertical shading represents the assumption "all unicorns are mammals", and the horizontal one — the assumption "all mammals are animals":

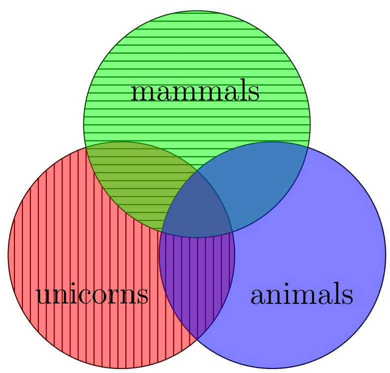


Figure 8: Venn Diagram of the Syllogism

Optionally, the same information can be represented by the Euler diagram

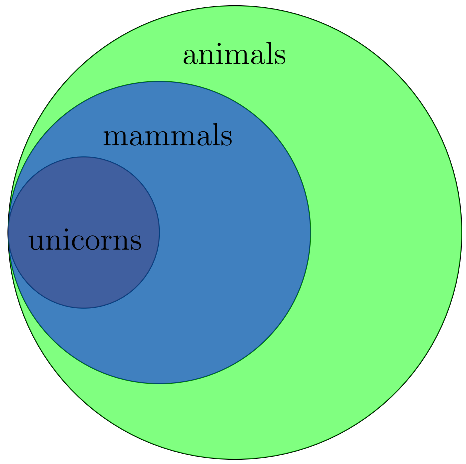


Figure 9: Euler Diagram of the Syllogism

The conclusion we want to test is whether or not this arrangement of the categories guarantees existence of at least one object that would be both a unicorn and an animal.

This example exposes one potential ambiguity in interpreting everyday language. In our particular case, does the phrase "all unicorns are mammals" imply existence of unicorns?

Definition (of `existential presupposition`): The convention stipulating the presumption of existence of an entity mentioned in a noun phrase within a factual[17] context, is called `existential presupposition`. ♢

Existential presupposition — just like any other accepted, but not explicitly stated assumption — may cause many errors in logical reasoning. It is better to avoid it, agreeing to resolve this ambiguity in meaning by the requirement of stating the existence explicitly. Thus, for the rest of these notes, we adopt the convention of *rejecting* the existential presupposition. When we say "all unicorns are mammals", that will mean is really this: "all unicorns — if they exist — are mammals". With our convention in effect, this statement is true when applied to the world we live in — the world with no unicorns — since an implication with a false assumption is always true. Somewhat paradoxically, in our world the statement "all unicorns are not mammals" is also true.

[17] as opposed to counterfactual

With existential presupposition rejected, the above picture may reflect the situation with (some or all) of the categories being empty. Taking empty "unicorns" category (which happens to be the case in the real world) provides a counterexample to this argument. This counterexample shows that the conclusion "some unicorns are animals" is not supported by the assumptions and thus this argument is invalid.

HOMEWORK: We just realized that "some unicorns are animals" is a false conclusion from the premises of this argument. Can we conclude that "all unicorns are animals"?

Example (syllogism): Some people understand logic, but dogs are not people, therefore no dog understands logic. ♢

We start our analysis with a Venn diagram:

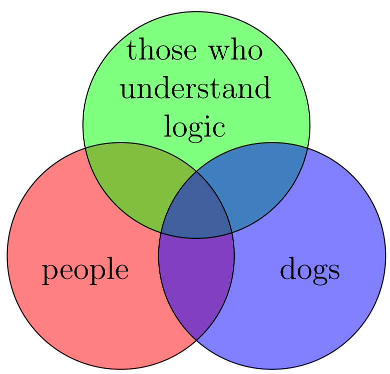


Figure 10: Starting Venn Diagram of the Syllogism

The assumption "some people understand logic" can be visually expressed by drawing an interval within the intersection of "people" and "those who understand logic". The interval is a visual way to indicate the existence of an object (somewhere along that interval) without making an unwarranted assumption about the specific location of that object. The interval, as opposed to a point, indicates that the object in question may be on either side of the "dogs" boundary. [18]

[18]Similarly, in quantum mechanics one gives up on identifying the precise location of a

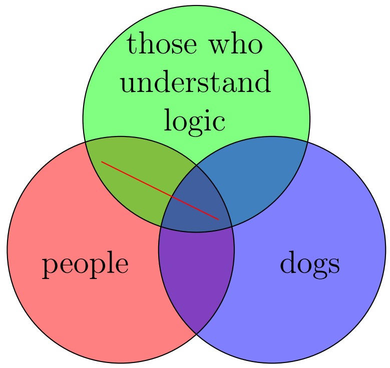


Figure 11: Venn Diagram of the Syllogism with the Existence Clause Marked

The assumption "dogs are not people" can be expressed in the familiar way, namely by shading the region where nothing is permitted to be. In this case, the forbidden region is the intersection of "people" and "dogs": Note that the forbidden region removes the ambiguity expressed by the interval notation, which can now be replaced by a single point.

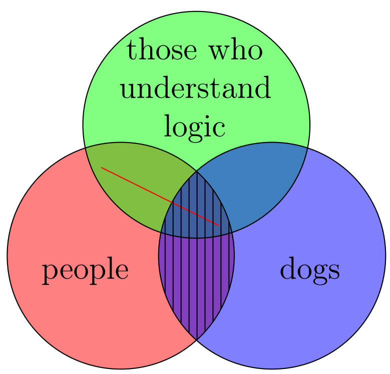


Figure 12: Completed Venn Diagram of the Syllogism

The same information can be expressed by the following Euler diagram:

quantum particle, settling instead for a region where the particle is likely to be found.

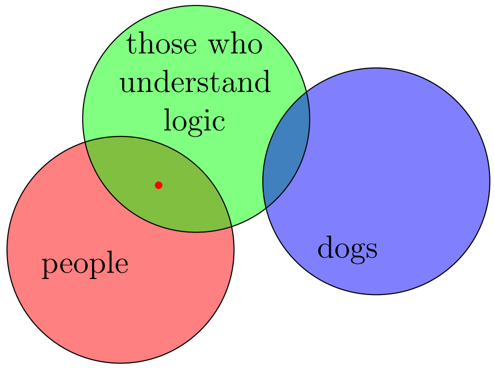


Figure 13: Euler Diagram of the Syllogism

Recall that we want to test the conclusion “no dog understands logic”. This conclusion does not follow from the assumptions. Indeed, imagine a world where in addition to the object marked by the red dot (that object must exist because of the assumptions), there is just one more object depicted by the yellow dot in this Euler diagram:

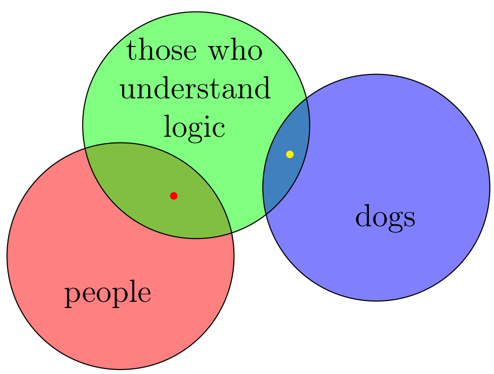


Figure 14: Counterexample for the Syllogism

In other words, make the world with only two objects: one person who understands logic (the red dot above), and one dog who understands logic (the yellow dot above). In that world, the assumptions of this argument are satisfied, but the conclusion is false. Thus this world represents a counterexample for this argument, showing the argument to be invalid. (It is entirely beside the point that such a world does not look like our real world. We are considering the argument in its formal sense, disregarding its connection to reality.)

3.3 Ontology: Definition of Syllogism

We managed to get through most of Syllogistic Logic without giving precise definition of what a `syllogism` is. Aristotle used the term συλλογισμός [syllogismos] in the general sense of logical inference, to describe a *valid* argument with several premises. In later studies, the word "syllogism" came to mean an argument — valid or invalid — of some very specific form. As discovered by Leibnitz, those arguments can be resolved by Euler-Venn diagrams we considered earlier. However, when understood in the modern sense, syllogisms comprise a very limited class of arguments and clearly fall short of encompassing all rational reasoning. Perhaps this is an example when the method — Euler-Venn diagrams — is more important then the object this method is applied to. Even though Euler-Venn diagrams can be used for more general arguments (like some of those with more than three categories), the needs of rational reasoning quickly outgrow what those diagrams can handle, necessitating the use of more potent tools of validation, like proofs.

We continue our study of syllogisms to pay homage to Aristotelian tradition, even though these ideas will be superseded by the Proof Theory. The remainder of the section on Syllogistic Logic, notwithstanding its cultural and historical significance, may help you better appreciate the coherence, strength and clarity of Proof Theory when it is viewed against the backdrop of the traditional validation techniques.

Let's try to define what a syllogism is by specifying its form using Backus notation.

Definition (of `syllogism`): A `syllogism` is a categorical argument that has the following form:

<syllogism> ::=
<major_premise>
<minor_premise>
____
<conclusion>

<major_premise> ::= <clause>

<minor_premise> ::= <clause>

<conclusion> ::= <clause>

<clause> ::= <quantifier> <subject> <copula> <predicate> .

<subject> ::= <category>

<predicate> ::= <category>

<quantifier> ::= <universal> | <existential>

<universal> ::= All

<existential> ::= Some

<copula> ::= are | are not

<category> ::= ...

where the ellipsis '...' in the last production rule stands for the list of the categories specific to syllogism.

The example starting on 34 follows this format after being slightly rephrased:

```
Some people are those who understand logic.
All dogs are not people.
----------------------------------------------
All dogs are not those who understands logic.
```

provided we specify:

```
<category> ::= people
             | dogs
             | those who understand logic
```

♢

In the above definition, we used the Backus notation itself to introduce the terms `clause`, `major premise`, `minor premise`, `subject`, and `predicate`.

> **HOMEWORK:** What are the starting, non-terminal and terminal symbols in this syllogism grammar?

