A Logic of Exceptions

Using The Economics Pack *Applications of Mathematica* for Elementary Logic

Thomas Colignatus, March 2011

http://thomascool.eu

Applications of *Mathematica*

Thomas Colignatus is the name of Thomas Cool in science.

Published by Cool, T. (Consultancy & Econometrics)

Thomas Cool *Consultancy & Econometrics*

Rotterdamsestraat 69, NL-2586 GH Scheveningen, The Netherlands

http://thomascool.eu

NUR 120, 738; JEL A00; MSC2010 03B01, 97E30

ISBN 978-90-804774-7-6

Mathematica is a registered trademark of Wolfram Research, Inc.

This book uses version 8.0.1

Aims of this book when you are new to the subject

The following should have been achieved when you finish this book.

- You will better understand the major topics in logic and inference.
- You can check whether common inferences are valid or not, and tell why.
- You can better determine a rejoinder that others would sooner accept.
- You can read this book *as it is,* thus also without *Mathematica*. Without ever running a program, you will still benefit from the discussion. However, if you have a computer and practice with the programs, then you end up being able to run the logic and inference routines and interprete their results.

Note that the software can be downloaded freely from the internet and be inspected; however, if you want to run it, then you need a licence.

Aims of this book when you are an advanced reader

The following should have been achieved when you finish this book.

- One of the aims of this book has been to present logic so that new readers get a clear view on the subject. When you read this book in this way as well, then you will benefit from those aims (see above) and be able to discuss logic in this fashion as well.
- You will be an advanced student but you might lack in balance between either the math or the history and philosophy. Having digested this book you will find a better place for both, enhancing the part that you lack.
- You will better understand the main choice between (1) the theory of types, (2) proof theory, (3) three-valued logic.
- You will refocus your research towards issues that matter more.

Note that this book has two agenda's: First to develop logic and inference from the bottom up and educate students in a *logic of exceptions*. Secondly, to show how the confusions generated by Russell, Tarski, Hilbert and Gödel can be solved. The first objective is more permanent while the second objective is more transient. Once researchers have adopted the solutions in this book, there probably remains little value in teaching new students about old confusions. In ten years time it may just be history, worth of only a paragraph. Yet for the moment it still is an objective:

- You can explain to others that Gödel's verbal explanations of his theorems do not match the mathematics, and, that proper development of the assumptions shows those verbal statements to be incorrect. You will probably reconsider your view on those theorems and start to see that they are rather irrelevant for logic and inference.

Abstract

- The book gives an *Introduction to Elementary Logic*, from the novice level up to Gödel's incompleteness theorems. The first chapter has a direct hands-on approach so that the novice can directly learn about the connectives of propositional logic and grasp the power of inference. This, and the sense of achievement, should stimulate the student to continue, while it also provides a basis to reflect on what already has been learned. The later chapters then delve into theory, with a closing chapter with notes on formalization.
- Discussed are (1) the basic elements: propositional operators, predicates and sets; (2) the basic inference notions: inference, syllogism, axiomatics, proof theory; (3) the basic extra's: historical notions, relation to the scientific method, the paradoxes. The new elements in the book are: (4) a *logic of exceptions*, solutions for paradoxes, analysis of common mistakes, routines in *Mathematica*.
- Confronted with the Liar paradox ("This sentence is false"), modern logic recognizes three main approaches: (1) the theory of types, with levels in language, (2) the abandonment of truth and adoption of a theory of proof with undecidability instead of inconsistency, and (3) three-valued logic. The first two approaches sacrifice useful forms of selfreference. Also, the Gödel incompleteness theorems are based upon a cousin of the Liar ("This sentence is unprovable"), which makes the result rather unconvincing, while the philosophical implications on undecidability seem too large in relation to the dubious nature of such a paradoxical statement. Hence three-valued logic is a more fruitful approach. The logic community has hesitations about three-valued logic. The programs implemented in *Mathematica* show that it works quite nicely.
- Gödel's incompleteness theorems are rather useless for understanding logic and the foundations of mathematics. Moreover, Gödel's verbal and "meta-mathematical" explanations and interpretations of the theorems appear not to match its mathematics. His deduction is based on allowing statements to refer to themselves selfreferentially but disallowing this for the whole system itself. The proper approach is to allow for the latter and to adopt the axiom $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$ or that when S proves something then it considers it also proven that it proves something. In this case the Gödeliar collapses to the Liar, bringing us back to the situation 500 years B.C.
- The key development of this course is the *logic of exceptions.* An intelligent handler of information always keeps in mind that there may be exceptions to the rules applied. This approach does not only allow for freedom of expression but also makes for sound reasoning. And it gives the right results.

- The *Mathematica* programs support the development in the book. The programs become available within *Mathematica* and within *The Economics Pack* - Cool (1999, 2001) available at http://thomascool.eu - and then evaluating:

```
Needs["Economics`Pack`"]

ResetAll

Economics[Logic]
```

Note: You can read this book *as it is,* thus also without *Mathematica*. Without ever running a program, you will still benefit from the discussion. See the internet for other logic programs.

Note: On the palette for *The Economics Pack* there is a button for the User Guide. Click there and you will find the entire text of this book available there.

Keywords

Logic, inference, induction, axiomatics, proof theory, propositional logic, predicate logic, set theory, foundations of mathematics, incompleteness theorems, liar paradox, antinomies, intuitionism, history of logic, morality, modality, certainty, epistemology, methodology of science, general philosophy, general economics

Preface

In 1980 I was a graduate student in econometrics and became interested in logic and the methodology of science. This seems like a departure from economics, but there is some logic to it, as should become clear below. The study got more involved than anticipated and resulted in a 500 page draft Introduction textbook in Dutch, and in 1981 a 100 page summary discussion in English called "In memoriam Philetas of Cos et alii" on the Liar paradox, Russell's set paradox, Brouwer's intuitionism, and the "Gödeliar", The major point in 1980-1981 was, and will be below, the development a *logic of exceptions*, meaning that people are free to define concepts and derive conclusions as they wish, up to the moment that they run into a contradiction so that they have to make some revision. This is how science works in practice and this is what logic and inference should allow for. The definition of truth would include the proviso that it holds in general *except* for such expressions as the Liar or Gödeliar. Logicians make a great show about their paradoxes but once you adopt such a *logic of exceptions* then you can achieve a great economy in discarding most of what they say.

In 1981 there were some factors that caused me to drop the subject again. The logicians that I approached on my findings did not react logically. Graduation became more important as well. What started to worry me more in those days were the newspaper reports on "The Global 2000 Report to the President" (Barney (1981)), reminding us that the world population would rise from 4 billion in 1975 to 6.4 billion in 2000. Studying logic had also caused me to come across Keynes's obituary of Frank Ramsey, and I could only agree with it:

> "If he had followed the easier path of mere inclination, I am not sure that he would not have exchanged the tormenting exercises of the foundations of thought, where the mind tries to catch its own tail, for the delightful paths of our own most agreeable branch of the moral sciences, in which theory and fact, intuitive imagination and practical judgement, are blended in a manner comfortable to the human intellect." J.M. Keynes, obituary notice of F.P. Ramsey, The Economic Journal 40, March 1930; also quoted in Hao Wang (1974:25).

In 1981 my analysis was that the world's economic problems were primarily caused in the West. This may seem paradoxical since most misery is on the other continents, yet, it makes economic sense. Policy in the West was and is to protect its own employment by barriers to trade. Solving unemployment in the West would create scope to liberate trade and enhance the possibilities for the developing nations. I also wrote an analysis that development aid is useless when there is no good government at the receiving end. Hence after graduation in 1982 my research agenda became focussed on unemployment in the West, and the draft books on logic were put aside. By 1989 I had found my explanation for unemployment. Publication was delayed but succeeded in Colignatus (1992b). Eventually some propositional logic turned up in Cool (1999, 2001), *The*

Economics Pack. Deontic logic turned up in Colignatus (1992b) and then in (2001, 2007), "Voting theory for democracy". Colignatus (2000, 2005) were better didactic statements of the explanation of unemployment in the West. Colignatus (2006) is an application to a Caribbean economy. Just now, in December 2006 and January 2007 I find some time and the opportunity to reconsider those shelved books on logic. August 2007 added some Reading Notes (Chapter 11).

My hope is that these pages will be beneficial for those who study logic and the methodology of science. These subjects tickle the mind and the number of books on them is enormous, yet I still think that some readers might benefit from the concept of a *logic of exceptions.* And, not just the idea, that can be phrased in a single sentence, but also developed as a course in logic from the bottom up.

After these 25 years and with hindsight there appears to be a sound economic reason as well, which I hadn't seen in 1980-1981, and which makes the digression into logic even more acceptable to economics. Since the Egyptians, mankind has been trying to solve the problem of bureaucracy. One frequent approach is the rule of law, say, that a supreme law-giver defines a rule that a bureaucracy must enforce. It is difficult for a law however to account for all kinds of exceptions that might be considered in its implementation. Ruthless enforcement might well destroy the very intentions of that law. Some bureaucrats might still opt for such enforcement merely to play it safe that nobody can say that they don't do their job. Decades may pass before such detrimental application is noticed and revised. There is the story of Catherine the Great regularly visiting a small park for a rest in the open air, so that they put a guard there; and some hundred years after her death somebody noticed that guarding that small park had become kind of silly. When both law-givers and bureaucrats grow more aware of some *logic of exceptions* then they might better deal with the contingencies of public management. It is a long shot to think so, of course, but in general it would help when people are not only aware of the rigour of a logical argument or rule but also of the possibility of some exception.

Our intended audience is a general public of students. We assume that you know some arithmetic and some reasoning, and that you are willing to develop your knowledge alongside with working on this book.

As said, Cool (1999, 2001) *The Economics Pack* already contains programs in *Mathematica* for propositional logic. These have been updated and extended. These programs can be used at an undergraduate level. As has been observed before, *Mathematica* itself is already a decision machine for the predicate calculus. This book uses these programs to develop logic from the bottom up. The focus is on logic, not on the programs. But if you have these programs available, then you can have a hands-on experience, verify the

conclusions, try your own cases, and, write your higher-level programs.

