

Some of the Big Ideas in
Philosophy, Logic and Set Theory

Joseph Cunningham

Contents

Chapter 0

Preface

Iverson bracket.

Definite description operator

Use

$$\begin{cases} x & \text{condition 1} \\ y & \text{condition 2} \end{cases}$$

as inline ternary operator. I.e. it should return an expression.

Part I

Introduction

Chapter 1

What are you reading?

Chapter 2

Which philosophy? Why philosophy?

I will give arguments. On which basis of truth you choose to accept these arguments, or not, is up to you.

Part II

Some Philosophy

Chapter 1

Some thoughts on Philosophy

Chapter 2

Epistemology

Chapter 3

Metaphysics

Chapter 4

Philosophy of language and truth

Part III

Formal systems

Chapter 1

Formal systems

The ideas in the previous section were expressed somewhat informally. Now an attempt will be made at a more formal discussion of the ideas. This will lead us to the ideas behind formal systems.¹ We will start with a very broad idea and then try to narrow it down by arguing that certain constraints apply and we may disregard certain possibilities or that certain possibilities are equivalent.

1.1 Towards a definition

1.1.1 Transmitting ideas and reasoning about the world

Essentially we want a system that can be used to transmit ideas. A priori it is difficult to know what ideas are.

At this point it must also be stressed that logic is not one monolithic entity. Different “logical languages” or logical systems have been developed for different uses. This may seem strange, because one imagines there is only one true truth. It is however somewhat analogous to the existence of different natural languages. It is possible to say the same thing in either English or French.

In addition there is no logical system that is capable of expressing everything. This means logical systems sometimes need to be extended. For example modal logic adds the ability to qualify statements made in classical logic. Conversely it is often both practically and theoretically useful to keep a logical system as concise as possible, dropping unnecessary extensions. This makes working with the logical system much easier.

In general a logical system is a type of formal system. A formal system consists of a formal language as well as ways to *derive* new statements from old ones. These new statements are conclusions that can be inferred from the given statements which are known as premises.

TODO: arbitrary, circular, abstraction (equivalence and different models)

¹What follows will be quite a personal exposition of formal systems. All of the results are very standard and may be found in any work on mathematical logic, but each author has their own preferred definitions and notation, depending on what problems they are trying to solve. My exact definition of a formal system is one that I have not found anywhere else, but I believe it to be a useful one.

1.2 On the foundation of formal systems

1.2.1 What and why?

We want a system, a method to transmit information and insights between us humans. In physics, that is specifically about the physical world, but for now we will work more generally. In general we could use any system imaginable.

1.2.2 Pre-Socratic meditations

1.2.2.1 Anaximander and the apeiron

1.2.2.2 On change: Heraclitus and Parmenides

1.2.3 Defining formal systems

We are allowed to do anything we can. Whether we should is another question entirely. Relevant slices

1.2.4 The art and science of modeling reality

1.2.5 Iterative development

1.2.5.1 Building physical intuition

1.2.6 A link with thoughts

Formal argument not on paper, but in mind

1.2.7 Model theory

Finally a couple of remarks should be made about the level of generality of a formula. The conception of a formula introduced above is one of a mathematical manifestation of a consequence of a set of general ideas, which in this case are obviously ideas about how the physical world works. We may also call this ensemble of ideas, or any subset of it, a physical model. In general a formula can be a manifestation of a model at any level of generality, from very specific cases to being fully accurate representations of the entire model.

Example

So for instance say we want to know how long it would take an apple to reach the ground if we dropped it off a building. This obviously depends on the height of the building. We may therefore write down a formula that relates the height of the building with the time in free-fall. This may well be a very nice formula, but maybe we want to allow for cases in which the apple is given some initial velocity in the longitudinal direction, i.e. the apple is either thrown upwards or downwards. We can write down a formula for that. This new formula should give the same results as the first formula if we take the initial velocity to be zero. Thus it is more general than the first formula. We can generalise the formula in many ways, maybe allowing the apple to be thrown in different directions or maybe allowing the strength of gravity to change to account for different locations on earth and in space. If we generalise enough eventually we will get Newton's equations of motion. In fact we can generalise further to obtain quantum mechanical or relativistic equations.

These very general formulae can be used to describe entire models.

1.3 A formal definition

1.3.1 The language

1.3.1.1 The apeiron and limitations

intelligibility and resolution

1.3.1.2 Formal languages

A formal language consists of

1. a collection of symbols that may be used to write in the language. This collection is called the alphabet. All “sentences” (i.e. formulae) in the formal system are just strings of characters from the alphabet;
2. a way to specify which strings of characters are actually valid in the language. For example the string of characters “ajf ekjl ljejk jlkkgkgkgddd” is written in the same alphabet as the English language, but is most certainly not English. This is done by giving a set of production rules known as a (formal) grammar.² In logic valid sentences are called well-formed formulae of wffs for short.