> **HOMEWORK:** For a given syllogism, we can infer from the above definition that all premises are clauses. Can you think of any other such statements that say something about any syllogism and follow the format of a clause?

While correct in specifying the format of a syllogism, the above grammar does not fully characterize what it means to be one. For example, it permits the clause "some dogs are dogs" which is not something we would like to consider as acceptable.

***

Aside. This is a typical problem in computer science, where programming languages may be too complicated to be fully described in terms of their form alone. Defining what constitutes a valid computer program in a given programming language is often done in several stages. First, using Backus notation, one describes the proper *syntax* of a program. Then, specification of the meaning of constituent commands allows to formulate *semantic* correctness. Finally, the analysis showing that the program does what it is designed to do demonstrates its *pragmatic* correctness. We could follow that method by adding *semantic* requirements on the top of the *syntactic* definition given on the previous page. Specifically, we can require the categories and the clauses of a syllogism to be related to each other like the vertices and sides of a triangle. Each clause, like a side of a triangle, must contain two different categories, which would be like the vertices of that triangle, with the same incidence structure.

***

Definition (of `syllogism's figure`): The specific arrangement of any three categories among the clauses of a syllogism is called the `figure` of that syllogism. There are four possible figures, listed in the grammar that appears on the next page. ◇

Since we can list all the figures as separate production rules, syllogisms are, after all, simple enough to be completely described by their syntax alone. But before we do that, we need to clarify our terminology a bit. In the above definition of syllogism we called the first category of any clause the `subject` of that clause, and the its second category — the `predicate`[19] of that clause. However, in the following Backus grammar, we will use the word "subject" as a short hand for "the category that is the subject of the conclusion of the syllogism in question". Likewise, the word "predicate" will stand for "the category that is the predicate of the conclusion of the syllogism in question". To avoid possible confusion between the two different uses of those terms, pay attention to whether or not we are applying the words "subject" and "predicate" to a particular clause or to the whole syllogism.

The third category of a syllogism, which occurs in each premise but is absent from the conclusion, is referred to as the "middle" in the grammar below.

[19] Later, the word "predicate" will be used in a more general sense.

```
<syllogism>    ::= <FIGURE_1> | <FIGURE_2> | <FIGURE_3> | <FIGURE_4>

<FIGURE_1>     ::= <quantifier> <middle>    <copula> <predicate> .
                   <quantifier> <subject>   <copula> <middle> .
                   ----------------------------------------------
                   <quantifier> <subject>   <copula> <predicate> .

<FIGURE_2>     ::= <quantifier> <predicate> <copula> <middle> .
                   <quantifier> <subject>   <copula> <middle> .
                   ----------------------------------------------
                   <quantifier> <subject>   <copula> <predicate> .

<FIGURE_3>     ::= <quantifier> <middle>    <copula> <predicate> .
                   <quantifier> <middle>    <copula> <subject> .
                   ----------------------------------------------
                   <quantifier> <subject>   <copula> <predicate> .

<FIGURE_4>     ::= <quantifier> <predicate> <copula> <middle> .
                   <quantifier> <middle>    <copula> <subject> .
                   ----------------------------------------------
                   <quantifier> <subject>   <copula> <predicate> .

<quantifier>   ::= <universal> | <existential>
<universal>    ::= All
<existential>  ::= Some

<copula>       ::= <affirmative> | <negative>
<affirmative>  ::= are
<negative>     ::= are not

<subject>      ::= ...
<middle>       ::= ...
<predicate>    ::= ...
```

Each of the last three production rules needs a specific (and distinct) category in place of its ellipsis "...". Thus, interpreting the example that was introduced on 34:

```
Some people are those who understand logic.
All dogs are not people.
-----------------------------------------------
All dogs are not those who understands logic.
```

as a Figure 1 syllogism, we would specify:

```
<subject>      ::=  dogs
<middle>       ::=  people
<predicate>    ::=  those who understand logic
```

Also note that in the traditional syllogystics,

- the "existential quantifier" was instead called the "particular quantifier";
- the form "no dog understands logic" was used instead of "all dogs do not understand logic" which we adopted to make our grammar more regular;
- existential presupposition was in effect, so that "all dogs are not people" was understood to mean "there exists at least one dog and those dogs that exist are not people" [20].

HOMEWORK: Why is the number of possible figures of a syllogism exactly four?

Definition (of `syllogism`): Syllogism is a logical argument, which is constructed according to the grammar specified on the previous page. In determining whether or not an argument is a syllogism, we disregard the issue

[20]sometimes existential presupposition is called the "`existential import to universal quantifier`" in this context

whether or not that argument is *logically* valid. The only question that needs to be addressed when classifying an argument as a syllogism is its conformance with the above *grammar* rules. ♢

Definition (of `mood of a clause`): The specific combination of a quantifier and a copula is called the `mood` of a clause in a syllogism. There are four possible moods. ♢

> **HOMEWORK:** Why is the number of possible moods of a syllogism clause exactly four?

Definition (of `syllogism class`): All syllogisms that are the same, except for possibly their categories, form a `syllogism class`. For example, the syllogism we considered on page 28 belongs to the same class as "all cats are pets, all pets like to play; therefore all cats like to play". Syllogisms that belong to the same class are represented by the same Venn and Euler diagrams. ♢

> **HOMEWORK:** How many syllogism classes are there in total? (Hint: how many moods are there per clause? per syllogism?)

3.4 Apologia: Reduction; Mnemonics

Traditional way of validating syllogisms was based not on Euler-Venn diagrams but on the technique called *reduction*. Figure 1 was called the "perfect figure", and the syllogisms classes in that figure were called "perfect syllogisms".[21] Validity (or lack thereof) of perfect syllogisms was taken as self-evident. Reduction was a step-by-step process connecting[22] the imperfect syllogism in question to the corresponding perfect syllogism. The validity of the syllogism under consideration would rest on the validity of

[21] Here and elsewhere we will often say "syllogism" when in fact referring to a syllogism class.

[22] more on the nature of that connection later

the corresponding perfect syllogism and on the soundness of each step. This incremental nature of validation by reduction forshadows the idea of a `proof` which we will explore in depth later in these notes.

This validation technique was ingrained into an eleborate mnemonic[23] system developed by medieval logicians. We describe that system (and the technique of reduction) in this section.

3.4.1 Encoding the Mood of a Clause

Vowels “a”, “i”, “e” and “o” encoded the mood of a clause:

- the word “**a**ff**i**rmo” (Latin for “affirm”) giving the vowels for the two affirmative moods:
  - **a**: <universal> <affirmative>
  - **i**: <existential> <affirmative>
- and the word “n**e**g**o**” (Latin for “negate”) giving the vowels for the two negative moods:
  - **e**: <universal> <negative>
  - **o**: <existential> <negative>

3.4.2 Encoding the Syllogism Class

Each class of valid[24] syllogisms was denoted by a single word with three vowels:

FIGURE 1 MP, SM	**FIGURE 2 PM, SM**	**FIGURE 3 MP, MS**	**FIGURE 4 PM, MS**
Barbara	Cesare	Datisi	Calemes
Celarent	Camestres	Disamis	Dimatis
Darii	Festino	Ferison	Fresison
Ferio	Baroco	Bocardo	Calemos
Barbari	*Cesaro*	***Felapton***	***Fesapo***
Celaront	*Camestros*	***Darapti***	***Bamalip***

[23] i.e. aiding memory retention of certain information

[24] in the sense of logical validity, rather than conformity with the grammar on page 41 defining syllogisms

Those vowels encoded the mood of the major premise, minor premise, and the conclusion respectively, as noted on page 44. For instance, the word "Barbara", having three vowels "a", described the class of syllogisms with three universal affirmative clauses, like the one introduced on page 28.

In the above table, the syllogisms in italic are valid only with the existential presupposition. Those in non-bold italic admit a stronger conclusion. Each column corresponds to one figure and groups together the syllogisms with the same distribution of the categories among their clauses. Under the figure designation in the column headers, only the categories for the major and the minor premise are listed (since, by definition, the categories of the conclusion are **SP** for all the figures). This information about the figures, which comes from the syllogism grammar on page 41, is encoded in the phrase

Sub **p**re prima
bis **p**re secunda
tertia **bis** **s**ub

recalling [25] that the middle category is

1. the **s**ubject of the major premise, and the **p**redicate of the minor premise in the first ("prima") figure;

2. the **p**redicate of both premises in the second ("secunda") figure; and

3. the **s**ubject of both premises in the third ("tertia") figure.

HOMEWORK: Give an example of a "Barbara" syllogism. Using the same categories, convert it into a "Barbari". Then explain how the existential presupposition justifies "Barbari".

[25] William of Sherwood, *Introduction to Logic*, translated with an introduction and notes by Norman Kretzmann (University of MN Press: 1966), as cited in Sara L. Uckelman, *Syllogism Mnemonics*, November 16, 2017

3.4.3 Encoding Imperfect Syllogism's Reduction

The name of a valid imperfect syllogism class encodes not only the mood of its clauses, but also its reduction. The initial letter indicates which perfect syllogism corresponds to the imperfect one considered. **Example:** the initial **D** in the name of the imperfect syllogism "**D**isamis" signifies that it corresponds to the perfect syllogism "**D**arii". Letters "r", "t", "l", "n", as well as the non-initial "b" and "d", don't have any mnemonic meaning. Each non-initial letter **S**, **P**, **M** and **C** defines one reduction step:

- **S**: "**Si**mpl**e**x conversio", or simple clause conversion, is the interchange of the subject and the predicate in the clause denoted by the preceding vowel. It yields an *equivalent* proposition for clauses with **i** or **e** moods. **Examples:** an **i**-mood clause "some vertebrates are fish" is equivalent to "some fish are vertebrates"; an **e**-mood clause "all humans are not dogs" is equivalent to "all dogs are not humans". **Remark:** on the level of the whole syllogism, this transformation may distort the distribution of categories among the clauses to the extent that the resulting argument as a whole would no longer corresponds to any figure, and thus — while being equivalent to the original syllogism — would no longer be a syllogism[26] itself. If so, the resulting argument must be restored to the syllogism form by the **M** transformation considered below, where we also consider an example of this situation.