It must be acknowledged that logic can become very abstract, so that some of the more fundamental conclusions of this book will require attention again at a more advanced level. If you are such an advanced student then you are free to start with the later chapters. However, given the apparently widespread misunderstandings, you are well advised to work through the book sequentially. It will not hurt to act as if you are new to the subject. In fact, this book follows the strategy to first create some competence in logic and only then discuss the theory, so that both ways unite in a clear departure from current misconceptions. You will miss out on that development if you would jump too many chapters.

I thank again the logicians who helped me in 1980-1981. So much time has passed now that it doesn't seem proper to mention their names, as this might cause needless confusion. But in all cases I must thank Howard DeLong for his wonderful 1971 book that introduced me to the subject of logic. I can still advise it to everyone (except that you must keep a *logic of exceptions* in mind while reading it).

Addendum March 2011: (1) The advent of *Mathematica* 8.0.1 causes an update of the software. In the book some minor typing errors have been corrected. In *Mathematica* 7 Equivalent was introduced and hence this book and software will frequently use $Equivalent to maintain the printing order and consistency with the development of three-valued logic. (2) I thank Richard Gill (mathematical statistics, Leiden and Royal Academy of Sciences) for his kind review of the first edition, Gill (2008). (3) Also the book *Voting Theory of Democracy* (3rd edition 2011) has been updated. (4) New are *Elegance with Substance* (2009) on mathematics education, and *Conquest of the Plane* (2011) that extends on the "paradoxes by division by zero" already mentioned below. (5) Reflecting on "Logicomix" (2009): it is a wonderful book but it would benefit from a note that Russell's Paradox is resolved (see p128-129 below) so that its readers see better that the story is only a (partly fictitious) historical episode, see Colignatus (2010).

Brief contents

Table of contents

Part I. Overall introduction

0.1 Conditions for using this book

The basic requirement for using this book is that you have at least a decent highschool level of mathematics or are willing to work up to that level along the way. We assume that this book could be used in the first year of a college or university education.

You can read this book *as it is,* thus also if you do not have *Mathematica*. Even without ever running a program, you will still benefit from the discussion.

Readers new to the specific formats of *Mathematica* are advised to check the appropriate subsections on those, since those notations will be used.

Yet, if you have *Mathematica* and want to run the programs, then this book assumes that you have worked with *Mathematica* for a few days. You must be able to run *Mathematica*, understand its handling of input and output, and its other basic rules. Note that *Mathematica* closely follows standard mathematical notation. There are some differences with common notation though since the computer requires very strict instructions. Note also that *Mathematica* comes with an excellent Help function that starts from the basic "Getting Started" and works up to the most advanced levels. There are also many books that give an introduction.

When you want to run the `Logic`` and `Inference`` programs, you should also have a working copy of *The Economics Pack*, applications for *Mathematica,* by the same author with the software downloadable from the internet.

0.2 Structure of the book

This book is for both beginners and advanced readers. They are guided by the respective sections below.

The book basically has:

1. The basic elements: propositional operators, predicates and sets.
2. The basic inference notions: inference, syllogism, axiomatics, proof theory.

3. The basic extra's: historical notions, relation to the scientific method, the paradoxes.
4. The news: A *logic of exceptions*, solutions for paradoxes, debunking of nonsense, routines in *Mathematica*.

This book has more motivations: (a) the intellectual jolt that a student gets from the introduction to symbolic logic, (b) the great value of plain logic for economic and statistical inference, or any science, (c) the paradoxes and the fundamental issues that they cause, (d) the indeed exciting history of philosophy, mathematics and logic, (e) the presentation of the concept of a *logic of exceptions*. The latter presentation required the environment of the first issues as well.

In my professional life as an econometrician I have observed too often that people just don't keep account of possible exceptions. Perhaps it helps future generations when students of elementary logic get it soft-wired that a sound mind reckons with exceptions. The age-old Liar paradox might actually be of use.

Chapters 7 and up justify what the book sells to the novice readers. For the advanced reader, there are some notes on formalization in Chapter 10 and some Reading Notes in Chapter 11. Readers new to the subject who want to understand this part, should use the library, but are advised to have this book available to guide them through the arguments. You are advised to use the internet as well. Logic is an interesting subject and various people and organisations have devoted attention to it.

0.3 A guide

Since *Mathematica* is such an easy language to program in, it also represents something like a pitfall. It is rather easy to prototype the solution to a problem, or to write a notebook on a subject. But it still appears to be hard work to maintain conciseness, to enhance user friendliness and to document the whole.

Keep in mind the distinction between **(a)** issues in logic, **(b)** how a solution routine has been programmed, **(c)** the way how to use the routines.

This book focusses on (a). It however also provides a guide on (c) but neglects (b). Thus, the proper focus is on the *why*, i.e. the content of issues in logic, for which we want to apply these routines. But this also requires that we explain *how* to use them. If you want to know more about how the routines have been programmed, then you might use the routine `ShowPrivate[]`.

0.4 Getting started

When you want to run the programs then you must do the following.

The Economics Pack becomes fully available by the single command `<<Economics`All``. It is good practice however to use a few separate command lines to better control the working environment. Three lines can be advised in particular.

0.4.1 The first line

You start by evaluating:

```
Needs["Economics`Pack`"]
```

This makes the `Economics[]` command available by which you can call specific packages and display their contents. Before you use this, read the following paragraphs first.

0.4.2 The second line

`CleanSlate`` is a package provided with *Mathematica* that allows you to reset the system. You thus can delete some or all of the packages that you have loaded and remove other declarations that you have made. The only condition is that `CleanSlate`` resets to the situation that it encounters when it is first loaded. You would normally load `CleanSlate`` after you have loaded some key packages that you would not want to delete. The ResetAll command is an easy way to call `CleanSlate``. Your advised second line is:

```
ResetAll
```

`ResetAll`	ResetAll calls CleanSlate, or if necessary loads it. This means that your notebook does not have to distinguish between calling CleanSlate` and evaluating CleanSlate[]

Note that if you first load CleanSlate` and then the Economics Pack, then the ResetAll will clear the Pack from your working environment, and thus also remove ResetAll. If you would happen to call ResetAll again after that, then the symbol will be regarded as a Global` symbol.

0.4.3 The third line

After the above, you could evaluate EconomicsPack to find the list of packages.

```
EconomicsPack
```

Select the package of your interest, load it, and investigate what it can do. For example:

```
Economics[Logic]
```

You can suppress printing by an option `Print → False`. You can call more than one package in one call. If you want to work on another package and you want to clear the memory of earlier packages, simply call ResetAll first. This also resets the `In[]` and `Out[]` labels.

`Economics[xi, ...]`	shows the contents of xi` and if needed loads the package (s). Input xi can be Symbol or String with or without back– apostrophe. To prevent name conflicts, Symbols are first removed. Economics[ ] doesn' t need the Cool`, Varianed` etc. prefixes
`Economics[All]`	assigns the Stub– attribute to all routines in the Pack (except some packages)
`EconomicsPack`	gives the list {directory → packages}

Note: Economics[x, Out → True] puts out the full name of the context loaded.

This book will use basically these packages:

Economics[Logic, Logic`Deontic, Logic`SwitchGraphics, Logic`SetGraphics, Logic`AIOE, Probability]

Also the Inference` package is used but only load that when you use it in the appropriate chapter, since it uses the InferenceMachine which slows down loading.

0.4.4 Using the palettes

The Pack comes with some palettes. These palettes have names and structures that correspond to the chapters in *The Economics Pack* itself.

- The master palette is "TheEconomicsPack.nb" and it provides the commands above and allows you to quickly call the other palettes or to go to the guide under the help function.
- The other palettes have "TEP_" as part of their name, so that they can easily be recognised as belonging to the Pack. These "TEP_" palettes contain blue buttons for loading the relevant packages and grey buttons for pasting commands.
- The exception here is "TEP_Arrowise.nb" that only deals with the package for making arrow diagrams.

The logic and inference palettes are part of the TEP_Enhancement palette.

0.4.5 All in one line

You can also load the Pack by the following single line.

<<Economics`All`

This evaluates Needs["Economics`Pack`"] and Economics[All], and opens the palettes. It

does not call ResetAll, however.

0.5 For the beginner

The best introduction into logic is to start doing it. Chapter 1 has been written so that you can start right away.

If you think that you are an advanced student in logic, then you should continue with the next section 0.6. Otherwise, if you are beginning, you should continue with Chapter 1 and work up to and including Chapter 6.

PM. You should practice a lot, and use the examples in this book that teach you about the different properties of the various logical schemes. Only afterwards, and only if you are willing to proceed to the advanced level, then you could start reading the next section 0.6, and then continue with Chapter 7 onwards. Keep in mind that you could look in other textbooks on logic in the library too, notably DeLong (1971), since it cannot be fully excluded that some chapters here presume some knowledge that has not been fully stated here. It is good to read those other textbooks, since reading these will clarify to you that those other books *do*n't *give you a hold* on the problem while *this book does.*

Since various *Mathematica* programs are provided, you can have an hands-on experience, and this will allow you to better understand the issues.

OK, this is the point where you switch to Chapter 1.

0.6 For the advanced reader

0.6.1 Aims of this book

This paragraph is only to remind you of the aims set out on the first page of the book.

0.6.2 Summary

The book is a textbook that gives an introduction to elementary logic and defines the notion of a *logic of exceptions*.

It is claimed that the Liar is solved and that the Theorems of Gödel are misleading to the highest degree, i.e. are only half-truths.

First a distinction is made between simple and complex Liars. The simple Liar is solved by two-valued logic, the complex Liar by three-valued logic. It is shown that standard logic induces three-valued logic in a necessary way. The approach of Russell and Tarski is rejected as unnecessary from a logical point of view and too complicated to be useful.

Also, three-valued logic can deal with the so-called strengthened Liars of three-valued logic. Having established this, the three-valued logic is applied to the paradox of Russell, the logic of Brouwer, and the theorems of Gödel.