These aspects of a formal system are known as the syntax of the formal system.

1.3.1.3 Tokenisation

TODO Leary p.7

1.3.1.4 Grammars

unique readability

1.3.2 Texts: proofs / calculations / derivations

We usually define formal system in order to be able to write texts in them. Depending on the context and the system, such a text is more commonly known as a proof, a calculation or a derivation. We will (somewhat unconventionally) use the term text, in order to be as general as possible.

For a given formal system, a text typically starts with a number of wffs called premises. Then other wffs are added to the page, but only if they follow from some so-called rule of inference. We can write as many as we want. When we are satisfied and have finished writing the text, the last wff is called the conclusion.

²There are also other ways to specify which sentences are valid, for example using a recogniser. This topic is way beyond the scope of a text about physics. Interested readers should find a book on parsing or formal language theory, such as Ullman, Aho (TODO ref)

A text may be summarised by stating only the premises and conclusion, separated by \vdash . Such a summary looks something like

premise 1, premise 2, premise 3 ... \vdash conclusion.

Example

TODO a derivation, a calculation and a solution to a system of equations

The other reason we use the term text is because it is fairly neutral and we do not want to suggest an interpretation at this stage. How a text written in a formal system, and in particular what it means when a formula is written down, is not a property of the formal system itself. It is an interpretation that is given later.

In principle a formal system just tells us which orderings of symbols on a page are valid. Of course in practice we want our formal system to say something true about this reality, but that link to reality, called the semantics (or a semantic interpretation) of the formal system, is only specified later and not really an integral part of the formal system. Although, of course, we will not talk much about formal systems that do not have a semantic interpretation that corresponds to, at least approximately, some part of reality.

1.3.3 Inference rules

Inference rules are rules that specify how to obtain new formulae from certain old ones. They are processes that take a number of wffs as inputs and outputs a new wff. A priori there are no other restrictions on the types of processes that are allowed.

In principle we can specify inference rules however we want. One might use operations from language theory such as slicing and concatenation.

Example

An example of an inference rule is the following:

1. Discard the first three symbols of each premis.
2. Concatenate the resulting formulae.
3. If this formula is well-formed, it is a conclusion that follows from the premises.

This is a perfectly good inference rule, even if it is unlikely to make any semantic sense. Unless the formal language was constructed such that the operations of slicing and concatenation were semantically meaningful.

Example

Another example of an inference rule that is valid for a formal system in which “aab”, “babb” and “abba” are wffs is: If the premises are “aab” and “babb”, then conclude “abba”

This inference rule is very specific. Typically explicitly specifying all the inference rules we want, like this, is impractical because there are too many of them.

Most commonly a rule of inference is stated using a schema (plural: schemata).

1.3.3.1 Schemata and metavariables

In order to define schemata, we need to have a new set of symbols, called metavariables.³ These symbols can be anything, so long as they are not part of the alphabet of the formal system. Typically we will use uppercase letters ($A, B, C \dots$).

Schemata are sequences of symbols that are either part of the alphabet or metavariables. Consequently all sentences in the language of the formal system are also schemata.

Now we can write whole classes of inference rules by giving several schemata as premises and a schema as conclusion. Specific inference rules are then obtained by substituting strings of symbols from the alphabet for each metavariable, such that if two metavariables are the same, their substitution is as well. Additionally we also have to require that each premise and the conclusion are all wffs.

TODO: types This technique is especially powerful because many the grammar of many logical systems is defined such that wffs consist of sub-wffs strung together using connective symbols. In this case many schemata are guaranteed to yield wffs if the metavariables are substituted with wffs.

1.3.4 Axioms and axiom schemata

Some authors allow the definition of a formal system to contain axioms. These are just wffs that may be written down, regardless of which wffs came before it in the text. We will simply view them as inference rules that do not have any premises.

Similarly, axiom schemata are classes of axioms that are specified by a schema.

A tautology is like an axiom in the sense that it may always be written in a text. The difference is that it is not explicitly supplied as an inference rule. Instead it is the consequence of several inference rules.

1.4 Semantics and interpretation

1.5 Formalisms

The word formalism is a favourite of many a physics professor. It is essentially a synonym of “formal system”, but it is typically only used for formal systems where the texts are called calculations.

TODO

³In computer science and programming these variables are called metasyntactic variables.

1.6 Equivalence of formal systems

1.6.1 Computational trinitarianism

1.6.2 On the relativity of truth

1.7 Metasystems

1.7.1 Duality

1.7.2 On the relativity of formal systems as unifying framework

1.7.2.1 Set-theoretic definition

set is formal systems is set ...

1.8 Incompleteness of formal systems

Part IV
(Mathematical) logic

Chapter 1

What is logic?