- **P**: "**P**er **a**ccid**e**ns conversio", or partial clause conversion[27], is a simple clause conversion combined with the reversal of the quantifier. Under the existential presupposition, this transformation yields a *weaker* statement[28] when the original clause had a universal mood[29], and a *stronger* one — when the original clause had an existential mood[30]. This conversion is used only for the syllogisms in bold italic font. **Examples:** an **a**-mood clause "all fish can swim" has a consequence "some of those who can swim are fish", as long as we accept that merely mentioning fish in the affirmative clause means that fish exist. Similarly, an

[26]in the sense defined by the grammar on page 41
[27]in this context, the word "partial" means limitation of the claim expressed by the original clause
[28]in the sense described on page 11
[29]namely mood **a** or **e**
[30]namely mood **i** or **o**

e-mood clause “all dogs are not birds” has a consequence “some birds are not dogs”. The above remark about the **S** clause conversion possibly breaking the syllogism grammar applies here as well, although this happens in only one instance, namely the “Bamalip” syllogism class[31].

- **M**: “**M**utatio syllogism”, is the interchange the premises of a syllogism. **Example (“Barbara”):** The syllogism “all mammals are vertebrates; all cats are mammals; thus all cats are vertebrates”, when mutated, results in “all cats are mammals; all mammals are vertebrates; thus all cats are vertebrates”. Since the premises are joined by an implicit conjunction, and conjunction is commutative, the mutated argument is *equivalent* to the original one. While trivial from the point of view of propositinal logic, this transformation may be necessary for restoring grammatical correctness of a syllogism distorted by the **S** or the **P** transformations. **Example (“Camestres”):** When we apply the **S** transformations[32] to the minor premise and to the conclusion of Camestres, its Figure 2 distribution of categories PM, SM, SP turns into PM, MS, PS. Since, by definition, S is the first, and P is the second category in the *conclusion* of a syllogism, we need to reverse the labeling of the subject and predicate category, which results in SM, MP, SP category distribution for the transformed argument. This distribution does not correspond to any figure and requires the use of the **M** transformation to get the MP, SM, SP distribution of the figure 1 syllogism “Celarent”.

- **C**: “**C**ontra syllogism”[33], applies to the whole syllogism like the **M** transformation, and is used only for “Baroco” and “Bocardo”, converting them into an equivalent[34] “Barbara” syllogism. This transformation makes new minor premise by taking the negation of the original conclusion, and new conclusion — by taking the negation of the original minor premise. As it turns out, the negation of a clause amounts to

[31]For example, all cats are mammals; all mammals are vertebrates; thus some vertebrates are cats.

[32]as encoded in the word “Camestres”

[33]I am not sure it was ever called that; traditionally it was used as an element in a proof by contradiction. For that reason, this particular validation type was called *reductio ad absurdum*, or “reduction by contradiction”.

[34]The claim that it is in fact an equivalence is the subject of the next homework.

the simultaneous reversal of its quantifier and copula[35]. **Example:** the negation of "some vertebrates are not dogs" is "all vertebrates are dogs".

3.4.4 Which Way Should the Reductions Go?

Note that each of the transformations **S**, **P**, **M** and **C** happens to be its own opposite, so that when applied repeatedly it toggles between the original imperfect and corresponding perfect syllogism[36]. For that reason, these transformations can in principle be applied in either direction. Which way should we go?

Theorem (*Reduction Implication*). *Each imperfect syllogism is a valid logical* consequence *of the corresponding perfect syllogism.* ♢

We will demonstrate it using case-by-case analysis.

Case 1: only **S** and **M** transformations are used in the reduction.
In this case we can make an even stronger claim, namely that the two syllogisms are actually *equivalent* to each other. This claim follows from examining the effect of these transformations on individual clauses (for **S**) or their conjunction that forms the assumption of the syllogism (for **M**).

> **HOMEWORK:** Verify, using Euler-Venn diagrams, that **S** transformation of a single clause in **i** or **e** mood results in an equivalent statement.

[35]This rule will be explained in more detail on page **??**.

[36]In mathematics, examples of operations with this behavior include taking the opposite (which sends 5 into -5 and vice versa) and taking the reciprocal (which sends $\frac{2}{3}$ into $\frac{3}{2}$ and vice versa). In (classical) logic this property is examplified by the negation which takes A into $\neg A$.

Case 2: The **C** transformation yields an equivalent argument.
It turns out that mere propositional analysis is sufficient to demonstrate it[37].

HOMEWORK: Demonstrate that the contra syllogism is equivalent to the original one. (SOLVED)

Solution. In the original syllogism, denote the major premise as P, minor premise — as p, and conclusion — as C. Then the original syllogism can be written as

$$(P \wedge p) \Rightarrow C$$

and the contra syllogism — as

$$\Big(P \wedge (\neg C)\Big) \Rightarrow (\neg p)$$

Replacing implications $X \Rightarrow Y$ with $(\neg X) \vee Y$, and using the De Morgan's law, together with some obvious properties like associativity and commutativity of disjunction, as well as removal of the double negation, we get:

$$\Big((P \wedge p) \Rightarrow C\Big) \Leftrightarrow \Big(\neg(P \wedge p) \vee C\Big) \Leftrightarrow \Big((\neg P) \vee (\neg p) \vee C\Big)$$

for the original syllogism, and

$$\bigg(\Big(P \wedge (\neg C)\Big) \Rightarrow (\neg p)\bigg) \Leftrightarrow \bigg(\neg\Big(P \wedge (\neg C)\Big) \vee (\neg p)\bigg) \Leftrightarrow \Big((\neg P) \vee (\neg p) \vee C\Big)$$

for the contra syllogism. Since the results of both equivalence chains are the same, we can conclude that the original and the contra syllogisms are equivalent.

[37]but a deeper analysis of the clause is needed to form its negation; the fact that the traditional approach relied instead on a more circuitus proof by contradiction may be a reflection of syllogistics preceeding propositional logic in terms of historical development

Case 3: The imperfect syllogism is a consequence of the corresponding perfect one when the **P** transformation is used in the reduction. Propositional analysis alone turns out to be sufficient in this case as well. Going back to the table on page 44 we can observe that:

- whenever the **P** transformation is applied to one of the premises[38], that premise is in the **a** (i.e. universal affirmative) mood, so that the transformation of the original clause yields a *weaker* statement; and
- whenever the **P** transformation is applied to the conclusion[39], that conclusion is in the **i** (i.e. existential affirmative) mood, so that the transformation of the original clause yields a *stronger* statement.

To sum it up, when the **P** transformation is applied to an improper syllogism, it either weakens its assumption, or strengthens its conclusion. The following two homework problems demonstrate that either way this results in a *stronger* perfect syllogism.

HOMEWORK: Verify, using Truth Tables, that weakening the assumption of an implication results in a stronger implication:

$$\Big(A \Rightarrow a\Big) \Rightarrow \Bigg(\Big(a \Rightarrow C\Big) \Rightarrow \Big(A \Rightarrow C\Big)\Bigg)$$

HOMEWORK: Verify, using Truth Tables, that strengthening the conclusion of an implication results in a stronger implication:

$$\Big(C \Rightarrow c\Big) \Rightarrow \Bigg(\Big(A \Rightarrow C\Big) \Rightarrow \Big(A \Rightarrow c\Big)\Bigg)$$

[38] as in Felapton, Darapti, and Fesapo

[39] as in Bamalip

Traditionally, reductions without **C** transformations were performed in the direction reversing the flow of logical implication, namely starting with the imperfect syllogism in question and going back to the corresponding perfect syllogism. This process was called[40] `direct reduction`.

Example (direct reduction of "Disamis" syllogism): Consider a "Disamis":

```
Some sea creatures are vertebrates.
All sea creatures can swim.
------------------------------------------------
Some creatures that can swim are vertebrates.
```

"Di<u>s</u>amis" — the first **S** prescribes simplex conversion of the major premise:

```
Some vertebrates are sea creatures.
All sea creatures can swim.
------------------------------------------------
Some creatures that can swim are vertebrates.
```

"Disa**<u>m</u>**is" — the **M** prescribes mutatio:

```
All sea creatures can swim.
Some vertebrates are sea creatures.
------------------------------------------------
Some creatures that can swim are vertebrates.
```

"Disami<u>s</u>" — the last **S** in prescribes simplex conversion of the conclusionm:

```
All sea creatures can swim.
Some vertebrates are sea creatures.
------------------------------
Some vertebrates can swim.
```

Since these transformations retained equivalence of arguments, and we accept as self-evident the validity of the last argument — which is none other than "Darii" [41] — the original "Disamis" syllogism must be valid as well. ♢

[40] somewhat confusingly, given its progression against the flow of implication

[41] as the first letter of "**<u>D</u>**isamis" told us in advance

The reduction of “Baroco” and “Bocardo” was not based on their equivalence to “Barbara” (which we establised in the homework on page 49, demonstrating that the **C** transformation yields an equivalent syllogism in general). Instead, the process called `indirect reduction` was performed. Indirect reduction is an instance of “reductio ad absurdum”, in other words, proof by contradiction. Medieval logicians viewed it as a reduction *through* “Barbara”, rather then reduction *to* “Barbara”.

Example (indirect reduction of “Baroco” syllogism): Take an example of the “Baroco” syllogism:

```
All dogs are mammals.
Some vertebrates are not mammals.
-------------------------------------
Some vertebrates are not dogs.
```

Assume its conclusion is false, meaning that its negation, “all vertebrates are dogs”, is true. If so, then the assumptions of the contra syllogism

```
All dogs are mammals.
All vertebrates are dogs.
-------------------------------
All vertebrates are mammals.
```

are true. But since the contra syllogism is “Barbara” whose validity we accept, it yields the conclusion “all vertebrates are mammals”. This conclusion contradicts the minor premise of the original “Baroco” syllogism, “some vertebrates are not mammals”. This contradiction demonstrates that our assumption — that “Baroco” conclusion was false — does not hold. Therefore, the conclusion of “Baroco” must be true, and the “Baroco” argument as a whole must be valid. $\diamondsuit$

3.5 History: Aristotle

Aristotle was a Greek philosopher and the founder of the Peripatetic School. He is credited with establishing logic as a discipline.