Secondly, then, three-valued logic appears to be behind set theory. Sensical sets can be defined in two-valued logic but if one allows freedom of definition then nonsensical results need to be catalogued as such.

Thirdly, Heyting's axiomatization of Brouwer's system of inference appears to be inadequate.

Fourthly. The theorems of Gödel appear to be proven by the use of the Liar of proof theory, here called the Gödeliar, "This statement is not provable". When this Gödeliar is essentially as nonsensical as any other Liar, these theorems appear to be build on nonsense. Distinguished are weak and strong systems and it is only by weakness that Gödel's system does not show the contradiction that is behind the proof. When we use adequately strong systems then the nonsense can be dealt with by three-valued logic. In particular, while Gödel limits selfreference to only statements, (1) we can also allow that statements refer to the system as a whole and (2) we can accept an axiom $((S \vdash p) \Rightarrow (S \vdash (S \vdash p)))$. The Gödeliar collapses to the Liar and the theorems disappear. Hence Gödel's "metamathematical" argument that his theorems would hold for any system is invalid, they just hold for his too weak assumptions.

Fifthly, the book contains historical and philosophical context relevant for the arguments.

Sixthly, all steps in the argument are backed up by formal notes in Chapter 10. Evaluations in *Mathematica* support the argument but of course have limited formal status.

Seventhly, the book advises to the freedom of thought and tolerance for the opinions of others. Dogmatists are the exception that prove that rule.

0.6.3 The paradoxes

The state of the art on the paradoxes and Gödel's theorems is rather disappointing. There is a great diversity of opinion what the proper solution approaches are. And, where there seems to be agreement, exactly there rather dubious approaches are embraced. The key example to illustrate this are the theorems of Gödel (1931). These theorems are proven with the use of a weak variant of the Liar paradox and they would not hold since this variant is as senseless as the Liar itself, i.e. without a True | False value relevant for empirical application. It is only because of weak assumptions, merely a twist, that the Gödeliar seems true. That the latter twist is not minded or realized

shows very clearly from the widespread appreciation of those 'results', as is witnessed by the following quotes:

Malitz (1979): "(...) Gödel's famous incompleteness theorem of 1931, one of the milestones of mathematics" (p60) and "(...) one of the outstanding results of twentieth century mathematics, for this theorem caused a profound alteration in the views held about mathematics and science in general. No longer can mathematics be thought of as an idealized science that can be formalized using self-evident axioms and rules of inference in such a way that all things true are provable." (p128)

Kneebone (1963): "(...) in a terse but incisive paper, distinguished alike by astonishing originality, profundity of conception, and mastery of intricate detail, Gödel carried metamathematics over at a single stride into its second and more reflective phase." (p229)

Van Heijenoort (1967): "Gödel's paper immediately attracted the interest of logicians and, although it caused some momentary surprise, its results were soon widely accepted." (p594) and "There is not one branch of research, except perhaps intuitionism, that has not been pervaded by this influence." (p595)

DeLong (1971): "(...) the justifiably most famous results of contemporary logic: Gödel's first and second incompleteness theorems." (p160) and "Great achievements in science have often revolutionized philosophic thought, as can be seen in the work of Newton, Darwin, or Einstein. The philosophic impact of metatheory is likewise profound (...)" (p191)

Davis (1965), "This remarkable paper is not only an intellectual hallmark, but it is written with a clarity and vigor that makes it a pleasure to read." (p4)

In their 'popularization' of Gödel's theorems, Nagel & Newman (1975) speak about an "intellectual symphony". While it is noise.

In contrast to these views it is useful to consider a more mundane approach to solve the paradoxes of the Liar and Gödeliar. The following table gives a scheme of paradoxical relations that result into contradictions *if not handled properly*, plus the solution approaches. The difference between the simple and complex variant of a paradox is entirely psychological. In the simple variant one can directly identify the selfreference and then slash the equality sign. In the complex variant it takes a few steps to identify the selfreference. In two-valued logic one can simply forbid this kind of negative selfreference, in three-valued logic one can declare such instances to be senseless, i.e. without truthvalues True | False, and use the third value Indeterminate. Both methods actually apply to each cell of the table. Yet, given that the complex variant has been created precisely to allow for this kind of selfreference, it is more useful to associate two-

valued logic with the simple variant and three-valued logic with the complex variant.

	Two − valued logic	*Three − valued logic*
Liar	L = << L is false >>	The sentence in this block is false.
Gödeliar	G = << G is unprovable >>	Num[G] is the number of the statement that says that the statement with Num[G] is not provable.

The approach in this table clearly contrasts with the earlier quotes. Not only is the Liar solved but Gödel's theorems are diagnosed as consequences of a crippled Liar in a weak set of conditions, making them half-truths or fallacies.

Given that contrast, the reader is of course invited to study the argument in this book with care.

0.6.4 A logic of exceptions

Although the discussion below originated from a study in logic and an analysis of the Liar, it has appeared proper to extend it with discussions on the paradox of Russell, the theories of Brouwer, and the theorems of Gödel. The reason being that these topics appear to be useful applications of that *logic of exceptions*.

0.6.5 A textbook with news

A common format for textbooks is that they state accepted knowledge. News is relegated to the journals, and it may percolate there a few decennia before it sinks down into the textbooks. This textbook contains *basic* accepted knowledge and then continues with the news. There is no harm in that approach since the news follows logically from those basics. The main point to be aware of is that the whole development is presented as a textbook indeed. The novice can start from the basics and work up to three-valued logic and proof theory.

The textbook is based upon a 500 page draft textbook of 1980 in Dutch and a 100 page draft summary of the main results of 1981 in English. The argument could be recalled with only a little re-reading so that little had been forgotten. The subject has been approached wholly fresh again, again building up the development from the novice's point of view. The basis has been that English summary, neglecting lots of pages in Dutch. This procedure skips the need for exact translation and copying. With the available raw materials, texts in English have been created afresh and those should prove to be didactically clear. Historical quotes of course needed to be typed over. The

line of argumentation of 1980-1981 has not been changed however. The final text presented here has now been processed perhaps ten times (including the 1980-1981 period), so that one should hope for some didactic clarity.

Also, the author hasn't followed the literature since 1980-1981. Yet it is inevitable that there will be some development in the literature since then, on which this book cannot report. This is actually no cause for alarm, since, had the researchers in logic and philosophy in the last 25 years resolved the Liar and Gödeliar paradoxes (in the manner of this book) then surely this would have come to general attention and also this author would have noticed just from reading the newspapers. Given the apparent happy state of stagnation in these subjects the argument written 25 years ago is still as fresh as it can be.

0.6.6 Empirical base

The book closes with some notes on formalization. The author is an econometrician whose university education included the basic math courses also followed by students of math and physics at the University of Groningen. As my field remains econometrics, I do admit of having studied some logic and philosophy because of being caught by some fundamental issues, but I don't have the time nor the interest to formalize it all. The sound philosophy is that the schemes work so that they can be formalized if so desired. What is new for this author are the programs in *Mathematica* written during the holiday season of Chrismas 2006. Those neatly support the body of the text. The earlier version of *The Economics Pack* of the mid 1990s already contained simple propositional logic and inference but these have now been quite expanded, of which, given the table above, the now also practical implementation of three-valued logic of course might be judged didactically crucial.

0.6.7 How to proceed

Of course, once you have worked yourself up to Chapter 9 and 10, and the Reading Notes in Chapter 11, then only you, the advanced reader, will truly benefit from the main intellectual result of this book. This part gives a foundation and justification what we tell to the beginning students. You might find that this result might be challenging. The arguments in the introductory part may look easy or simple, but that is just the phrasing that I chose, and the argument is quite abstract. The accessible phrasing makes that the argument can be followed by the average student, but, in my experience, it takes a more abstractly trained mind to really understand the arguments. There still is value in a good education.

1. A direct hands-on introduction to logic

1.1 Learning by doing

The best introduction into logic is to do it.

You have been familiar with logic since birth but it can be exciting to look at it in a new way. The systematic way. This will give you a power of deduction that can startle you.

The various notations and routines will be explained in more detail in the remainder of the book. The idea of this part is that you first try to understand what happens. You can do that with some confidence. The important Stoic logician Chrysippus (280-207 B.C.) already reported that even some dogs are capable of reasoning. He saw a dog chasing after a spoor, coming at a fork with three ways, sniffing two and, without sniffing, racing up the remaining one.

Let us take a look at Not, And, Or, Implies, Equivalence, Xor, Unless, tautologies and contradictions.

1.2 Not

1.2.1 Language

Consider the sentence "John loves Mary" and the sentence "John doesn't love Mary". Notice anything particular ? Well, the two sentences are opposites. One sentence is the denial and negation of the other. Together they form a pair that gives all that is possible. If you were to suggest that there is a middle way such that John could love Mary a little bit or under some conditions then this still means the negation of "John loves Mary" in that he doesn't completely love her as the statement expresses.

1.2.2 Symbols

We will use the symbol Not to express this opposition. Thus if p = "John loves Mary" then Not[p] = "John does not love Mary". We might have defined p the other way around but once one assignment for p is chosen then we know what Not[p] stands for.

1.2.3 Switch model

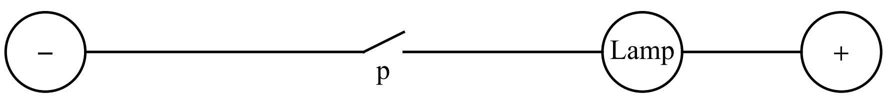
Electrical switches are a model for logic. The basic switch layout consists of the following elements (see the drawing below):

1. An electrical current flows for the negative pole (on the left) to a positive pole (on the right).
2. There is a lamp that will burn when there is an electric current and that will not burn when there is no current. (It is a good lamp.)
3. There is a switch that determines whether there is a current so that the lamp will burn. If the switch is "on" then the wire is closed and the lamp will burn. If the switch is "off" then the wire is open and the lamp will not burn.

Thus "the switch p is on" exactly covers "the lamp burns", and we can exploit this for a model of logic.