In this part some useful concepts from logic will be discussed.

The word “logic” comes from the ancient greek word $\lambda\acute{o}\gamma\omicron\varsigma$ that originally meant “word”, but came to mean “thought” and “reason” as well. Most historical surveys of logic start with the ancient Greeks. The big name in this context is Aristotle (384–322 BC). He introduced many core concepts of logic, such as terms, predicates, propositions and syllogisms. Stoic logic was also very influential.

During the medieval period, scholasticism was the dominant method of critical thought. The scholastics were mainly interested in reconciling Christian theology with classical philosophy. For modern logic William of Ockham (1287 – 1347) is of particular note.

The next big innovation (if fact in many ways the first big innovation since Aristotle) is the development of symbolic logic. This development was started in 1854 by George Boole who introduced the principles of what is now called Boolean algebra. In 1879 Gottlob Frege pushes this concept further with the publishing of his *Begriffsschrift*. His aim was to construct a “formula language of pure thought”, such that it would be possible to prove all truths of arithmetic deductively from a limited number of logical axioms.

The theory that all mathematics can ultimately be deduced from purely formal logical axioms is called logicism. This programme was championed by Bertrand Russell and Alfred Whitehead who published *Principia Mathematica* together in three volumes (1910, 1912 and 1913).

Around this time mathematics was suffering its foundational crisis. This is a fascinating topic, both from a technical and historical perspective, but not one I wish to get into here. It suffices to say that from then on logic was of utmost importance to both philosophy and mathematics. When computers started appearing, computer scientist realised that Boolean algebra was a very useful basis on which to build much of theoretical (and less theoretical) computer science and also wanted a piece of the pie. Logic is now taught and studied at three different faculties; consequently there is a lot of competing notation when it comes to logic.

1.1 Platonism, formalism and intuitionism

TODO

1.2 Logical formal systems

1.2.1 Syntactic elements

1.2.2 Semantics

A formal system is essentially only a particular way to write symbols on paper (i.e. a *text*). In theory these symbols do not actually need to mean anything, although in practice we will not consider any formal systems that do not have a useful interpretation. In fact in subsequent chapters the interpretation will guide the construction of the formal systems and suggest inference rules. If a formal system has a particular interpretation, we say it is a model for that interpretation.

The semantics of a formal system essentially tells us what it means for a formula to be written in a text.

Example

In logic, the meaning of a formula having been written in a proof is typically that it is true (assuming the premises are true).

In a calculation the meaning is typically that it has the same value as the premise.

Once that meaning (i.e. the interpretation) has been fixed, it may be possible to *empirically* see that some formulae must be conclusions. These so-called semantic conclusions are written using \models instead of \vdash .

If a formal model accurately models its interpretation then \models and \vdash are interchangeable.

Example

In a logical system that models arithmetic, 5 should be a conclusion of $2 + 3$, because empirically if you take two things and then three, you get five things. Luckily the rules of arithmetic allow us to calculate 5 from $2 + 3$, so in this case the model is accurate.

Chapter 2

Propositional logic

TODO: rewrite. When symbolic logic was first introduced, it was in a form similar to this. The basic building blocks of propositional logic are propositions. They are essentially assertions that may be true or false, for example

- Socrates is bald;
- It is raining;
- Circle A is bigger than circle B ;
- ...

Such assertions are assigned letters (traditionally the letters p, q, r, s, \dots) are used. Propositional logic cannot say anything about elements smaller than a proposition - for example it would not be possible to say that for any circle A there is a circle bigger than A - only whether a proposition is true or false.

Where propositional logic is useful on the other hand, is in stringing multiple propositions together. This can be done using several different connectives. Thus, for example, from the propositions “Socrates is bald” and “Socrates is wearing a hat” we may infer “Socrates is bald **and** is wearing a hat”.

2.1 Syntax

In propositional logic the alphabet consists of three sorts of symbols: letters for primitive propositions, symbols for the logical connectives (like the **and** above) and parentheses for resolving ambiguities:

1. Primitive propositions, denoted p, q, r, s, \dots ;
2. Connectives:
 - (a) \neg which represents a negation,
 - (b) \vee which represents the disjunction **or**
 - (c) \wedge which represents the conjunction **and**,
 - (d) \implies which represents the implication **if ... then** and
 - (e) \iff which represents the equivalence **if and only if**;

3. Parentheses: (and).

We can then define wffs recursively in the following manner:

1. The characters p, q, r, s, \dots are wffs;
2. If A is a wff, then $\neg A$ is a wff;
3. If A and B are wffs, then
 - (a) $A \vee B$,
 - (b) $A \wedge B$,
 - (c) $A \implies B$ and
 - (d) $A \iff B$are wffs;

4. Nothing else is a wff.

The connectives \wedge, \vee, \implies and \iff are called binary because they “connect” two wffs. The connective \neg is called unitary.