Figure 15: Ἀριστοτέλης [Aristotle] (384–322 BC)

Aristotle may have been the first to recognize that the validity of an argument may result from its mere form rather than meaning. He systematically studied syllogisms and discovered some of the ideas of propositional logic (which historically came after the syllogistic one), like the law of excluded middle and the law of contradiction. Thus, syllogistic logic is also called "Aristotelian logic". Aristotle's treatise of logic, the Organon [2] survived to this day. It provided the foundation for logical studies up to 19th century AD.

The early discovery of this syllogisms by Aristotle and their significance during the antiquity made syllogistic logic one of the cornerstones of liberal arts. The idea of liberal arts [42] can be traced to 4th century BC Greece, where it had at least two distinct roots. The first was the political organization of a Greek polis, or a city-state. Polices had a form of direct democracy that placed great emphasis on the ability of an individual to formulate their ideas and express them in an engaging and convincing way. The second root was

[42]"artes liberalis", literally "the skills of the free" in Latin

the development of mathematics [43] resulting in the idea[44] of mathematical nature of the world.

Roman aristocrat, politician and philosopher Anicius Manlius Severinus Boethius (c480–c524 AD) was the key figure connecting the ancient Aristotelian logic with Christian theology, and preserving the intellectual heritage of Greek philosophy through the European Dark Ages ensuing from the collapse of the Roman Empire. His translation and commentary of Aristotle's works as well as other Greek classics formed the the bulk of *Logica Vetus*, or the "old logic" that served as the foundation for the development of liberal arts in the medieval Europe up to 11th century AD.

(By 9th century AD liberal arts were organized into the Trivium (grammar, dialectic and rhetoric) and the Quadrivium (music, arithmetic, geometry and astronomy), with all the seven subjects together comprising philosophy[45]. In the European Renaissance, the disciplines of the Trivium were complemented by history, poetry, ethics and Greek, forming the core of the "Studia humanitatis" or the "humanities" as we know it now.)

Paris philosopher and theologician Peter Abelard (1079–1142 AD), is considered to be the greatest logician since Antiquity. He developed many subjects introduced by Aristotle, like modal logic, temporal logic and truth-functional theory of logical gates. Probably he was the first to realize that Aristotle's treatment of syllogisms implicitly relied on existential presupposition.

Introductiones in Logicam, thought to be written around 1240 by William of Sherwood (1190–1249)[46], is the earliest known source of the syllogism mnemonics.

Rediscovery of some lost works of Aristotle at the dawn of European Renaissance in 13 century AD brought about the era of *Logica Nova*, or "modern logic". in the form of Scholasticism, a philosophy that emphasized joining faith and dialectical[47] reasoning. The idea of dialectics found its later development in Natural Deduction of the Proof Theory, with its characteristic feature of creating the worlds which are permitted to fail in an informative — and thus productive — way. Scholastics used the so-called "critical organic

[43]probably also influenced by the Egyptian school of geometry

[44]expressed explicitly by Pythagoras

[45]φιλοσοφία, philosophia, literally "love of wisdom" in Greek.

[46]this book survives as a single manuscript dating from late 13th century

[47]literally "through conversation" in Greek, meaning "finding the truth through the collision of the opposites"

method" of philosophical analysis. That method was based on Aristotle's Organon and placed a big emphasis on the study of syllogisms.

Contrary to the terminology used to describe Euler-Venn diagrams, Gottfried Wilhelm Leibniz used them to analyze syllogisms long before Euler and Venn.

Figure 16: Gottfried Wilhelm Leibnitz (1646–1716)

In his paper "De Formae Logicae Comprobatione per Linearum ductus", probably written after 1686, Leibniz proposed the creation of a universal language that he called `characteristica universalis` ("universal characteristic" in Latin). That idea inspired Frege to create his Begriffsschrift two hundred years later.

Another major advance in the study of syllogisms came in the works of George Boole. Boole's algebraic treatment of syllogisms in [5] formed the foundation of the algebraization of logic and defined what we now call "Boolean algebra". Boolean algebra studies equations where variables can assume only two possible values: true and false. They are called `boolean variables`.

Figure 17: George Boole (1815–1864)

4 Digression: Lambda Calculus

4.1 Lambda Notation

The `functional notation` "$f(x)$" may denote the value of the function f when that function is given the input x. The same notation "$f(x)$" can also be used — in a different context — to refer to the function f itself. For example, it is customary to say "define a function $f(x) = x^2$". Often the function is not named, but still referred to as "the function x^2" — essentially by what is an unnamed analog of the functional notation "$f(x)$". This possibility of calling the expression with x a "function" creates the basis for taking "$f(x)$" as a notation for a function as well. In this example, we are dealing with the square function whose graph is the familiar parabola:

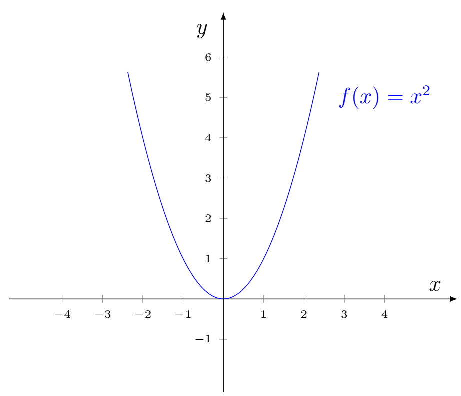


Figure 18: Traditional, but sloppy picture

What happens if $x = 2$? In this case, the "$f(x)$" must stand for $f(x) = 4$ — that's the meaning of the functional notation! But then for an arbitrary x, the "$f(x)$" should denote a number, not a function. So, by using the same notation to denote two different things, we have cornered ourselves into a contradiction.

This sloppiness of notation is tolerated — and even preferred — in the typical mathematical writing. If the intended reader is a human being, assumed to be capable of inferring the precise meaning from the context, and the work is computational, where formulas are repeated again and again, perhaps with minor variations, the sloppiness can be tolerated in the name of making the writing succinct.

Once people began looking into the foundations of mathematcs, the utmost precision became more important than the convenience of carrying long computations in a short hand. The notation highlighting the disticntion in meaning — the Lambda notation — was introduced. It later proved immensely useful in programming, where instructions are written (in part) for a computer incapable — at least initially — of making contextual inferences about the intended meaning. From that exacting standpoint, "$f(x)$" can only denote the output of the function f, and mixing up the function and the formula that defines it, as in "the function x^2", becomes illegitimate.

While the form of Lambda *notation* varies, the *concept* of it is the same: to make it explicit that the whole function, rather than its single output value, is being referred to. This concept, regardless of how it is denoted, is called the `Lambda abstraction`.

The original notation introduced by Alonso Church was $\lambda x.x^2$. It used the Greek letter λ, called "Lambda", giving the name to this whole concept. Modern mathematical texts more often denote the same by $x \mapsto x^2$. In these notes, we will denote the Lambda abstraction as

$$\left(x : x^2\right).$$

Whenever possible to do so without confusion, we will omit the outer parentheses. Modern computer programming languages usually express this concept with something more verbose:

```
function ( x ) {
    return x * x ;
}
```

although the shorter form `x => x*x` can sometimes be used as well.

One advantage of the more verbose programming notation (and perhaps the main reason for its use programming) is the possibility of using its slight modification for giving a name to the function being defined.[48] The mathematical idea of defining the function f by the equation $f(x) = x^2$ can be expressed in programming as

```
function f ( x ) {
    return x * x;
}
```

[48]Lambda calculus does have the expressive power to represent the idea of naming *constant* objects. We will discuss it in more details in the section devoted to contexts and models, which starts on page **??**.

The named function can be applied, whenever we need it, using the notation `f( 5 )`. We could use any other name in place of "`f`". For example, if we define

```
function square ( x ) {
    return x * x;
}
```

then the "`square`" and "`f`" will be the same. We will borrow this format for naming things when we introduce our own notation for proofs later in these notes.

Note also that the above program snippet is equivalent to

```
square = function ( x ) {
    return x * x;
}
```

With Lambda notation, we can make things precise:

$$\Big(x : f(x)\Big)$$

describes the whole function that takes any input x into the output $f(x)$, and $f(x)$ by itself denotes the value of the function f that corresponds to the input x. In our example, $(x : x^2)$ is the whole parabola, and x^2 is an individual number on the y-axis, which is the square of some other number x on the x-axis:

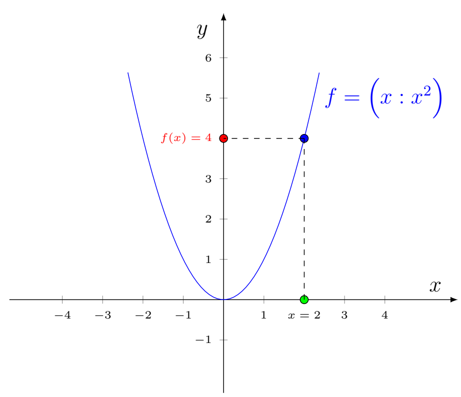


Figure 19: Corrected picture

4.2 Lambda Calculus Grammar

The untyped Lambda calculus deals with expressions constructed according to the following grammar rules:

<term> ::= <variable> | <abstraction> | <application>

<abstraction> ::= (<variable> : <term>)

<application> ::= <abstraction> (<term>)

<variable> ::= a | b | c | d | ...