- The present graph shows the "off" position, thus Not[p]. This still leaves you free to interprete p in a positive or negative manner, or in any way that you conceive of the world (thus p = "Joe is not old" or p = "Joe is young").

BasicSwitch[p]



1.2.4 Truthtable

We can explain Not also by means of a "truthtable". This table lists all the states of the world that are possible and then gives the consequences for each separate state. The possible states for statement p are True | False. The lamp burns or it does not burn.

We are now so far in understanding Not that we introduce a special symbol for it. Not[p] will be denoted $\neg p$ since this makes for easier reading.

- Not[True] gives False, and Not[False] gives True.

TruthTable[Not[p]]

$$\begin{pmatrix} p & \neg p \\ \text{True} & \text{False} \\ \text{False} & \text{True} \end{pmatrix}$$

1.2.5 Calculation

A final way to understand Not is to use arithmetic. When you use truth values 1 and 0 and when you think of a p having only values 1 or 0 then Not[p] gives 1 - p.

- Truth and Falsehood can be transformed into 1 or 0 by means of the function *Boole* (named after an important logician).

?Boole

```
Boole[expr] yields 1 if expr is True and 0 if it is False.
```

- Boolean values 1 and 0 can be transformed back into True and False by using equations. When the value 1 expresses truth then asserting $\neg p$ means that $p == 0$.

¬ p // ToAlgebra

$1 - p = 1$

1.3 And

1.3.1 Language

Consider the sentences p = "John loves Mary", q = "John lives in London" and r = "John loves Mary and John lives in London". Again, do you notice anything particular ? OK, there may be more points to observe but what you should become aware of is the word "and", called *conjunction.* The sentence r can be seen as *p and q*.The compound sentence r is only true if both its constituents p and q are true.

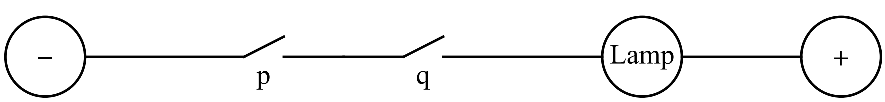
1.3.2 Symbols

We will use the symbol And or neater $\wedge$ to express the conjunction. Thus we can write $r = p \wedge q$.

1.3.3 Switch model

- This electrical circuit requires that the two switches are on before the lamp burns. This models $p \wedge q$.

AndSwitch[p, q]



1.3.4 Truthtable

The truthtable now keeps track of three statements. There are two states of the world for p, namely $\{p, \neg p\}$. There are two states of the world for q, namely $\{q, \neg q\}$. What happens with $p \wedge q$ depends upon the $2 \times 2 = 4$ combinations, namely $\{p, q\}$ or $\{p, \neg q\}$ or $\{\neg p, q\}$ or $\{\neg p, \neg q\}$. The light only burns when both switches are on, thus $\{p, q\}$.

- The statement $p \wedge q$ is only true if both substatements are true.

SquareTruthTable[p ∧ q]

$$\begin{pmatrix} (p \wedge q) & q & \neg q \\ p & \text{True} & \text{False} \\ \neg p & \text{False} & \text{False} \end{pmatrix}$$

1.3.5 Calculation

When we take the Boolean values 1 or 0 for p and q then $p \wedge q$ is the same as Min[p, q] == 1, the minimum of both.

- With values 1 or 0 we can also use multiplication.

p ∧ q // ToAlgebra

$p\, q = 1$

1.4 Or

1.4.1 Language

When x is positive then x^2 is positive. When x is negative then x^2 is positive as well. In fact, x^2 is positive for all x except 0. The solution for $x^2 > 0$ can be given as "x is positive or x is negative" and the "or" will be called *disjunction*.

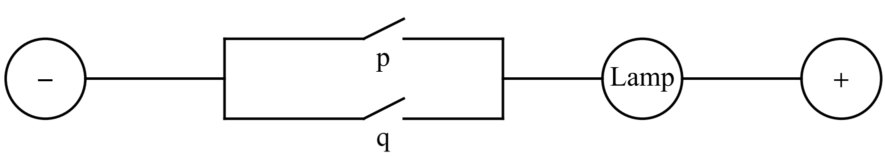
1.4.2 Symbols

We will use the symbol Or or neater ∨ to express the disjunction. Thus we can write $p \vee q$. The "∨" comes from Latin "vel" = "or". The "∨" for Or inspired the upside "∧" for And.

1.4.3 Switch model

- In this electrical circuit only one switch needs to be on before the lamp burns. This models $p \vee q$.

OrSwitch[p, q]



1.4.4 Truthtable

$p \vee q$ will be False only when the state of the world is $\{\neg p, \neg q\}$.

- The statement $p \vee q$ is only true if any substatement is true.

SquareTruthTable[p ∨ q]

$$\begin{pmatrix} (p \vee q) & q & \neg q \\ p & \text{True} & \text{True} \\ \neg p & \text{True} & \text{False} \end{pmatrix}$$

1.4.5 Calculation

When we take the Boolean values 1 or 0 for p and q then $p \vee q$ is the same as Max[p, q] == 1, the maximum of both.

- Algebra shows a dependency between Not, And and Or, since $p \vee q$ is the same as $\neg (\neg p \wedge \neg q)$.

p ∨ q // ToAlgebra

$$1 - (1 - p)(1 - q) = 1$$

% // Simplify

$$p + q = p\, q + 1$$

The algebra verifies that $p \vee q$ is the same as $\neg (\neg p \wedge \neg q)$, but we can also use the truthtables, and we can always resort back to language: “John eats pie or Charles watches television” is the same as “It is not the case that John doesn’t eat a pie and that Charles doesn’t watch television”.

1.5 Implies

1.5.1 Language

The sentence “If it rains then the streets are wet” contains the “If ... then ... “ conditional statement called *implication*. The first part is called the *antecedens* and the second part the *consequence.* Note that the streets can be wet because the city street cleaners passed by. The event that makes an implication false is when it rains and the streets are not wet. In all other cases the implication is true. An important situation is when it doesn’t rain: then the implication still is true (and it has a hypothetical nature).

1.5.2 Symbols

The implication is symbolized with Implies or neater ⇒. Thus we can write Implies[p, q] or $p \Rightarrow q$.

1.5.3 Switch model

The switch model for $p \Rightarrow q$ is actually the switch model for $(\neg p) \vee q$ since these two are equivalent. We can show this with a truthtable or in a calculation. We can also see this in language, where above statement on rain is equivalent to "It doesn't rain or the streets are wet". Recall the age-old line "Your money or your life".

1.5.4 Truthtable

- The statement $p \Rightarrow q$ is only false if p is true and q is false.

SquareTruthTable[p ⇒ q]

$$\begin{pmatrix} (p \Rightarrow q) & q & \neg q \\ p & \text{True} & \text{False} \\ \neg p & \text{True} & \text{True} \end{pmatrix}$$

- A trick is to substitute $\neg p$ for p and see that this gives the table for Or. NB. The ¬ affixes to the closest unit.

SquareTruthTable[¬ p ⇒ q]

$$\begin{pmatrix} (\neg p \Rightarrow q) & q & \neg q \\ p & \text{True} & \text{True} \\ \neg p & \text{True} & \text{False} \end{pmatrix}$$

1.5.5 Calculation

Implies can be represented with $p \leq q$. But solution algorithms for inequalities are different from those for equalities. A translation into equalities is using the equivalence with $\neg p \vee q$ (noting that the "not" affixes to the closest variable).

p ⇒ q // ToAlgebra

$q\,p - p + 1 = 1$

% // Simplify

$p\,q = p$

1.6 Equivalent

1.6.1 Language

Above, we already have been stating that one expression was equivalent to another. By equivalence we mean that the one is true if and only if the other is true. This "... if and only if ..." can also be expressed by "... iff ...". The equivalence also scores true when both sides are false.

1.6.2 Symbols

Equivalent[p, q] will be expressed with $p \Leftrightarrow q$. This expresses that an equivalence holds when both $p \Rightarrow q$ and $q \Rightarrow p$ hold.

1.6.3 Switch model

Actually the electrical switch for Not already relies on equivalence too. The light burns iff the switch is on.

1.6.4 Truthtable

- The statement $p \Leftrightarrow q$ is only true if both variables have the same truthvalue. In this case we must use the full expression "Equivalent" since the symbolic format $\Leftrightarrow$ is only defined for output. Note the True on the diagonal from upper left to lower right, and the False elsewhere.

SquareTruthTable[Equivalent[p, q]]

$$\begin{pmatrix} p \Leftrightarrow q & q & \neg q \\ p & \text{True} & \text{False} \\ \neg p & \text{False} & \text{True} \end{pmatrix}$$

1.6.5 Calculation

Equivalence is neatly expressed by $p == q$. NB. This has not been programmed but can be deduced.

ToAlgebra[p ~Equivalent~ *q*, All → True]

$\{(p\,q - p + 1)(p\,q - q + 1) = 1,\ (p = 1 \bigvee p = 0),\ (q = 1 \bigvee q = 0)\}$

Reduce[%, {p, q}]

$((p = 0 \bigwedge q = 0) \bigvee (p = 1 \bigwedge q = 1))$

1.7 Xor

1.7.1 Language

The sentence "Either I stay at home tonight or I'll give you a call to go somewhere" contains the "either ... or ..." that expresses that the two component sentences are mutually exclusive. The statements cannot be both true or both false.

1.7.2 Symbols

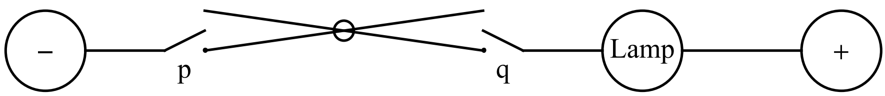
The symbol for "either ... or ..." is based upon Or, called Xor or neater $\veebar$. Thus we would write Xor[p, q] or $p \veebar q$.