Based on this we can rewrite many logical assertions in the language of propositional logic.

Example

If we call “Socrates is bald” p and “Socrates is wearing a hat” q , we can write “Socrates is bald **and** is **not** wearing a hat” as

$$p \wedge \neg q$$

2.2 Inference rules

So far we have not achieved very much; we have a way to partially rewrite some English sentences using obscure symbols. The advantage of the symbolic notation is that it allows us to isolate the logical structure of sentences.

Example

The English sentences “Socrates is bald **and** is **not** wearing a hat”, “Socrates is bald **and** has **not** got a hat on” and “Socrates has **no** hair **and** Socrates is **not** wearing a hat” all mean the same, but are written differently. In English it is no trivial matter to work out in general which sentences mean the same thing.

In propositional logic we have a way to determine if two sentences mean the same thing, using inference rules. We can also use these rules to derive new truths from things we already know.

Example

One of the inference rules we will see is that if we know both p and q to be correct, then we can derive $p \wedge q$.

Using this, we can formally derive “Socrates is bald **and** is wearing a hat” from the propositions “Socrates is bald” and “Socrates is wearing a hat”.

In what follows, a number of inference rules will be listed. In general we will be able to infer new inference rules from old ones, so the list of possible rules is endless. For theoretical purposes it is useful to have a list of inference rules from which all others can be derived, preferably as small as possible. One such list will be given. TODO reconstruction

2.2.1 A minimal list of inference rules

The rules in this list have been chosen because they in some way contain the essence of each connective and thus can define them. Other, equivalent, choices can be made.

- **Implication**

- *Modus ponens*

$$A \implies B, A \vdash B$$

- *Conditional proof*. This introduces the implication \implies . In a proof we may always state a wff as a hypothesis. We then enter a subproof and may then use the inference rules as usual assuming the hypothesis is correct. In order to break out of the subproof, we write $A \implies B$ where A is the hypothesis and B is the result of the subproof. Any wff that has been established in the main proof may be reiterated in the subproof.

- **Conjunction**

- *Conjunction introduction*

$$A, B \vdash A \wedge B$$

- *Conjunction elimination*

$$A \wedge B \vdash A$$

$$A \wedge B \vdash B$$

- **Disjunction**

- *Disjunction introduction*

$$A \vdash A \vee B$$

$$B \vdash A \vee B$$

- *Disjunction elimination*

$$A \vee B, A \implies C, B \implies C \vdash C$$

- **Equivalence**

- *Biconditional introduction*

$$A \implies B, B \implies A \vdash A \iff B$$

- *Biconditional elimination*

$$A \iff B \vdash A \implies B$$

$$A \iff B \vdash B \implies A$$

- **Negation**

- *Double negation*

$$\neg\neg A \vdash A$$

- *Reductio ad absurdum*

$$A \implies B, A \implies \neg B \vdash \neg A$$

At this point a couple of observations can be made.

- Some results can be derived without any premises. These are called tautologies (TODO: remove def). The most famous (and controversial) one is probably the law of excluded middle

$$\vdash p \vee \neg p.$$

Example

A derivation of the law of excluded middle may be given as follows:

1. Start a conditional proof with the hypothesis $\neg(p \vee \neg p)$
 - (a) Start a conditional proof with the hypothesis p
 - i. Disjunction introduction with the hypothesis (a) yields $p \vee \neg p$
 - (b) The conclusion of the subproof yields $p \implies (p \vee \neg p)$
 - (c) Start a conditional proof with the hypothesis $\neg p$
 - i. The wff 1. may be reiterated in the subproof: $\neg(p \vee \neg p)$
 - (d) The conclusion of the subproof yields $\neg p \implies \neg(p \vee \neg p)$
 - (e) Reductio ad absurdum of (b) and (d) yields $\neg p$
2. The conclusion of the subproof yields $\neg(p \vee \neg p) \implies \neg p$
3. Start a conditional proof with the hypothesis $\neg(p \vee \neg p)$
 - (a) Start a conditional proof with the hypothesis $\neg p$
 - i. Disjunction introduction with the hypothesis (a) yields $p \vee \neg p$
 - (b) The conclusion of the subproof yields $\neg p \implies (p \vee \neg p)$
 - (c) Start a conditional proof with the hypothesis p
 - i. The wff 1. may be reiterated in the subproof: $\neg(p \vee \neg p)$
 - (d) The conclusion of the subproof yields $p \implies \neg(p \vee \neg p)$
 - (e) Reductio ad absurdum of (b) and (d) yields $\neg(\neg p)$
 - (f) Double negation in (e) yields p
4. The conclusion of the subproof yields $\neg(p \vee \neg p) \implies p$
5. Reductio ad absurdum of 2. and 4. yields $\neg\neg(p \vee \neg p)$
6. Double negation in 5. yields $p \vee \neg p$

This concludes the proof.