I hope you recognize that the abstraction is exactly the Lambda notation, and the application is the functional notation.
Example (well-formed Lambda term):

$$\Big(x : x(y)\Big) \Big((z : z)\Big)$$

The first set of big parentheses defines a function $\Big(x : x(y)\Big)$ which takes a function and applies it to some fixed y. Note that this is a function on functions! The second set of big parentheses contains the identity function $(z : z)$ which spits back its input as its output. So, the whole thing says: evaluate with the input y the identity function. Thus this expression must equal to y. ♢

In pure Lambda calculus, all objects are terms, in other words Lambda expressions themselves. However, in "the real world", Lambda calculus is usually used not in its pure form, but as an addition to some other underlying reality[49]. In these notes, we allow ourselves to use variables, constants and externally defined functions which are not Lambda terms. This will permits us to talk about things like the square function $\Big(x : x^2\Big)$ and numbers, as in

$$\Big(x : x^2\Big)(5).$$

This last expression says "apply the square function to 5", so it must equal 25.

[49]This is very similar to the distinction between axiomatic set theory, where every object is a set, and the "naive" set theory, which is permitted to consider some external objects, called `urelements`, as in $\{1, 2, 5\}$.

4.3 Lambda Calculus Reductions

One Lambda expression can be turned into another — simpler — one. It can be done in a sequence of steps called `lambda-reductions`. These Lambda reductions are purely syntactic rules. The truly remarkable fact about them is not so much what they are, but their ability to encompass the meaing of the word "computation". When Lambda calculus was turned into a programming language, it was discovered that computation can be viewed as Lambda reduction of the original Lambda term yielding the result of the computation as the reduced form of that Lambda term. The precise definition of Lambda reduction is a bit technical, so instead of diving into those technical details, we illustrate the Lambda reduction rules with examples.

1. β-`reduction`. Example:

$$\left(x : x^2\right)(y) = y^2.$$

In other words, if our function takes x into x^2, then when applied to y it will take it to y^2. This is the precise meaning of the idea of substitution.

2. α-`conversion` (sometimes called α-`equivalence`). Example:

$$\left(x : x^2\right) = \left(y : y^2\right).$$

It says that a function is defined by its action on its input, not by how its input is denoted. This allows us to choose arbitrary (and preferably meaningful) names for our variables.

3. η-`reduction`. Example:

$$\left(x : f(x)\right) = f.$$

This is merely saying that the abstraction defining the function that takes x into $f(x)$ does not define anything new: it is the same thing as the function f itself.

HOMEWORK: Compute

$$\Big(y : (x : \sqrt{x})(y)\Big)\,(49).$$

4.4 Substitution Instance

Definition (of `substitution instance`): Suppose $f = \Big(x : f(x)\Big)$ is a function and c is an object. The expression $f(c)$ is called a `substitution instance` of f. ♢ (Using this terminology we can say that Lambda notation resolves the distinction between the function as a whole and its particular substitution instance.) The same expression may be a substitution instance in different ways.

Example (ambiguity of substitution instance): Suppose c is a constant, and f is a function that depends on two variables. (For example, f can be the arithmetic operation of addition: $f(x, y) = x+y$.) Define the following three functions, each depending on a single variable:

$$l = \Big(x : f(x, c)\Big) \qquad d = \Big(x : f(x, x)\Big) \qquad r = \Big(x : f(c, x)\Big).$$

The expression $f(c, c)$ is a substitution instance of each one of the three:

$$f(c, c) = l(c) = d(c) = r(c).$$

We need to remember about this possibility when talking about substitution instances. ♢

4.5 History: Schönfinkel, Curry, Church, Kleene, Rosser, McCarthy

Elements of `Lambda calculus` appeared earlier, but as a complete system, it was invented by Alonzo Church [8]. He envisioned it as a framework for building foundations of mathematics.

Figure 20: Alonzo Church (1903–1995)

Church was building upon the work of Моисей Эльевич Шейнфинкель [Moses Schönfinkel] who invented combinatory logic [37] and Haskell Curry, who developed it [9].

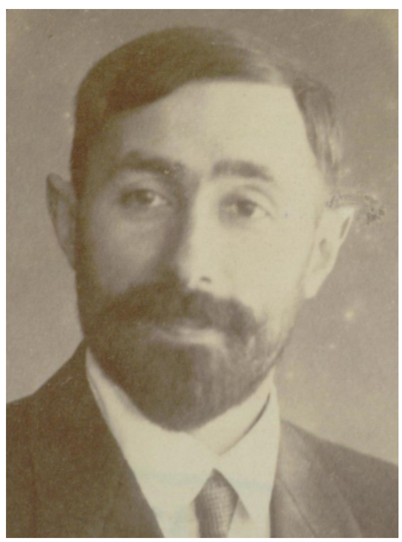


Figure 21: Моисей Эльевич Шейнфинкель [Moses Schönfinkel] (1889–1942)

Figure 22: Haskell Brooks Curry (1900–1982)

However, their original hope came to a crushing defeat: in 1935 Stephen Kleene and J. B. Rosser presented the Kleene–Rosser paradox [26] demonstrating inconsistency of the combinatory logic and Lambda calculus[50].

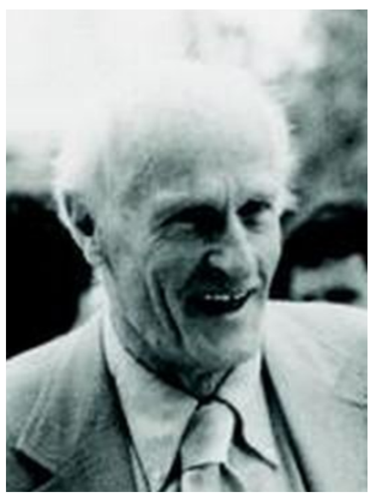


Figure 23: Stephen Cole Kleene (1909–1994)

[50]perhaps more precisely, they demonstrated inconsistency of *modeling logic* within combinatory and Lambda calculus, rather than inconsistency of those two by themselves.

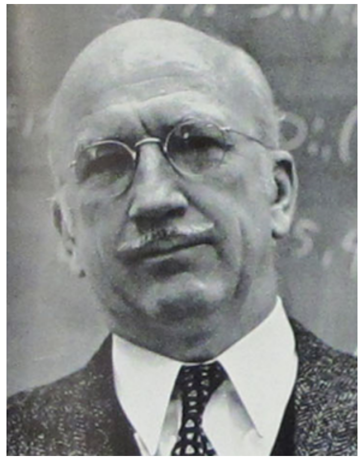


Figure 24: John Barkley Rosser, Sr. (1907–1989)

Kleene–Rosser construction was simplified by in 1942 Curry himself [10] and became known as the Curry's paradox[51].

Lambda calculus was fixed by Church in 1936 by introducing `types`. That later development of the theory became known as the `simply-typed Lambda calculus`.

[51]This footnote, describing the Curry's paradox, is definitely not for the first reading of this text. The original intended use of combinatory logic and untyped Lambda calculus was to model predicate logic within these two theories. Those models would give the obvious — predicate — interpretation to the application terms, so that $P(X)$ would mean "X has the property P". To model implication, a constant Ξ "ur"-term was added, with $\Xi\left(A(B)\right)$ given the meaning "$A \Rightarrow B$". Within that framework, all logical gates can be represented by Lambda terms. (In what follows, we just use the logical gate $\neg$ itself, even though we really mean the Lambda term representing it.)

However, both combinatory logic and untyped Lambda calculus have a remarkable object, called the Y-combinator, which, for every term F, has the property

$$Y(F) = F\left(Y(F)\right).$$

In a sense, the Y finds the `fixed point` for any function — this is why it is called the "fixed point combinator".

Using the Y-combinator, define the term $C = Y(\neg)$ — we denote it C in honor of Curry. Then the property of the Y-combinator tells us that

$$\neg C = \neg\left(Y(\neg)\right) = Y(\neg) = C,$$

which is a contradiction.

Albeit not in its originally intended role, the early — untyped — version of Lambda calculus proved to be extremely useful as a foundation of computer science and a particularly convenient and elegant model of computability. John McCarthy's invention [29] of the computer programming language Lisp, based on Lambda calculus, in the late 1950's, gave a physical embodiment to that model. Other programming languages similar in spirit to Lambda calculus have been created; they fall into what is called the `functional programming paradigm`.

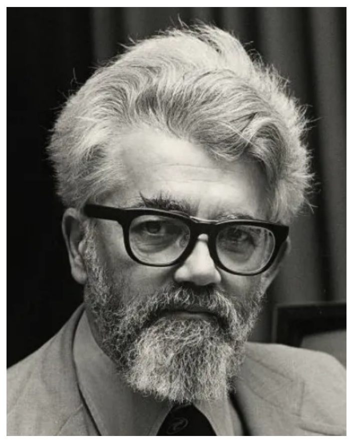


Figure 25: John McCarthy (1927–2011)

5 History Sketch

5.1 Timeline

The content of these notes follows logical progression which is not always the same as the chronological order of the invention of those ideas. For that reason, this section will provide a brief timeline of the development of logic, repeating some of the names and facts mentioned earlier.

Ἐπιμενίδης [Epimenides of Crete], 7th or 6th century BC He is the earliest semi-mythical character with a reference in a later source. Later Greek philosophers attributed the liar's paradox to him. There is no evidence he actually considered it himself — the paradox is based on a fragment of a verse attributed to him that hints at a contradiction. Reformulated for clarity, the paradox goes like this. Epimenides says: "Cretans always lie." But he is a Cretan himself. Is his sentence true or false? Three distinct ideas already emerge here. The first one is the idea of self-reference — the shadow of the ouroboros. Then, this paradox already centers on the issue of a *statement* being *true or false*, thus anticipating the framework of propositional logic. Finally, in a somewhat implicit way, this paradox hints at the possibility of the truth and falsehood being decidable based on the *form* statement itself without any externalities brought to bear.

Σωκράτης [Socrates] (c.470–399 BC) He is the first historical figure whose name is inseparable from the history of logic.