1.7.3 Switch model

The Xor electrical circuit is called the “hotel circuit”. It is used along staircases so that you can control the light from downstairs as well as from upstairs, so that you don’t have to climb the stairs just to turn the lamp on or off. The light burns only when a single switch is on so that the other is off. This models $p \veebar q$. PM. In the graph, the lines in the middle cross but do not touch. Also, when a switch is open, then it connects to the upper wire. The drawing shows a small gap otherwise you might think that there is no switch but a permanent connection.

- The either .... or ... switch.

XorSwitch[p, q]



1.7.4 Truthtable

- Note the False on the diagonal from upper left to lower right, and the True elsewhere. Xor is the negation of Equivalent. Thus Xor[p, q] is equivalent to saying that $\neg (p \Leftrightarrow q)$.

SquareTruthTable[Xor[p, q]]

$$\begin{pmatrix} p \veebar q & q & \neg q \\ p & \text{False} & \text{True} \\ \neg p & \text{True} & \text{False} \end{pmatrix}$$

1.7.5 Calculation

Xor[p, q] can neatly be represented as $p \neq q$ provided that p and q are 0 or 1. Yet, solving inequalities is different from solving equalities, so we can better use p == 1 - q, understanding that each variable is 1 or 0.

ToAlgebra[Xor[p, q]]

$p + q = 1$

1.8 Unless

1.8.1 Language

The sentence “I’ll take an apple, unless you pay, for then I’ll take a big chocolate fudge” states the normal situation, an apple, unless a condition applies.

1.8.2 Symbols

We can use Unless[p, q, r]. When there is no consequence r then Unless is a form of Or.

1.8.3 Switch model

There is no particular switch model, except the main switch down in the basement to turn off the whole system in case something is wrong.

1.8.4 Truthtable

- Without r Unless is just Or. A StrongUnless[p, q] is ($\neg q \Leftrightarrow p$).

SquareTruthTable[p ~Unless~ q]

$$\begin{pmatrix} (\neg q \Rightarrow p) & q & \neg q \\ p & \text{True} & \text{True} \\ \neg p & \text{True} & \text{False} \end{pmatrix}$$

- Unless works best with three variables. Just like equivalence it appears to be a conjunction of two Implies'es. Because of the third variable the square truthtable prints differently and hides the $\{r, \neg r\}$

SquareTruthTable[Unless[p, q, r]]

$$((\neg q \Rightarrow p) \bigwedge (q \Rightarrow r))$$

$$\begin{pmatrix} & q & \neg q \\ p & \{\text{True, False}\} & \{\text{True, True}\} \\ \neg p & \{\text{True, False}\} & \{\text{False, False}\} \end{pmatrix}$$

1.8.5 Calculation

The calculation scheme is just the combination of the earlier logical symbols. This may be one of the reasons why common books on logic tend to forget about Unless, even though it plays an important role in logic as we shall see below.

ToAlgebra[Unless[p, q, r]]

$$(p\,(1 - q) + q)\,(r\,q - q + 1) = 1$$

1.9 Tautologies and contradictons

In these diagrams there are only one or two points where we created a switch. These represent contingent statements that can be true or false. All other points are *connected* or *unconnected*.

- All points in the wire itself can be seen as switches that are always closed, and thus as statements that are "always true". These statements will be called *tautologies*.

- All points "not drawn" can be seen as switches that are always open, and thus as statements that are "always false". These statements will be called *contradictions*.

When we build an electrical switchboard such that the lamp always burns whatever switch we flip, then we might as well put in a direct connecting wire. When we build a switchboard such that the lamp never burns whatever combination of switches we try, then we might as well take away all switches and wires.

- This is a tautology (always true). TruthTableForm reads from right to left.

TruthTableForm[p ⋁ ¬ p]

Or	*p*	Not *p*
True	True	False True
True	False	True False

- This is a contradiction - it is always false.

TruthTableForm[p ⋀ ¬ p]

And	*p*	Not *p*
False	True	False True
False	False	True False

1.10 Testing statements

Combinations of logical operators can be tested by the truthtables, by solving algebraic equations or by calling a solver like LogicalExpand or Simplify. It is possible to construct all kinds of electrical circuits that mimic complex propositions, and determine their truthvalues. In fact this is what happens in computers. The electrical switches shown to you however are only pictures and these routines cannot be combined to create a switchboard.

Calling a solver directly prints the answer but it hides all the intermediate steps. When you are beginning with the systematic study of logic then you would be interested in those steps to check them. The truthtable method then is the best approach. The above truthtables used a square format but when statements have a different number of variables then it is better use a full table form, where all states of the world are put into separate blocks, and where you might check the steps in the evaluation.

We said that Xor is the negation of equivalent. Let us check that.

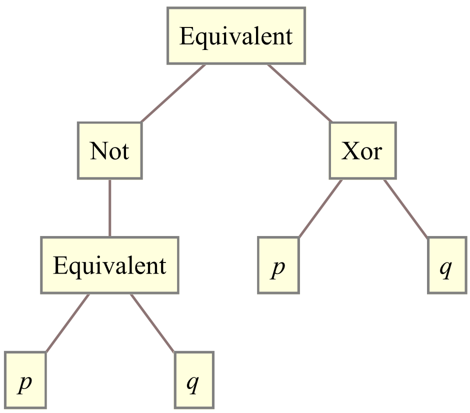
- This is the statement that we will check.

try = Equivalent[Xor[p, q], ¬ Equivalent[p, q]]

$\neg (p \Leftrightarrow q) \Leftrightarrow p \veebar q$

- This is the structure of the statement.

TreeForm[try]



- This is the truth table in extensive form, that reads from right to left. Each block gives the True | False combinations of the variables and the operators that they occur in. The left column shows only True, so that the equivalence that we tested is always true: a tautology.

TruthTableForm[try]

Equivalent	Not Equivalent p q	Xor p q
True	False True True True	False True True
True	True False True False	True True False
True	True False False True	True False True
True	False True False False	False False False

1.11 A summary of what logic is

Assuming that information is expressed in sentences, then we can observe that some expressions are always true, others always false, and again others sometimes true and sometimes false. We can also observe that the property of always being true or always being false depends entirely upon the structure of the sentence, not upon the actual state of the world. Using the concept of logical necessity (and thus not physical necessity) we find the following distinction:

1. Sentences that are always true = laws of logic = tautologies = necessarily true
2. Sentences that are always false = anti-laws of logic = contradictions = necessarily false
3. Sentences that are sometimes true and sometimes false = contingently true or false

The subject of logic consists of investigating when deductions are valid, when they are invalid, and when they are inconclusive. Part of that enquiry concerns determining when statements are necessarily true, necessarily false, or contingent by their very structure and their relation to inference itself.

1.12 You can directly use a main result

Since *Mathematica* is so easy to use, you can directly use a main routine and result in this book. That main result is Decide[text] that can extract new conclusions ("the news"). A small example gives a direct introduction into the logical issues and it shows how you can apply the routines. (See "Getting Started" in the very beginning first if you really want to run the programs.) The routine Decide works only for propositional logic, and the text must be formulated in "Englogish", a simplified English-like language. Englogish statements still can have upper case letters and punctuation. Negation is only recognized when "Not" appears at the beginning of a sentence. Englogish also uses "If ... then ..." instead of the more pedantic "Implies".

The following examples use the inference scheme called *modus ponens.* When our information set is $\{ p \Rightarrow q \, , \; p \}$ then we infer q as the news. The news is what did not occur as a separate entry in the list and that we thus were not aware of.

`Decide[x, opts]`	performs Conclude[Deduce[Conclude[x], opts]]. The procedure automatically turns on AndOrEnhance

Englogish will be discussed below in more detail, and similarly the procedure of "AndOrEnhance".

- The extraction of a decision may be possible by Decide.

Decide["If it rains then the streets are wet. It rains."]

AndOrEnhance::State : Enhanced use of And & Or is set to be True

{ the streets are wet}

- Recall: the symbol "¬" expresses "Not", so that Not[p] or $\neg p$ expresses the negation of p.

Decide["If the apple falls then it hits my head. The apple falls. If it hits my head and I am Newton then I invent gravity. If I invent gravity then I am Newton. Not I am Newton."]

{¬ i invent gravity, it hits my head}

- When you use "If ... then ...", "not", "or" and "and" as operators then you can use foreign languages.

Decide["If il pleut then je t'aime. Il pleut."]

{ je t'aime}

- Socrates doesn't want to see Plato unless Plato wants to see him. Plato doesn't want to see Socrates if Socrates wants to see him. Plato wants to see Socrates if Socrates doesn't want to see him. Well, first check what *unless* means:

(Not["Socrates wants to see Plato"] ~Unless~ "Plato wants to see Socrates")

(¬ Plato wants to see Socrates ⇒ ¬ Socrates wants to see Plato)

"If not Plato wants to see Socrates then not Socrates wants to see Plato. If Socrates wants to see Plato then not Plato wants to see Socrates. If not Socrates wants to see Plato then Plato wants to see Socrates." // Decide

{¬ socrates wants to see plato, plato wants to see socrates}

- Much reasoning relies on counterfactuals. The best way to deal with these in the environment of the `Logic`` package is to qualify what one supposes. The following is a valid argument, a reductio ad absurdum i.e. proof by contradiction, on the counterfactual like "The streets would be wet, wouldn't they, if it rained ?":

"Suppose that it rains. If suppose that it rains and our suppositions fit reality then it rains. If it rains then it is wet. Not it is wet." // Decide

{¬ it rains, ¬ our suppositions fit reality}

- Turn off the enhancement of And and Or. PM. This will be explained below.

AndOrEnhance[False]

AndOrEnhance::State : Enhanced use of And & Or is set to be False

1.13 Paradoxes, antinomies and vicious circles

Next to the beauty of symbolic logic, one other motivation to study logic are the paradoxes, antinomies and vicious circles that arise both in daily life and in science. Some are more than 2000 years old and they actually helped stimulating mankind to develop modern logic, mathematics and science.