- Obviously proofs using just these inference rules tend to be very long, repetitive and not very illuminating. The proof above can be shortened quite considerably with some of the inference rules we will see later.
- In these inference rules the implication \implies has been given a special place. It is the only connective that appears outside the rules that define it. From this we may wonder if it is possible to define the other connectives in terms of implication. This will be discussed in more detail in the next section.
- All valid deductions may be rephrased as implications. If we have a valid deduction

$$A, B, \dots \vdash C$$

then we may use this deduction to deduce

$$\vdash A \wedge B \wedge \dots \implies C.$$

1. Start a conditional proof with the hypothesis $A \wedge B \wedge \dots$
 - (a) Conjunction elimination of 1. yields A, B, \dots
 - (b) Use the valid deduction $A, B, \dots \vdash C$ to deduce C
 2. The conclusion of the subproof yields $A \wedge B \wedge \dots \implies C$

This concludes the proof.

Conversely, if we have the premises A, B, \dots , the tautology $A \wedge B \wedge \dots \implies C$ allows the deduction of C .

1. Start with the premises A, B, \dots
 2. Conjunction introduction of 1. yields $A \wedge B \wedge \dots$
 3. Because it is a tautology, the wff $A \wedge B \wedge \dots \implies C$ can be deduced here.
 4. Modus ponens of 2. and 3. yields C

This concludes the proof.

We have now shown two statements that look like implications:

- **If** we have a deduction $A, B, \dots \vdash C$, **then** we may deduce $\vdash A \wedge B \wedge \dots \implies C$.
- **If** we have a tautology $\vdash A \wedge B \wedge \dots \implies C$, **then** we may deduce $A, B, \dots \vdash C$.

We may now use the inference rule *biconditional introduction* to infer their equivalence. TODO: deduction theorem.

- The previous point is proves a result from the meta-theory: the theory about the theory. It also shows a first practical application of the theory. TODO integrate with formal system theory.

2.2.2 More inference rules

For ease of use we split this section up into two parts:

- Rules of inference proper and
- Rules of replacement: these are symbolic expressions that are equivalent and may always be substituted one for the other.

All of these rules can be deduced from the previous rules.

2.2.2.1 Rules of inference

1. *Modus tollens*

$$A \implies B, \neg B \vdash \neg A$$

2. *Modus ponendo tollens*

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

3. *Disjunctive syllogism*

$$A \vee B, \neg A \vdash B$$

4. *Hypothetical syllogism or transitivity of the implication*

$$A \implies B, B \implies C \vdash A \implies C$$

5. *Dilemma*

$$A \vee B, A \implies C \vdash C \vee B$$

6. *Absorption*

$$A \implies B \vdash A \implies A \wedge B$$

7. *Negation of equivalence*

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

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

8. *Reductio ad absurdum of a hypothesis.* If, in a conditional proof that that start with the hypothesis A , both the wff B and $\neg B$ appear as valid, one may conclude $\neg A$ in the main proof. This concludes the subproof.

2.2.2.2 Rules of replacement

1. **Double negation.** This rules is repeated here because it can in general be thought of as a rules of replacement.

$$\neg\neg A \iff A$$

2. **Commutativity**

(a) *Of equivalence*

$$(A \iff B) \iff (B \iff A)$$

(b) *Of conjunction*

$$(A \wedge B) \iff (B \wedge A)$$

(c) *Of disjunction*

$$(A \vee B) \iff (B \vee A)$$

3. Associativity

(a) *Of conjunction*

$$((A \wedge B) \wedge C) \iff (A \wedge (B \wedge C))$$

(b) *Of disjunction*

$$((A \vee B) \vee C) \iff (A \vee (B \vee C))$$

4. Distributivity

$$A \wedge (B \vee C) \iff (A \wedge B) \vee (A \wedge C)$$

$$A \vee (B \wedge C) \iff (A \vee B) \wedge (A \vee C)$$

5. Negation

(a) *Of conjunction* also known as one De Morgan's laws

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

(b) *Of disjunction* also known as one De Morgan's laws

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

(c) *Of implication*

$$\neg(A \implies B) \iff A \wedge \neg B$$

6. Transposition

$$(A \implies B) \iff (\neg B \implies \neg A)$$

7. Material implication

$$(A \implies B) \iff (\neg A \vee B)$$

8. Tautology

$$A \iff (A \vee A)$$

$$A \iff (A \wedge A)$$

2.3 Other formulations of predicate logic

In this section three other, equivalent formulations of predicate logic are given. Many interesting technical details could be given, but this can easily get quite complicated and a text on physics is not really the place. (Read: I haven't a clue what I'm talking about). This section can easily be skipped.