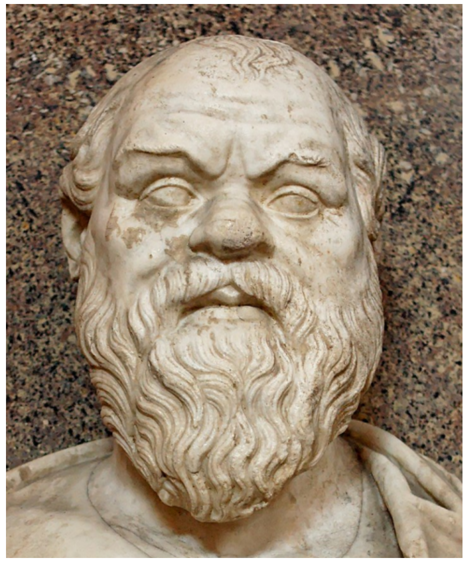


Figure 26: Σωκράτης [Socrates] (c.470–399 BC)

Athens of the fourth century BC had a form of direct democracy that placed a big value on the ability of people to formulate and articulate their ideas, and to use reasoning to convince others of their merit. That environment fostered the culture of public argument and necessitated the study of the general laws of argument we now call logic. Socratic school emerged against that background. While Socrates did not leave any books of his own, he founded a school and one of his students, Πλάτων [Plato] (c.429–c.347 BC), recorded some of the Socrates conversations in [31]. Socratic Dialogues brought rational reasoning in focus and made it a continuing theme in the development of culture.

Εὐκλείδης [Euclid of Megara] (c.435–c.365 BC) He was another pupil of Socrates (who reportedly was present at Socrates' death) Euclid founded the Megarian school of philosophy. The philosophers of that school already considered the liar's paradox, attributing it to Epimenides. Some of Euclid's successors developed logic to such an extent that they became a separate school, that became known as the Dialectical school. The work of the dialectical school on modal logic, logical conditionals, and propositional logic played an important role in the development of logic in antiquity.

Ἀριστοτέλης [Aristotle] (384–322 BC) [pg. 53] Aristotle was a student of Plato who established logic as an independent field. In his work

Organon [2], he fully developed the syllogistic logic, including the categories, predicates, and quantifiers.

Εὐκλείδης [Euclid of Alexandria] (c.325–c.265 BC) Euclid was a Greek mathematician who lived in Ptolemaic Egypt. He used `axiomatic method` in his study of geometry. His Στοιχεῖα [Elements] [11], a mathematical treatise consisting of 13 books, summarized all mathematics known at that time and became the standard for a rigorous treatment of any subject for the next millennia.

Χρύσιππος [Chrysippus of Soli] (c.279–c.204 BC) [pg. 27] Chrysippus was a student of Aristotle who succeeded him as the head of the Peripatetic School. He perfected the discipline of propositional logic, but only fragments of his works survive to this day [7].

Dissolution of the Roman empire resulted in the time of great upheaval in Europe, and many cultural treasures were lost. Fortunately, many of the ancient ideas and sources were preserved by the Islamic scholars, and reemerged in the medieval Europe around the turn of the first millennium. Medieval scholasticism placed a big emphasis on study of syllogisms.

Gottfried Wilhelm Leibnitz (1646–1716) [pg. 55] Leibniz was a German philosopher who co-invented, with Isaac Newton, the Mathematical Analysis. His approach was based on the concept of "monads" that represented `infinitesimals`. While intuitive and for this reason favored by physicists, the concept of infinitesimals looked problematic to generations of mathematicians that followed Leibnitz. The traditional foundation of analysis avoided infinitesimals and relied instead on the machinery of inequalities developed by Weierstrass and Cauchy. Only the new logical advances of Abraham Robinson around 1960 resolved these difficulties and restored infinitesimals to a fully legitimate status. In logic, he used what we call "Euler-Venn diagrams" to analyze syllogisms, and put forward, sometime after 1686, the idea of characteristica universalis. (That idea inspired Frege to create his Begriffsschrift.)

1847 — George Boole (1815–1864) [pg. 56], a self-taught British scientist, invents what we now call `Boolean algebra` and used it in [5] to study syllogisms with algebraic methods.

1873,74 — Georg Cantor (1845-1918), a German mathematician, outlined the basics of infinite set theory. His original theory suffered from the same problem as Begriffsschrift of Frege, invented just a few years later. However, his theory became "the garden of Eden" for mathematicians, providing both the framework for building all other mathematical concepts, and a challenge and focus of efforts on the foundation of the subject. These efforts culminated in several axiomatic set theories.

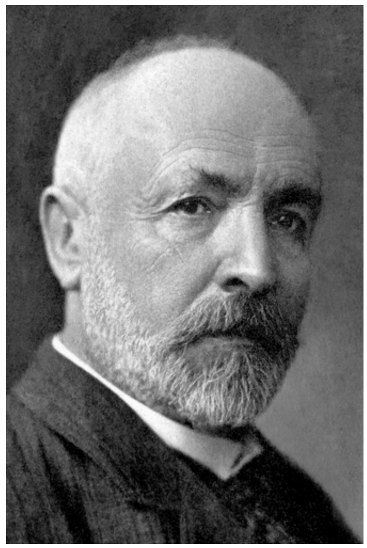


Figure 27: Georg Ferdinand Ludwig Philipp Cantor (1845–1918)

1879 — Friedrich Ludwig Gottlob Frege (1848–1925) [pg. **??**], a German mathematician, invents [13] the "Begriffsschrift" and opens a new chapter in logic.

1889 — Giuseppe Peano (1858–1932), an Italian mathematician, publishes a logical definition of natural numbers (Peano axioms of arithmetic) in his book [30].

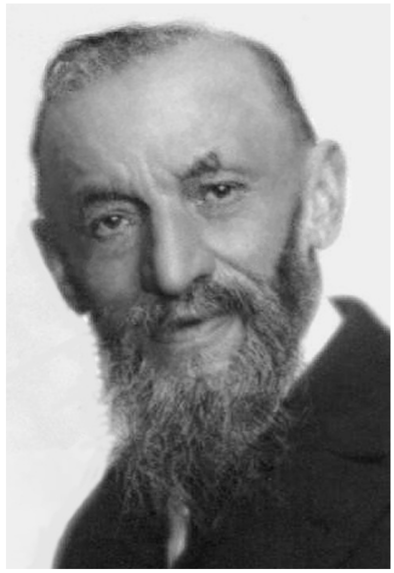


Figure 28: Giuseppe Peano (1858–1932)

1897 — Cesare Burali-Forti (1861–1931) an Italian mathematician, publishes a result [6] that (unknowingly to author) shows inconsistency of Cantor's set theory [52]. This result foreshadows the Russel's paradox that came 5 years later.

[52] That result is now known as the Burali-Forti's paradox. Assuming that the set $\mathcal{O}$ of all ordinal numbers existed, Burali-Forti proved that $\mathcal{O}$ must be well-ordered itself, and thus be its own member: $\mathcal{O} \in \mathcal{O}$, implying that $\mathcal{O}$ is smaller than $\mathcal{O}$ — which is a contradiction.

Figure 29: Cesare Burali-Forti (1861–1931)

1902 — Bertrand Arthur William Russell (1872–1970) [pg. **??**], a British philosopher, sends a letter [36] to Frege which contains what is now known as the "Russell's paradox". Initiates the study of Mathematics foundations with Principia Mathematica.

David Hilbert (1862–1943) David Hilbert (1862–1943) and Wilhelm Ackermann (1896–1962). Grundzüge der theoretischen Logik (Principles of Mathematical Logic). Springer-Verlag efforts in logic [22]

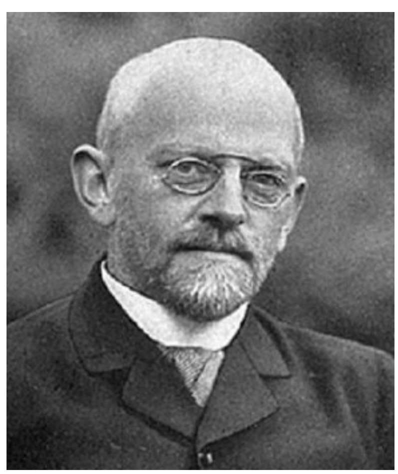


Figure 30: David Hilbert (1862–1943)

1924 — Моисей Эльевич Шейнфинкель [Moses Schönfinkel] (1889–1942), [pg. 63] a Soviet mathematician, student of Hilbert and a member of the Göttingen Logic School, invents `combinatory logic` as the framework for foundations of mathematics.

Jan Łukasiewicz (1878–1956)

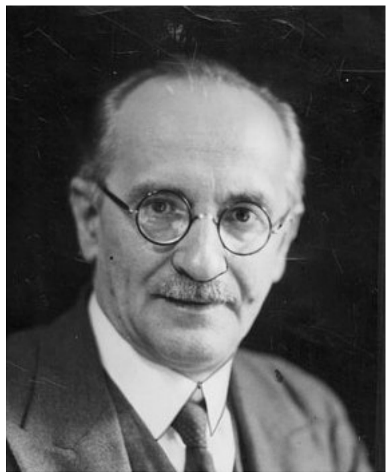


Figure 31: Jan Leopold Łukasiewicz (1878–1956)

One of the founding fathers of the Lwów-Warsaw logic school.

Figure 32: Kazimierz Twardowski, Jan Łukasiewicz, Alfred Tarski, and Stanisław Leśniewski - Warsaw University Library

In his 1926 seminars, made an observation that “real” mathematicians don’t prove their theorems using the logical theories known at the time (including those by Łukasiewicz himself, Frege, and Hilbert). Poses the challenge to his colleagues to create a system that can be used in the real world.

1927 — Stanisław Jaśkowski (1906–1965) [pg. **??**], a Polish logician (and a student of Łukasiewicz) who accepted the challenge posed by his mentor, communicates his first (graphical) form of Natural Deduction at the First Polish Mathematical Congress [24].

1930 — Jacques Herbrand (1908–1931) [pg. **??**], a French mathematician, introduces Herbrand semantics in his thesis.