Regard the following statements:

1. Don't look here !
2. Be spontaneous !
3. Generalisations are not bad by themselves. You just shouldn't make bad generalisations. This is of course a generalisation.
4. A teacher, one day: "2 and 2 is 4", and another day: "2 and 2 is 22".
5. I lie. (The Liar paradox, Greek "pseudomenai". If "I lie" is true, then I lie. When "I lie" is a lie, then I speak the truth.) Variants are "This sentence is not true" or "This is a contradiction".
6. Socrates: "What Plato says is not true." Plato: "What Socrates says is not true." They don't say anything else.
7. "She watched television and missed the train" is logically the same as "She missed the train and watched television".
8. Hopelessly in love but confused: "I can only stop thinking about you by thinking that I don't want to think about you anymore, but then I still think about you; so let me try not to think that I don't want to think about you."
9. When there are no square circles, then how can I think and say so ?
10. A theory determines what we observe empirically. But aren't theory and empirics opposed ?
11. When morals rule, horrible things happen.
12. Cervantes, "Don Quichote", the passage on The Brigde: In the Land of the Just, a law is passed that visiting strangers have to state the reason of their visit, and liars will be hanged. The border of the Land of the Just is given by a river, and next to the bridge across it a Court is established to judge visitors on the new law. One day a stranger arrives who states: "I have come to be hanged."
13. Some issues cannot be described - but is that not an adequate description itself ?
14. A catalogue lists various cases. Sometimes the catalogue will mention itself, sometimes not. What about making a catalogue of all catalogues that don't mention themselves ? Will that new catalogue mention itself or not ? (Bertrand Russell's set

paradox.)

15. An age old story of the crocodile: A crocodile has taken a child and has been hunted down by the father of the child. The beast reacts: "I promiss to return the child to you, if you properly predict whether I will return the child to you or not." The father, showing his wit: "You will not return my child to me."
16. "This sentence counts exactly six words" is true, and thus "This sentence does not count exactly six words" is false.
17. Theories of logic are always consistent. This is proven as follows. They are either consistent or inconsistent. When they are consistent, then they are consistent. When they are inconsistent, then they can prove anything, also that they are consistent. In both cases they are consistent.
18. A problem for logic are the nonsensical utterances of idiots and politicians. But the term "idiot" (somebody who doesn't know about things that matter) was introduced originally by the Greek for those who did not participate in the discussions on the management of the state, i.e. those who didn't participate in politics.
19.

```
StringLength["The number of Berry is the smallest natural number that cannot \
be defined by at most 108 numbers or letters."]
```

108

Some of the paradoxes above are in the realm of pragmatic psychology, see Watzlawick et al. (1967). In that case the focus is on the psychological aspects of paradoxical communication, notably on complex and possibly contradictionary information, that even might contribute to mental disorder. Legend has it that some Philetas of Cos even died as a result of his hard thinking on the Liar Paradox. In that respect the reader has been warned. Watzlawick et al. provide dramatic reading, in particular since they don't emphasize the logical solvability of various issues.

However, all these different fields are too much for a simple book on logic and thus it will be wise to limit the field of application.

PM 1. The logician, poet and grammarian Philetas of Cos (ca. 340-285 BC) is reported to have said: "Traveller, I am Philetas; the argument called the Liar and deep cogitations by night, brought me to death." Apparently speaking in the after life. See Bochenski (1970:131) and Beth (1959:493).

PM 2. Hughes & Brecht (1979) give an anthology of paradoxes.

PM 3. Many people use "paradox" for "contradiction" but the proper interpretation is a "seeming contradiction", meaning that it impresses us as a contradiction at first while closer examination clarifies some hidden logical construction. From the Greek "doxa" (meaning, expectation) and "para" (next, beside), thus like being "beside the point" (that is

expected). The word “antinomy” comes from “nomos” (law, custom) and “anti” (against). See Curry (1963).

1.14 A note on this introduction

Many *Introductions to Elementary Logic* start with longer texts before they introduce the logical operators. Above we have presented them at the very beginning. The idea is that readers are familiar with logic since birth so that they can easily grasp symbolic logic without much ado. Longer introductory texts tend to bore and then kill interest by obscuring a core contribution of symbolic logic. By beginning with the symbolic representation, readers will see “hey, I learn something” and will grow interested to know more about the subject. What is especially nice about the logical operators is that they have models in various representations (language, symbols, circuits, tables, equations) and that they combine in surprising ways. Symbolic logic is a gem of beauty, and, the fun of it is too that it greatly enhances reasoning power.

That being said, the following unavoidably continues with longer texts. The above has been a quick introduction but we have been sloppy and have been assuming all kinds of things. Let us try to reconsider what we have been doing and let us try to approach the matter in a more systematic manner.

Part II. Two-valued logic

The five chapters in this Part assume two-valued logic. Sentences can only be True or False. In Part III we consider some alternative approaches.

2. Logic, theory and programs

2.1 Prerequisites

2.1.1 A prerequisite on notation

Single quotes identify a concept and double quotes (quotation marks) identify a string of symbols. Thus ‘poverty’ is the concept of poverty, and “poverty” is just the string of symbols “p”, “o”, “v”, etcetera. The next point is to distinguish clearly between the using (asserting) and the mentioning (listing, quoting) of a sentence - as the ancient Greek already noted how an orator might read aloud a speech without knowing what it means. Socrates compared reasoning as speaking to oneself, in one’s soul, and so that one would not lie to oneself. In this same way asserting is speaking up, claiming truth for what is uttered. The following convention then is useful: (a) when a sentence is asserted or used then it occurs without quotation marks, (b) when a sentence is merely referred to then the sentence will be put between quotation marks. In that case the string of symbols that form the sentence is regarded as an object of itself. Let us use italics (e.g.“*p*”) to denote variables for sentences. Thus “p” is just the letter and *p* is a sentence such as “It rains.”. We can write *p* = “It rains.” to assign a constant string of symbols to a variable; *q* = Assert[“It rains.”] then is the proposition that it rains (with propositional variables and constants).

PM 1. This convention assumes that one will see from the context whether we intend to quote someone (e.g. John said: “Be there in time”) or whether we wish to apply this distinction between mention and use. We also depend upon the context to prevent confusion between referring to sentence *p* and using italics to give *emphasis* to an expression.

PM 2. In *Mathematica*, sentences are called Strings and you can manipulate Strings with various functions. This is not evaluated here, but if you do this in *Mathematica* then it shows a list of String functions.

```
?*String*
```

PM 3. Sentence variables like *p* can be identified by different typesetting, conventionally by the use of *italics*. If we didn’t do that then we would have to use quotes within sentences,

so that "The sentence "It rains" is true." would become "The sentence "p" is true", and then we would need to find a way to express that "p" would be a variable rather than the letter "p". Italics are more convenient.

PM 4. Socrates' insight of getting rid of all the noise and untruths in human discourse and concentrating on a person's own internal deliberation is no mere point. The proper attitude for science is to let all arguments sink through your mind so that you can arrive at your own internal conviction of what you think is the case. Let no one sway you with anything other than a better argument. And take the time to think it through. Another case is the position of a judge in court. The judge must not only establish the evidence but must also observe within himself or herself whether this convinces beyond reasonable doubt. This *conviction intime* (French) is one of the legal prerequisites before there can be judgement.

PM 5. Yet, Socrates' idea might be merely a begging of the question, when the situation rather is that many minds prefer meaning above sense. In other words, people can fool themselves, and Nature remains the best model.

2.1.2 Logic and the methodology of science

This book deals primarily with logic. However, discussions require examples, and in this book the methodology of science will function as the example application. For example, when scientists come up with a hypothesis or conjecture then it might be hard to prove it for all circumstances, but what they can do is design experiments targetted at refuting it, so that, if it passes the tests, they have more ground to attach value to it. This procedure is grounded in logic, and it is useful to see how that is so. There are various other issues in the methodology of science that logic can illuminate and that are illuminating for logic. In fact, logic and the methodology of science may be more intimately linked than merely as example.

A key issue is that when we reach a contradiction then some of the things discussed cannot exist (Schröder 1890, cited in Bochenski (1970:366)). Thus logic and mathematics have some impact on empirics: contradictions and impossibility proofs help researchers drop inconsistent hypotheses.

2.1.3 *Mathematica* - also as a decision support environment

Mathematica is a language or system for doing mathematics on the computer. Note that mathematics itself is a language that generations of geniusses have been designing to state their theorems and proofs. This elegant and compact language is now being implemented on the computer, and this creates an incredible powerhouse that will likely grow into one of the revolutions of mankind - something that can be compared to the invention of the wheel or the alphabet; at least, it registers with me like that. Note that, actually, it is not the invention of precisely the wheel that mattered, since

everybody can see roundness like in irisses, apples or in the Moon; it was the axle that was the real invention. In the same way next generations are likely to speak about the 'computer revolution', but the proper revolution would be this implementation of mathematics.

Mathematica already is a decision engine of a kind. If you run some algebraic solution routine then there is a lot of deduction before the answer pops up. However, that answer does not come as a neat English expression and does not read as a conclusion in the way that a good speaker would summarize his or her speech. The idea here is, then, to extend *Mathematica*'s powers as a decision machine, and to equip the system with linguistic deduction. Note also that we properly regard *Mathematica* as a language itself too, and hence we are in the subject field of better integrating language and thought.

Suppose that you have a block of text, for example a text that summarises your research results. Would it not be nice to submit it to your computer, and have it verify whether it is logically sound or errs against logic ? The `Logic`` and `Inference`` packages are a step into that direction. Admittedly, there still is a long way to go, but if we don't make the first steps we will not get far at all.

2.1.4 Use of The Economics Pack - Relation of logic to economics

The `Logic`` and `Inference`` packages used here are included in *The Economics Pack*. It might actually require an explanation why economics turns up. The basic reason is that both subjects are dear to the author, who thinks to see a deep connection somewhere, and who enjoyed writing the book and the programs.

There seem to be some historical parallels. For an economist it is always nice to remember that John Neville Keynes, the father of John Maynard Keynes, taught logic and wrote a book on the subject. J.M. Keynes was inclined to logic and reasoning, at least, as he himself wrote a treatise on probability and opposed his way of thinking to that of Jan Tinbergen and Ragnar Frisch who developed econometrics and the methods of number crunching. Economists thus might be interested in logic and inference, not only for the methodology of science, but also for becoming better economists (who understand both logic and number crunching).