2.3.1 With axiom schemata

This formulation uses axiom schemata instead of inference rules. These give families of tautologies. Specific tautologies can be obtained by filling in the upper case letters with specific wffs. Only one inference rule is kept: **modus ponens**.

Deductions are made by writing down tautologies and applying modus ponens.

The axiom schemata (AS) which are posited to be tautologies are:

$$\begin{array}{ll} AS1 & A \implies (B \implies A) \\ AS2 & (A \implies (B \implies C)) \implies ((A \implies B) \implies (A \implies C)) \\ AS3 & ((A \implies B) \implies A) \implies A \\ AS4 & (A \wedge B) \implies A \\ AS5 & (A \wedge B) \implies B \\ AS6 & A \implies (B \implies (A \wedge B)) \\ AS7 & A \implies (A \vee B) \\ AS8 & B \implies (A \vee B) \\ AS9 & (A \implies C) \implies ((B \implies C) \implies ((A \vee B) \implies C)) \\ AS10 & (A \iff B) \implies (A \implies B) \\ AS11 & (A \iff B) \implies (B \implies A) \\ AS12 & (A \implies B) \implies ((B \implies A) \implies (A \iff B)) \\ AS13 & \neg\neg A \implies A \\ AS14 & (A \implies B) \implies ((A \implies \neg B) \implies \neg A) \end{array}$$

2.3.2 With definitions

In our first formulation some of the inference rules acted like definitions. In the list axiom schemata clearly some of them fulfill the same rôle. Thus the list can be drastically shortened by explicitly introducing definitions, which we denote with the symbol \equiv .

This formulation of predicate logic consists of

- Definitions:

$$\begin{aligned} A \vee B &\equiv \neg A \implies B \\ A \wedge B &\equiv \neg(\neg A \vee \neg B) \\ A \iff B &\equiv (A \implies B) \wedge (B \implies A) \end{aligned}$$

- Axiom schemata: AS1 - AS3, AS13 and AS14
- Inference rule: modus ponens

2.3.2.1 Some relevant technicalities for definitions

This last formulation of predicate logic used explicit definitions. It was, however, equivalent to previous formulations. In a way this means the connectives were also “defined” in the previous formulations, albeit implicitly. An implicit definition fixes the meaning of a connective (or symbol or object in general) by saying which inferences containing said connective are valid.

Giving explicit definitions of connectives. It may be striking in this formulation of predicate logic with definitions that only three connectives are given explicit definitions. Clearly giving all connectives explicit definitions would be impossible as an explicit definition is defined in terms of something else and then there would be nothing else in terms of which to define

things. The choice of which connectives to give explicit and which to give implicit definitions is to some degree arbitrary. Here everything was defined in terms of \neg and \implies , but we could just as easily have defined \implies as

$$A \implies B \equiv \neg A \vee B$$

in which case everything would have been defined in terms of \neg and \vee .

If it is possible to give an explicit definition for a connective, then every formula containing that connective can be written using its definition. It is, in effect, redundant. Such a “redundant” symbol is properly called eliminable. Thus it is only possible to give an explicit definition of a symbol if it is eliminable. Additionally it may not be possible to give an explicit definition if too many other explicit definitions have already been given. For example, \wedge , \vee , \implies and \iff are all eliminable, but not all at the same time.

Not all symbols are eliminable. For example \neg is not.¹ So far we have managed to eliminate all connectives but two, one unitary and one binary. It is however possible to formulate predicate logic with only one connective, if we introduce new connectives.

Giving explicit definitions of new connectives. So far we have not actually given a proper, formal definition of “explicit definitions”, only some examples. It turns out that giving a good definition is not easy. Not every sentence containing a “ \equiv ” is a good definition. Take for example the definition

$$\vdash A \equiv A \implies B$$

for any wff B . This definition cannot be added to the system of propositional logic we have developed so far without substantially altering it.

Example

Naively extending propositional logic to include the connective \vdash , means that any proposition can imply anything. That is, for any A and B , $A \implies B$ is tautologically true. The derivation of this fact is as follows:

1. Start with AS1: $A \implies (C \implies A)$.
2. Using the definition, 1. yields $\vdash A$.
3. Substituting in the definition now yields $A \implies B$.

Obviously this definition creates many new tautologies. For this reason it is called a creative definition (or a non-conservative definition). In fact this system is especially bad: all wffs are now tautological, rendering the system completely trivial.

So now we want a way to construct an explicit definition such that

1. the symbol being defined is eliminable in the new system that contains it;
2. the definition is not creative.

¹This can be easily seen by attempting to define $\neg A$ using the other connectives. Such a definition may only contain the other connectives and A . There are a limited number of ways we can combine these building blocks. The expressions $A \wedge A$ and $A \vee A$ are both equivalent to A , so using those is quite useless. The expressions $A \implies A$ and $A \iff A$ are both tautologies. Combining a tautology, a connective and A can only ever yield A or another tautology, as the reader can readily verify. Repeating this process will not yield anything new, especially not $\neg A$. Thus \neg is not eliminable.