1930 — Haskell Brooks Curry (1900–1982) [pg. 64] publishes his paper [9] on combinatory logic.

1931 — Kurt Friedrich Gödel (1906–1978), at the time — an Austrian mathematician, publishes [16] the result now known as the “Gödel incompleteness theorem”. His result shows the limits of formal methods and curbs Hilbert’s hopes for axiomatization of mathematics.

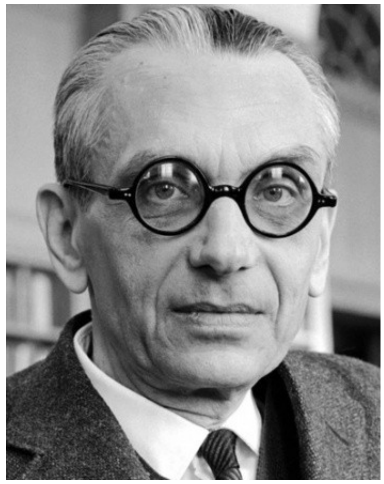


Figure 33: Kurt Friedrich Gödel (1906–1978)

1932 — Alonzo Church (1903–1995) [pg. 63], an American mathematician, invents the untyped lambda calculus [8].

1934 — Gerhard Gentzen (1909–1945) [pg. **??**] a German mathematician and a student of Hilbert, publishes [14] descriptions of several versions of Natural Deduction. Gentzen presents three different systems of deduction, including one for `intuitionistic logic`. Same year, 1934, Jaśkowski publishes his description of Natural Deduction, which is an independent effort from that of Gentzen [25], (See the comparison below.)

1935 — Stephen Cole Kleene (1909–1994), John Barkley Rosser, Sr. (1907–1989) [pg. 64], American mathematicians, present [26] what became to be known as the "Kleene-Rosser paradox", demonstrating inconsistency of logic model within combinatory and (untyped) lambda calculus.

??? — Haskell Brooks Curry (1900–1982) [pg. 64] simplifies Kleene-Rosser construction and presents the Curry's paradox showing inconsistency of any logic model within a system possessing a Y-combinator.

1936 — Alan Mathison Turing (1912–1954) describes the Universal Turing machine model of computation in [38]. This Universal Turing Machine provides an alternative model of computation to Church's lambda calculus, and while less suitable for human use, is immediately realizable in the physical world. It becomes the dominant model of computation until higher level programming languages start to take hold in 1950's.

Figure 34: Alan Mathison Turing (1912–1954)

1937 — Willard Van Orman Quine (1908–2000) an American philosopher, publishes [32] the “New Foundations” of the set theory.

1942 — John Barkley Rosser, Sr. (1907–1989) [pg. 65] an American mathematician, finds in [35] that Burali-Forti paradox applies to Quine’s “New Foundations” necessitating a revision of that set theory (by Quine himself).

1952 — Frederic Brenton Fitch (1908–1987) [pg. **??**] an American logician, introduces his Natural Deduction notation in the textbook [12].

1959 — John McCarthy (1927–2011) [pg. 66] an American mathematician and computer scientist, invents the computer programming language Lisp by modeling it after Church’s lambda calculus.

Abraham Robinson (1918–1974) Using model theory, Robinson was able to build a solid logical foundation for the classic — but held suspect for hundreds of years — infinitesimals-based approach to Mathematical Analysis. [33]

Figure 35: Abraham Robinson (1918–1974)

1965 — John Alan Robinson (1930–2016) a British-American mathematician, discovers the `resolution principle` and describes it in [34].

Figure 36: John Alan Robinson (1930–2016)

early 1970's — Robert Anthony Kowalski (1941–) a British-American mathematician, lays the theoretical foundations for the Prolog language, see e.g. [27]

Figure 37: Robert Anthony Kowalski (1941–)

5.2 Concluding Remarks

There are several different types of contributors and contributions in the above timeline. People who fostered the creation of new schools (Socrates, Aristotle, Hilbert, Łukasiewicz) not only advanced the subject itself, but gave cultural development an impulse that often persisted for many generations after them [53]. Masters of other fields who did not have logic as the main focus in all of their endeavors (Euclid of Alexandria, in some ways Leibnitz, Peano, to some extent Hilbert in his geometry works), — but wanted to *be* logical in their studies of other subjects: they moved the stake posts of logic into the new territory and filled the subject with the vital energy of its applications, giving the logicians that followed them the new spaces in which the subject could develop. There are those (Cantor, Frege, Schönfinkel, Hilbert, Curry, Church, Quine) who eagerly expanded the raw expressive power of logic — and those (Burali-Forti, Russell, Gödel, Kleene, Rosser) who pruned some of the wilder branches that ended up connecting truths and falsehoods, thus creating the short circuits of contradictions...

This unending cycle of generation and destruction, looped into a contradiction by its own self-reference, is driven by the conflict between the expansion of logic in its attempt to encompass the forever growing realms of human knowledge on the one hand, and taming of its power, forced by the need to guarantee the separation between the truth and the falsehood — on the other. While this cycle goes on, we are done — and we came back to our beginning.

[53] sometimes — as in the case of Aristotle — even *restarting* after a long hiatus.

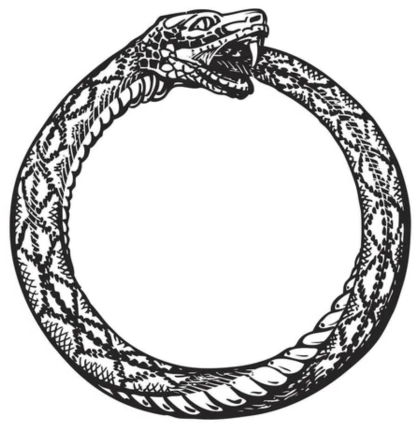


Figure 38: The Uroboros

5.3 Appendix: Historical Examples of Different Notations

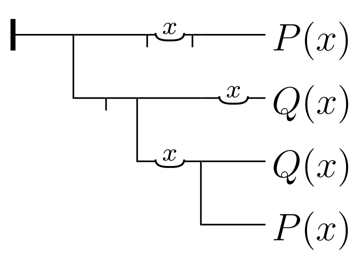
Consider a predicate logic argument:

$$\left(\left(\forall x : \left(P(x) \Rightarrow Q(x)\right)\right) \wedge \left(\exists x : \left(\neg Q(x)\right)\right)\right) \Rightarrow \left(\exists x : \left(\neg P(x)\right)\right)$$

This argument can be presented in the style of Gerhard Gentzen which we used before already:

$$\begin{array}{c} \forall x : \left(P(x) \Rightarrow Q(x)\right) \\[2ex] \exists x : \left(\neg Q(x)\right) \\ \hline \exists x : \left(\neg P(x)\right) \end{array}$$

Gentzen was the one who introduced the notations $\forall, \exists, \wedge$, and $\vee$, so it is not surprising that this form of writing looks very modern. The same argument would look completely differently in the amusingly idiosyncratic style of Friedrich Ludwig Gottlob Frege:



However, you may notice something familiar even here: the negation $\neg$ and the "turnstile" $\vdash$ that has been used ever since for an assertion of truth of some statement are the two symbols introduced by Frege which are still in use today.

Assuming now that we want to prove this argument, how would a natural deduction proof look in different notation systems?

1.		$\forall x : \Big(P(x) \Rightarrow Q(x)\Big)$	hypothesis
2.		$\exists x : \Big(\neg Q(x)\Big)$	hypothesis
3.	a	$\neg Q(a)$	sub-proof hypothesis
4.		$P(a) \Rightarrow Q(a)$	$\forall x$ elimination from 1 (taking $x = a$)
5.		$P(a)$	sub-sub-proof hypothesis
6.		$Q(a)$	$\Rightarrow$ elimination from 4, 5
7.		F	contradiction from 3, 6
8.		$\neg P(a)$	reduction ad absurdum from 5–7
9.		$\exists x : \Big(\neg P(x)\Big)$	$\exists$ introduction from 8
10.		$\exists x : \Big(\neg P(x)\Big)$	$\exists$ elimination from 2, 3–9

Figure 39: A Proof in the Style of Stanisław Jaśkowski

1. $\forall x : \Big(P(x) \Rightarrow Q(x)\Big)$	
2. $\exists x : \Big(\neg Q(x)\Big)$	
3. $\boxed{a}$ $\neg Q(a)$	
4. $P(a) \Rightarrow Q(a)$	$\forall$ **Elim:** 1
5. $P(a)$	
6. $P(a) \Rightarrow Q(a)$	**Reit:** 4
7. $P(a)$	**Reit:** 5
8. $Q(a)$	$\Rightarrow$ **Elim:** 6, 7
9. $\neg Q(a)$	**Reit:** 3
9. $\bot$	$\bot$ **Intro:** 8, 9
10. $\neg P(a)$	$\neg$ **Intro:** 5–9
11. $\exists x : \Big(\neg P(x)\Big)$	$\exists$ **Intro:** 10
12. $\exists x : \Big(\neg P(x)\Big)$	$\exists$ **Elim:** 2, 3–11

Figure 40: A Proof in the Style of Frederic Brenton Fitch

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{1}{\neg Q(a)}\,\text{Hyp}\quad\dfrac{\dfrac{\forall x:\Big(P(x)\Rightarrow Q(x)\Big)\quad\dfrac{2}{a:term}\,\text{Hyp}}{P(a)\Rightarrow Q(a)}\,\forall\text{Elim}\quad\dfrac{3}{P(a)}\,\text{Hyp}}{Q(a)}\,{\Rightarrow}\text{Elim}}{F}\,\neg\text{Elim}}{\neg P(a)}\,\text{Contr (3)}}{\exists x:\Big(\neg P(x)\Big)}\,\exists\text{ Intro}\qquad\exists x:\Big(\neg Q(x)\Big)}{\exists x:\Big(\neg P(x)\Big)}\,\exists\text{ Elim (1, 2)}$$