Interesting is also this observation by Schumpeter (1967:103-104): "Now it happens to be, that the rationalistic attitude has forced itself on the human mind in the first place out of the economic necessity; it is from the everyday economic problem that we humans derive our elementary training in rational thinking and acting - I do not hesitate to say, that all logic is derived from the pattern of economic decision making, or, to use a favorite expression of mine, that the economic pattern is the womb of all logic." It is kind of hard to see economics as the theory of the womb of all logic. Similarly, a baby

experimenting with "If I don't eat my porridge but pull the tablecloth then everything will fall on the ground" does not seem to be much involved in the economic pattern and rather learns its logic from plain Newtonian physics. But the quote is a nice one.

The key subject of economics is the analysis of social welfare. Part of social welfare is determined by the market mechanism by which transactions are conducted with money. Each market transaction is Pareto improving - and a Pareto improvement is defined as a change such that some people advance while nobody sees his or her position deteriorating. Each market transaction is voluntary, and if the participants would not see an improvement, they would not partake in it. This property is so interesting that we would like to see it also in other aspects of social welfare. One such other part of social welfare depends upon group decisions in which voting occurs. We see it as one of the ways how people aggregate their preferences to arrive at a social optimum. For example, in democracies, decisions on taxes or on government expenditures are influenced by the ballot. See Colignatus (2001, 2007) on Voting Theory for Democracy. (PM. That book uses a *fixed point* to solve voting paradoxes, and the "fixed point" idea was originally developed by L.E.J. Brouwer who gets a chapter in this current book.) A third aspect of social welfare consists of plain talk, social conventions, psychology etcetera - but this we do not deal with here (see however Aronson (1992ab)). Logic obviously occurs in all these areas of economics.

A special application is the logic of morals, also called "deontic logic". The section on deontic logic in "Voting Theory for Democracy" is reproduced here below since it belongs to logic as well as. There are two other section of Colignatus (2005) that would have found good reproduction too but that are left out for the time being. The first is the discussion of the *definition & reality methodology*, see the summary below under induction. The other is on the triad *determinism, volition and chance*, see the summary below on Chapter 7 on three-valued logic.

This said, it should be clear that this book has not been written for economic students only. It is intended to be used by anyone entering into any serious study.

2.2 Logic environment in *Mathematica*

2.2.1 This book has been written in *Mathematica*

This book has been written in *Mathematica*, a system for doing mathematics. That program is both a text editor and a calculator at the same time. The text produced here is not only what the author has typed but also what the programs have generated. Those programs have been written to produce that output. You can change the input in

Mathematica and generate different results.

Crucial notations to know are:

- $x = a$ means that variable x gets the value a, $x == a$ means the logical statement that x and a are identical
- % refers to the result of the former evaluation
- a function call can be entered as $f[x]$ or as x // f
- x /. r means that substitution rules r are applied to x

Input in *Mathematica* are not just numbers but can be structured objects. That's why users of *Mathematica* rather don't speak about mere calculation but about *evaluation*.

2.2.2 Notation

One consequence of using *Mathematica* is that we will use its notation so that it can understand our formulas. The following propositional operators and functions are available standardly in *Mathematica*. The program also has an extensive Help function so that you can find more details on these and other concepts. In fact, the full text of this book is available in *The Economics Pack*, and thus can be found as an Add-On Application in the Help function of *Mathematica.*

¬ *p* or Not[*p*] or !*p*	not ; here ¬ is found by ESC not ESC
p ∧ *q* ∧ ...	and ; also input as And[*p, q,* ...] or *p* && *q* … Here ∧ is found by ESC and ESC
p ∨ *q* ∨ …	or ; also input as Or[*p, q,* ...] or *p* ‖ *q* … Here ∨ is found by ESC or ESC
p ⊻ *q* ⊻…	exclusive or (either ... or ...) ; also input as Xor[*p*, *q*, …] . Here ⊻ is found by ESC xor ESC
Nand[*p*, *q*, …] and Nor[*p*, *q*, …]	nand and nor (also input as ⊼ and ⊽)
p ⇒ *q*	implies ; here ⇒ is found by ESC => ESC
If[*p*, *then*, *else* (, *otherwise*)]	give *then* if *p* is True, and *else* if *p* is False (and if none of these then *otherwise*)
LogicalExpand[*expr*]	expand out logical expressions

Above the line are "operators" and below the line are "functions". The variables *p* and *q* are expressions in *Mathematica*. The symbol ∨ comes from Latin "vel" = "or". The symbol & comes from Latin "et" = "and" written in a special way. The symbol "!" was inspired by "?", derivative of "Q." (a Q with an abbreviation point) standing for Latin "Quaestio" = "I question", so that the question mark originally was a sentence by itself. In *Mathematica* you must enter Implies as the above and not with ESC => ESC (note the blank before the arrow) which just gives a double right arrow that merely looks the same.

Note the difference between Not that has one-dimensional input, Implies that has two-dimensional input, and And and Or that can have more-dimensional input. Implies, And and Or are all called "binary" operators since that is how they have been originally defined, while the extension of And and Or is a feature of *Mathematica*.

PM 1. When you evaluate !p with "!" at the beginning of a line then old versions of *Mathematica* started printing or reading some file p. Now, see Run.

PM 2. The logical symbol And is without connotation of time or causality. In natural language there is such a connotation but we will neglect it. Compare "She watched television and missed the train" and "She missed the train and watched television". Or: "John and Mary married and Mary became pregnant" with "Mary became pregnant and John and Mary married."

2.2.3 Input and evaluation

Next to plain text of the text editor there are also "input" and "output" cells for the evaluator. You enter commands in input cells (shown in bold type) and the result of the computer evaluation is printed below it (shown in "traditional form").

- *Mathematica* can recognize some statements that can be evaluated directly to be True or False. Other statements may remain unevaluated when they don't supply sufficient information.

1 + 1 == 2

True

p ∧ q

$(p \wedge q)$

2.2.4 Full form and display

For pattern recognition, objects need to have a fixed format, called their FullForm. This form reads easier for computer programs but less easy for the human reader. Hence, the output cells can be displayed in a different form. When working with *Mathematica* you should always have the FullForm in mind.

- Implication has three input formats: (1) the FullForm, (2) an infix form, (3) when you type ESC = > ESC then *Mathematica* creates a neat arrow that also stands for Implies. In this case variables p and q are undefined and the relation remains unevaluated. The FullForm in these cases are all the same. NB. Don't mix up Implies with just a double arrow.

Implies[p, q]

$(p \Rightarrow q)$

p ~Implies~ q

$(p \Rightarrow q)$

p ⇒ q

$(p \Rightarrow q)$

% // FullForm

Implies[p, q]

2.2.5 Logical routines

Another consequence of using the *Mathematica* environment is that we depend upon its evaluation conventions. If you type a contradiction in an input cell then *Mathematica* does not immediately compute a falsehood in output. The reason is that some expressions are not meant to be entirely deductive. For example, in equations x == 0 and x != 0 you want to find an empty solution set {} and not False. Similarly, *If* is rather a (meta) control function while *Implies* belongs to the logical object language. For applications of logic, a positive benefit of this weak definition is that this allows for a multi-valued logic.

■ Entering a contradiction does not evaluate to a falsehood.

p ∧ ¬ p

$(p \wedge \neg p)$

■ But the routines LogicalExpand and Simplify find it.

% // LogicalExpand

False

However, And and Or can still simplify a little bit and detect conditions that are always fulfilled.

■ *And* is False if one component is False.

And[p, q, True, s]

$(p \wedge q \wedge s)$

And[p, q, False, s]

False

■ *Or* is True if one component is True.

Or[p, q, True, s]

True

Or[p, q, False, s]

$(p \vee q \vee s)$

Mathematica's If is essentially a control function. It works best with three inputs so that it is better understood as *If* ... *then* ... *else*.... Actually the best way to understand this is to write some programs in *Mathematica*. If you cannot do so then the following should

explain it all.

- Function "If" is a control function with a result that depends upon input.

```
If[1 + 1 == 2, "Wonderful", "Too Bad"]
```

Wonderful

```
If[1 + 1 != 2, "Wonderful", "Too Bad"]
```

Too Bad

```
If[whatever, "Wonderful", "Too Bad"]
```

If[whatever, Wonderful, Too Bad]

```
If[whatever, "Wonderful", "Too Bad", "None Of These"]
```

None Of These

On the other hand *Implies* is a binary operator with limited freedom since it will evaluate to True if the antecedens is False. You may check the latter by considering “If Christmas and New Year fall on one day then you’ll get that new car” which is a promiss that is always true (though vacuously so).

- Implies is a binary operator that gives True if the antecedent is False. If the antecedent is True then the consequence must also be True.

```
Implies[1 + 1 == 2, "Wonderful"]
```

Wonderful

```
Implies[1 + 1 != 2, "Wonderful"]
```

True

```
Implies[whatever, "Wonderful"]
```

(whatever ⇒ Wonderful)

```
Implies[whatever, "Wonderful", "Too Bad"]
```

Implies::argrx : Implies called with 3 arguments; 2 arguments are expected. More…

Implies[whatever, Wonderful, Too Bad]

2.2.6 Getting used to *Mathematica*

The above only gives the basic necessities that you require for understanding the notation, texts and programs below. If you encounter problems below on how issues are implemented in *Mathematica* then it is advisable to dwell a bit longer on them, since discovering more about *Mathematica* is an investment that can pay off in various subjects. One key advantage is that you can write your own programs once you become comfortable with the language. A good way to look at *Mathematica* is to regard it as a

language indeed (and not just a computer program).

2.3 The subject area of logic

2.3.1 Aims of this book

This paragraph is only to remind you of the aims set out on the first page of the book.