An explicit definition with the following structure has these properties:

$$\text{definiendum} \quad \equiv \quad \text{definiens}$$

where

- the definiendum is an expression containing only the new symbol to be defined once, variables, and potentially brackets;
- the definiens contains no variables that are not also present in the definiendum.

The definition of $\vdash A$ fails on the second point. This is a reasonable definition of an explicit definition. However it is actually too strict. There are some perfectly good explicit definitions that do not fit this mold (TODO which?). Unfortunately finding the perfect definition is still an open problem.

Some new connectives. We can now give some examples of good explicit definitions:

- The Sheffer stroke, written as \uparrow or $|$, also known as NAND in computer science and engineering, can be defined as

$$A \uparrow B \quad \equiv \quad \neg(A \wedge B).$$

- The Pierce arrow \downarrow , also known as the Quine dagger \dagger , also known as NOR in computer science and engineering, can be defined as

$$A \downarrow B \quad \equiv \quad \neg(A \vee B).$$

- The exclusive disjunction \vee , also known as the “exclusive or” or XOR, can be defined as

$$A \vee B \quad \equiv \quad (A \vee B) \wedge \neg(A \wedge B).$$

Once either that the Sheffer stroke or Pierce arrow have been introduced, all other connectives can be simultaneously eliminated, including \neg .

2.3.3 With axioms

The concept of an axiom schema can be replaced with a new inference rules: *uniform substitution* (TODO slash notation). This rule states that in any tautology we may replace a letter denoting a primitive proposition with an arbitrary wff.

The final formulation of predicate logic consists of the following:

- Axioms:

$$\begin{aligned} A1 \quad & p \implies (q \implies p) \\ A2 \quad & (p \implies (q \implies r)) \implies ((p \implies q) \implies (p \implies r)) \\ A3 \quad & ((p \implies q) \implies p) \implies p \\ A4 \quad & \neg\neg p \implies p \\ A5 \quad & (p \implies q) \implies ((p \implies \neg q) \implies \neg p) \end{aligned}$$

- Definitions:

$$\begin{aligned} A \vee B & \equiv \neg A \implies B \\ A \wedge B & \equiv \neg(\neg A \vee \neg B) \\ A \iff B & \equiv (A \implies B) \wedge (B \implies A) \end{aligned}$$

- Inference rule: modus ponens and uniform substitution.

2.4 Semantics

2.4.1 Model-theoretic semantics

2.4.2 Proof-theoretic semantics

standard and nonstandard models

TODO valuation function

2.4.3 A note on constructivism

2.4.4 Truth tables

2.4.5 Semantic tableaux

2.4.6 A couple of paradoxes

Chapter 3

Critiques and extensions of propositional logic

3.1 Some problems with propositional logic

3.1.1 Interpreting the implication

3.1.2 Relevant derivation

3.1.3 Ex falso quodlibet

3.2 Multi-valued logic

3.3 Intuitionistic logic

3.4 Modal logic

3.5 Relevant logic

3.6 Paraconsistent logic

3.7 Non-monotonic logic

Chapter 4

First-order predicate logic

problem of multiple generality

4.1 Syntactic extensions

terms and formulae

polish notation

quantifiers

bound variables, scope and heuristics

4.2 Higher-order logic

Chapter 5

What constitutes a proof?

5.1 Proofs as convincing texts

not symbols. convince

5.1.1 Use of the word “arbitrary”

5.2 Types of proof

existence and uniqueness

two inclusions

Part V

Category theory

Chapter 1

Introduction

Many properties of mathematical systems can be unified and simplified by a presentation with diagrams and arrows.

Chapter 2

Categories, functors and natural transformations

2.1 Categories

2.1.1 Graphs

A (directed) graph consists of

- of objects a, b, c, \dots
- of arrows f, g, h, \dots
- two operations
 1. Domain, which assigns to each arrow f an object $a = \text{dom } f$
 2. Codomain, which assigns to each arrow f an object $b = \text{codom } f$

If f is an arrow, $\text{dom}(f) = a$ and $\text{codom}(f) = b$, then we write

$$f : a \longrightarrow b \quad \text{or} \quad a \xrightarrow{f} b$$

2.1.2 Metacategories

A metacategory is a graph with two additional operations:

1. Identity, which assigns to each object a an arrow $1_a = \text{id}_a : a \rightarrow a$.

$$\text{Identity} : \{\text{objects}\} \rightarrow \{\text{arrows}\} : a \mapsto \left(a \xrightarrow{1_a} a \right)$$

2. Composition, which assigns to each pair of arrows (f, g) with $\text{dom } g = \text{codom } f$ an arrow $g \circ f$, called their composite, with

$$\text{dom } f \xrightarrow{g \circ f} \text{codom } g$$

These operations are subject to the following axioms:

1. Associativity. For any given arrows in the configuration

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$$

one has the equality

$$h \circ (g \circ f) = (h \circ g) \circ f$$

2. Unit law. For all arrows $a \xrightarrow{f} b$ and $b \xrightarrow{g} c$, composition with the identity arrow 1_b gives

$$t_b \circ f = f \quad \text{and} \quad g \circ 1_b = g$$

This axiom asserts that the identity arrow acts as an identity for composition.