Figure 41: A Proof in the Style of Gerhard Gentzen

```
implication example( predicate P, Q )[
        assertion impl =
            implication( object x )[ P( x ) ]{ Q( x ) };
        assertion conc_neg = ( exists x: !Q( x ) );
    ]{
        implication this_implies_existence( object a )[
                assertion this_conc_neg = ( !Q( a ) )
            ]{
            assertion this_impl =
                deduce( implication()[ P( a ) ]{ Q( a ) } )[
                    impl( a )
                ];
            assertion thisass_impl_falsehood =
                deduce( implication()[ Q( a ) ]{ F } )[
                    negelim( Q( a ) )[ this_conc_neg ]
                ]
            implication absurdum()[
                    assertion this_prem = ( P( a ) )
                ]{
                assertion thisconc = deduce( Q( a ) )[
                    thisimpl()[ this_prem ]
                ];
                assertion falsehood = deduce( F )[
                    thisass_impl_falsehood()[ thisconc ]
                ]
            }
            assertion this_ass_false = deduce( !P( a ) )[
                negintro( P( a ) )[ absurdum ]
            ]
            deduce( exists x: !P( x ) )[
                exintro( a; x: !P(x) )[ this_ass_false ]
            ]
        }
        deduce( exists x: !P( x ) )[
            exelim( x: !P(x); exists x: !P(x) )[
                this_implies_existence;
                conc_neg;
            ]
        ];
}
```

But wait...The main strength of our notation was its ability to define new arguments that can be plugged into the old ones. So, in a way, the proof on the previous page is an unfair comparison that does not play to the strength of what we used in these notes. Given that we have already proven modus tollens `MT` on page **??**, we can prove this argument faster:

```
implication example( predicate P, Q )[
        assertion impl =
            implication( object x )[ P( x ) ]{ Q( x ) };
        assertion conc_neg = ( exists x: (!Q( x ) ) );
    ]{
        implication this_implies_existence( object a )[
                assertion this_conc_neg = ( !Q( a ) )
            ]{
            assertion this_impl =
                deduce( implication()[ P( a ) ]{ Q( a ) } )[
                    impl( a )
                ];
            assertion this_ass_false = deduce( !P( a ) )[
                MT( P( a ), Q( a ) )[ this_impl; this_conc_neg; ]
            ]
            deduce( exists x: !P( x ) )[
                exintro( a; x:!P(x) )[ this_ass_false ]
            ]
        }
        deduce( exists x: !P( x ) )[
            exelim( x:!P(x); exists x:!P(x) )[
                this_implies_existence;
                conc_neg;
            ]
        ];
}
```

References

[1] The grounding of elementary number theory. [28], pages 266–273. in [28].

[2] Aristotle. *Ὄργανον ["Organon", The Instrument]*. 4th Century BC. URL: https://www.rbjones.com/rbjpub/philos/classics/aristotl/obook_draft.pdf.

[3] Jonathan Barnes. *Logical Matters: Essays in Ancient Philosophy II*. Oxford University Press, 2012. URL: https://www.wiko-berlin.de/fileadmin/Jahrbuchberichte/1984/1984_85_Barnes_Jonathan_Jahrbuchbericht.pdf.

[4] Michael Beaney, editor. *The Frege reader*. Blackwell Publishing, 1997. URL: https://dl1.cuni.cz/pluginfile.php/767005/mod_resource/content/0/Frege%20-%20Reader.pdf.

[5] George Boole. *The Mathematical Analysis of Logic*. Cambridge: MacMillan, Barclay, & MacMillan; London: George Bell., 1847. URL: https://www.gutenberg.org/files/36884/36884-pdf.pdf.

[6] Cesare Burali-Forti. Una questione sui numeri transfiniti. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 11:154–164, 1897. URL: https://zenodo.org/record/2362091/files/article.pdf.

[7] Chrysippus. *Logical Investigations (Surviving Fragments)*. In [3], 3rd Century BC. URL: https://www.wiko-berlin.de/fileadmin/Jahrbuchberichte/1984/1984_85_Barnes_Jonathan_Jahrbuchbericht.pdf.

[8] Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(2):346–366, 1932. URL: https://raw.githubusercontent.com/emintham/Papers/master/Church-%20A%20Set%20of%20Postulates%20for%20the%20Foundation%20of%20Logic.pdf.

[9] Haskell Brooks Curry. Grundlagen der Kombinatorischen Logik [foundations of combinatorial logic]. *American Journal of Mathematics*, 52:509–536, 1930.

[10] Haskell Brooks Curry. The Inconsistency of Certain Formal Logics. *Journal of Symbolic Logic*, 7:115–117, 1942.

[11] Euclid. *Στοιχεῖα [The Elements]*. c.300 BC. URL: https://farside.ph.utexas.edu/books/Euclid/Elements.pdf.

[12] Frederic Brenton Fitch. *Symbolic Logic: An Introduction*. New York: The Ronald Press Company, 1952. URL: https://usa1lib.org/book/2725862/ae4b46.

[13] Gottlob Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens [Concept Script, a formal language of pure thought modelled upon that of arithmetic]*. Verlag von Louis Nebert, Halle, 1879. URL: https://gdz.sub.uni-goettingen.de/download/pdf/PPN538957069/PPN538957069.pdf, also a translation at: https://www.informationphilosopher.com/solutions/philosophers/frege/Frege_Begriffsschrift.pdf.

[14] Gerhard Gentzen. Untersuchungen über das logische Schließen [investigations into logical deduction], I and II. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. URL: https://pic.blog.plover.com/math/logic/lk/Gentzen1934.pdf, also a translation at: https://logic-teaching.github.io/prop/texts/Gentzen%201969%20-%20Investigations%20into%20Logical%20Deduction.pdf.

[15] Kurt Gödel. *Über die Vollständigkeit des Logikkalküls [On the Completeness of the Calculus of Logic]*. 1929.

[16] Kurt Friedrich Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik*, 38:173–98, 1931.

[17] David Hilbert. Les principes fondamentaux de la géométrie. *Annales scientifiques de l'École Normale Supérieure*, 17:103–209, 1900.

[18] David Hilbert. Axiomatisches denken. *Mathematische Annalen*, 78:405–415, 1917.

[19] David Hilbert. Die logischen grundlagen der mathematik. *Mathematische Annalen*, 88:151–165, 1923.

[20] David Hilbert. Über das unendliche. *Mathematische Annalen*, 95:161–190, 1926. URL: https://gdz.sub.uni-goettingen.de/download/pdf/PPN235181684_0095/PPN235181684_0095.pdf.

[21] David Hilbert. Probleme der grundlegung der mathematik. *Mathematische Annalen*, 102:1–9, 1930.

[22] David Hilbert. Die grundlegung der elementaren zahlenlehre. *Mathematische Annalen*, 104:485–494, 1931.

[23] Ackermann, Wilhelm Hilbert, David. *Grundzüge der theoretischen Logik [Principles of Mathematical Logic]*. Springer-Verlag, Berlin, 1928. URL: https://github.com/mdnahas/Peano_Book/blob/master/Peano.pdf.

[24] Stanisław Jaśkowski. Teoria dedukcji oparta na regułach założeniowych [theory of deduction based on suppositional rules]. *Księga pamiatkowa pierwszego polskiego zjazdu matematycznego, 1927 [Proceedings of the First Polish Mathematical Congress, 1927]*, page 36, 1929.

[25] Stanisław Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, 1:5–32, 1934. URL: https://www.logik.ch/daten/jaskowski.pdf.

[26] Rosser, John Barkley, Sr. Kleene, Stephen Cole. The inconsistency of certain formal logics. *Annals of Mathematics*, 36:630–636, 1935.

[27] Robert A. Kowalski. Predicate logic as programming language. November 1973.

[28] Paolo Mancosu, editor. *From Brouwer to Hilbert: The debate on the foundations of mathematics in the 1920s*. Oxford University Press, New York, 1998.

[29] John McCarthy. Recursive Functions of Symbolic Expressions and their Computation by Machine. 1959.

[30] Guiseppe Peano. *Arithmetices Principia Nova Methodo Exposita [The principles of arithmetic presented by a new method]*. Turin: Bocca Brothers, 1889. URL: https://github.com/mdnahas/Peano_Book/blob/master/Peano.pdf.

[31] Plato. *Σωκρατικὸς λόγος [Socratic Dialogues]*. between 399 and 387 BC. URL: https://webs.ucm.es/info/diciex/gente/agf/plato/The_Dialogues_of_Plato_v0.1.pdf.

[32] Willard Van Orman Quine. New Foundations for Mathematical Logic. *The American Mathematical Monthly*, 44:70–80, 1937.

[33] Abraham Robinson. *Non-standard Analysis*. Princeton University Press, 1966.

[34] John Alan Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Communications of the ACM*, 12:23–41, 1965. URL: https://dl.acm.org/doi/pdf/10.1145/321250.321253.

[35] Sr. Rosser, John Barkley. The Burali-Forti paradox. *Journal of Symbolic Logic*, 7:1–17, 1942.

[36] Bertrand Russell. Letter to Frege, of june 16, 1902, informing him of the contradiction in Grundgesetze der Arithmetik that has come to be known as Russell's paradox. [4]. URL: https://dl1.cuni.cz/pluginfile.php/767005/mod_resource/content/0/Frege%20-%20Reader.pdf.

[37] Moses Schönfinkel. Über die Bausteine der mathematischen Logik [on the building blocks of mathematical logic]. *Mathematische Annalen*, 92:305–316, 1924.

[38] Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society (published 1937)*, 2:230–265, 1936.

[39] Ludwig Wittgenstein. Logisch-Philosophische Abhandlung. 1919. URL: https://www.wittgensteinproject.org/w/index.php?title=Logisch-philosophische_Abhandlung_(Darstellung_in_Baumform),.

[40] Ludwig Wittgenstein. Tractatus Logico-Philosophicus. 1922. URL: https://www.wittgensteinproject.org/w/index.php?title=Tractatus_Logico-Philosophicus_(English),.

Index