Composition can be summarised diagrammatically by saying that it transforms $a \xrightarrow{f} b \xrightarrow{g} c$ to

$$\begin{array}{ccc} & b & \\ f \nearrow & & \searrow g \\ a & \xrightarrow{g \circ f} & c \end{array}$$

Many properties can be represented by using such diagrams. A diagram is called commutative if all possible compositions with the same domain and codomain are equal¹ to each other.

Example

1. The associativity axiom can be restated as the commutativity of the diagram

$$\begin{array}{ccc} a & \xrightarrow{h \circ (g \circ f) = (h \circ g) \circ f} & d \\ & \searrow h \circ g & \nearrow g \circ f \\ b & \xrightarrow{g} & c \end{array}$$

2. The unit law can be restated as the commutativity of the diagram

$$\begin{array}{ccc} a & \xrightarrow{f} & b \\ & \searrow f & \downarrow 1_b \\ & & b \end{array}$$

A metacategory is any interpretation that satisfies the axioms.

¹Depending on the definition of equality at hand this statement may be trivial. For example, arrows are defined as equal iff they have the same domain and codomain, then all compositions with the same domain and codomain are equal by definition. In other words, it is only possible to write commutative diagrams. If arrows are conceived to contain more structure than just domain and codomain, more interesting definitions of equality are possible and the assertion of the commutativity of a diagram becomes a non-trivial assertion.

Example

In the metacategory of sets all objects are sets and all arrows are functions. For any set S , the identity operation defined above yields the *identity function*

$$1_S : S \rightarrow S : s \mapsto s$$

The composite of two functions $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ is defined as

$$g \circ f : X \rightarrow Z : x \mapsto g(f(x))$$

for all $x \in X$.

The arrows of a category are often called its morphisms. For any object the identity arrow is uniquely defined. Therefore it is possible to identify the identity arrow with the object itself. This further means it is possible to dispense with objects altogether and deal only with arrows. Arrows-only metacategories can be defined characterised axiomatically as follows

An arrows-only metacategory C consists of

- arrows;
- ordered pairs of arrows, called composable pairs; and
- an operation assigning to each composable pair (f, g) an arrow $g \circ f$, called their composite.

We say “ $g \circ f$ is defined” to mean “ (g, f) is a composable pair”.

Given these data, an identity in C is *defined* to be an arrow u such that $f \circ u = f$ whenever $f \circ u$ is defined and $u \circ g = g$ whenever $u \circ g$ is defined. Then the data are also required to satisfy some axioms:

1. The composite $(h \circ g) \circ f$ is defined if and only if the composite $h \circ (g \circ f)$ is defined. If they are defined, they are equal (this triple composite may be written hgf).
2. The triple composite hgf is defined whenever both composites hg and gf are defined.
3. For each arrow g of C there exist identity arrows u and u' of C such that $u' \circ g$ and $g \circ u$ are defined.

The identity arrows u and u' in the third axiom may be interpreted as domain and codomain; they are necessarily unique (TODO why?).

The axioms of metacategories and arrows-only metacategories are equivalent.

2.1.2.1 Categories

A category is any metacategory where the objects and arrows form sets. Not all metacategories are categories. For example the metacategory of sets is not a category. This is because there is

no such thing as the set of all sets.²

Thus a category C contains a set of objects O and arrows A . We often drop the letters O and A and write

$$a \in C \quad \text{and} \quad f \text{ in } C$$

to mean that a is an object of C and f is an arrow in C . We also write

$$\text{hom}(b, c) = \{f \mid f \text{ in } C, \text{dom } f = b, \text{codom } f = c\}$$

for the set of arrows from b to c .

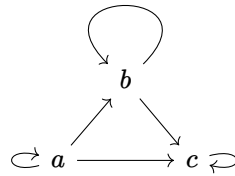
Example

Here are some elementary categories. In each case there is only one possible definition of composition.

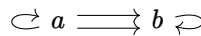
- **0** is the empty category.
- **1** is the category



- **2** is the category
- **3** is the category



- \Downarrow is the category



²See Russell's paradox.

Appendix A

Symbols

Appendix B

Formula reference

Appendix C

Bibliography