2.3.2 The subject of logic

The basic idea is that people handle information. They base judgements on former judgements, so that there is a dependence between the premisses and the conclusion of an argument. We can use $S \vdash p$ to denote that some system S proves some statement p, or that some person S (Socrates) asserts p. What is important here is that a proof contains the notion of *justification* so that $S \vdash p$ expresses that S is justified to assert p on the basis of his former assertions. It might be conceivable to eliminate S and include all those former assertions, so that we only mention the premisses and the conclusion, and so that we can directly judge whether the deduction is correct. It then becomes irrelevant whether S performed the deduction or claimed the proof and indeed asserted the final result. Rather, we are become interested whether the deduction is valid or invalid. If the deduction is valid and if we also accept the premisses, then we accept the conclusion too. In this book we assume that S indeed could be equivalent to the relevant statements on p.

Logic is a great liberating force, since we do no longer accept statements on basis of authority, but on inference, while there are objective ways to check upon the validity of inference.

Remember what we said in Chapter 1: The subject of logic consists of investigating when deductions are valid and when they are invalid and when inconclusive; and investigating when statements are necessarily true, necessarily false, or contingent by their very structure and their relation to inference itself.

2.3.3 Statements versus predicates, statements versus inference

2.3.3.1 Four combinations

At the very outset, you should be aware of the following four subcategories, with the numbers showing the order by which they will be discussed in this subsection:

Aspects of logic	Statements	Predicates
Statics (formulas)	(1)	(2)
Dynamics (inference)	(3)	(4)

A statement is a block of sentences. A single sentence is also a statement.

The logic that deals with statements only is called propositional logic. The logic that deals with statements and predicates is called predicate logic. The logic that was presented to you in Chapter 1 was propositional logic only. Statements are molecules consisting of atomic statements. As an atom consists of electrons and protons, an atomic statement consists of both the subject of the statement and the predicate in the statement.

2.3.3.2 Static statements

(Ad 1) An example of a static statement is "If it rains then the streets are wet" when it is formalized as "If P[1] then P[2]" or "P[1] ⇒ P[2]", and thus not analyzed in further detail. Here P[1] = "it rains" and P[2] = "the streets are wet". The "If ... then ..." between quotes is *implication* and should not be confused with *Mathematica*'s If. Thus, raining *implies* that the streets are wet. Claiming such a (hypothetical) relation differs from concluding that the streets are actually wet.

The atomic statements are joined to form compound statements (molecules) using the connecting operators of propositional logic (And, Or, Implies, Equivalent, Unless, Xor, ...).

2.3.3.3 Statics with predicates

(Ad 2) An example of an atomic statement is P[3] = "Socrates is mortal". It becomes a "static predicate case" when we consider the subject Socrates and the predicate Mortal as objects by themselves. Similarly, P[4] = "All men are mortal" contains the subject "men" and the predicate "mortal".

2.3.3.4 Dynamics with statements

(Ad 3) An example of "propositional inference" is the following (*modus ponens*):

English	**Analysis in propositional logic**
If it rains then the streets are wet.	If P[1] then P[2]
It rains.	P[1]
Ergo	——————
The streets are wet.	P[2]

The statements in the rows are mere sentences (Statics) while the steps across the rows are a deduction (Dynamics). The "If ... then ..." of the first row may be seen as belonging

to Statics, and is translated with ⇒ (Implies). "Ergo" itself may be translated as (i) the line itself, or (ii) the statement "I decide", or (iii) "Thus", or (iv) best kept as it is, to reflect the formal status of the deduction. In many natural languages like English the word "Thus" or "Therefore" is included in the conclusions statement, possibly to emphasize that this is the conclusion indeed. We better analyze Ergo as a function concerning the whole argument. We can reproduce the inference above as follows:

- The two-dimensional format is useful for presentations. This routine does not reason itself. The user must put in all steps and the routine just displays.

Ergo2D[P[1] ⇒ P[2], P[1], P[2]]

1	$(P_1 \Rightarrow P_2)$
2	P_1
Ergo	————————
3	P_2

It is also useful to have a linear format. We can display Ergo with the ⊢ symbol to express that both the assumptions and the conclusion are asserted.

Ergo[P[1] ⇒ P[2], P[1], P[2]]

$(\{(P_1 \Rightarrow P_2), P_1\} \vdash P_2)$

While ⇒ leads to longer statements (from p and q we get $p \Rightarrow q$) we find that ⊢ can cause shorter statements.

The notion of dynamics comes from the observation that we might include an index t for the steps in the argument. Sometimes people first state the conclusion and then build their case with the premisses, so that the temporal order isn't relevant but the steps in the inference. Notably:

Ergo[$(\mathbf{P[1]} \Rightarrow \mathbf{P[2]})_{\mathbf{t}}$, $\mathbf{P[1]}_{\mathbf{t+1}}$, $\mathbf{P[2]}_{\mathbf{t+3}}$]

$(\{(P_1 \Rightarrow P_2)_t, (P_1)_{t+1}\} \vdash (P_2)_{t+3})$

For an elementary development of logic we will neglect the time order and merely distinguish static statements versus inference. The only "dynamics" that remains is the two-step distinction between premisses and conclusion.

In the Ergo[...] object the commas before the conclusion can be seen as another expression of the conjunction And, while the final comma can be seen as an expression of Implies. This interpretation can be called "projection". An inference is called *valid* iff (if and only if) the projection of the inference into static propositional logic produces a tautology, i.e. a relation that is always true.

Ergo[P[1] ⇒ P[2], P[1], P[2]] /. Ergo["Projection"]

$(((P_1 \Rightarrow P_2) \wedge P_1) \Rightarrow P_2)$

When an inference is invalid then ¬ ($p \vdash q$). It appears to make for better reading to give this a separate symbol of itself, in display only.

- This is a way to express that the inference is invalid. NonSequitur is Latin for “it does not follow”. In two dimensional display you would cross out the whole scheme.

NonSequitur[p, q]

(*p* ↧ *q*)

`Ergo[p___, q]`	displays only, with ⊢ expressing that the assumptions *p* result into the conclusion *q*. It differs from Implies[*p*, *q*] in that the individual assumptions are all asserted so that the conclusion is not hypothetical
`Ergo[p]`	expresses that *p* is a theorem. Ergo can also be entered as Thus
`NonSequitur[p___, q]`	gives ¬ Ergo[*p*, *q*] and displays with ↧ (no explicit Not) expressing that the assumptions *p* do not support the conclusion *q*. Not to be confused with Ergo[*p*, Not[*q*]] which is given by Refuted[*p*, *q*]
`NonSequitur[p]`	expresses that p is a rejected conclusion, either false or yet unproven
`ErgoRules[]`	contains simplification rules on Ergo

Ergo means *thus* in Latin and *non sequitur* means *it does not follow*. These terms have been chosen to emphasize the formal nature, as in a court of law. Ergo is also taken as “Proves”. *Mathematica* recognizes only Ergo but for texts you could use ESC rT ESC. Ergo2D has the same input structure as Ergo but then prints in lines. SetOptions[Ergo2D, Label → Blank] leaves out line labels, and use Label → {...} for your own list.

You may have heard the phrase “to the contrary ....”. The point is that when you conclude that *p* does not prove *q* then it does not need to be the case that *p* proves *not-q*. A proof for *not-q* would be a refutation. Thus next to *contradiction* we also have *contrariness*. The following table sums up what that means. It has been in use since medieval times and it is a handy summary. The diagonals represent contradictions (application of ¬), and the columns give implications, notably, on the left, when *q* is proven then this implies that you wouldn’t prove *not-q* anymore. (Check the truth of the right column.)

- A table of the concepts of *contradiction* and *contrariness.* Note the implications in this scheme. The diagonal arrows run across the whole diagonals.

AffirmoNego[Ergo[p, q]]

Proven[q]				Refuted[q]
($p \vdash q$)	⟷	Contrary	⟷	($p \vdash \neg q$)
	↖		↗	
⇓		Not		⇓
	↙		↘	
(*p* ↧ ¬ *q*)	⟷	Subcontrary	⟷	(*p* ↧ *q*)
NonSequitur[¬q]				NonSequitur[q]

`Refuted[p___, q]`	gives the contrary Ergo[p, $\neg q$]
`AffirmoNego[Ergo[p___, q]]`	prints a table of contradiction and contrariness

Affirmo is Latin for "I affirm" and Nego is Latin for "I deny". To prevent a possible disappointment: Ergo, NonSequitur and Refuted are just objects and do not evaluate the validity of the argument.

2.3.3.5 Dynamics with predicates

(Ad 4) An example of inference in predicate logic is the following. It requires some notation. There are three different meanings of the verb "to be", some we just met, namely being an element (∈), being a sub-collection or subset (⊂) and being identical (=). For ∈ and ⊂ the order is important but = is a symmetrical relationship. There is also another kind of being, namely being *defined* identically (≡), which itself is asymmetrical but which implies =. Using these symbols we can now analyze the earlier statements.

English	Analysis in propositional logic	Analysis in predicate logic
Socrates is a man.	P[3]	Socrates ∈ Men
All men are mortal.	P[4]	Men ⊂ Mortals
Ergo	——————	Ergo
Socrates is mortal.	P[5]	Socrates ∈ Mortals

In this case, the inference creates a fifth expression, and we cannot judge the validity of the inference unless we analyse the sentences in detail and determine the relationship between the subject and the two predicates.

Ergo2D[P[3], P[4], P[5]]

1	P_3
2	P_4
Ergo	———
3	P_5

For the predicate calculus, the difference between statics and dynamics (statements versus inference) is the same as for propositional logic, only the internal rules differ.

2.3.3.6 Summing up

In summary, we find for inference as opposed to statics:

1. the deduction uses separate sentences
2. the truth of each sentence is asserted
3. the premisses contain sufficient information to warrant the conclusion

Logic as a research area consists both of the analysis of the structure of sentences (required for their interpretation) and the subsequent analysis of forms that cause valid or invalid reasoning (deduction).

PM 1. While "1 + 1 = 2" is necessarily true in arithmetic, logic does not study this statement since the structure of the statement has no relation to inference itself. But $p \Rightarrow q$ will be studied because of its relation to $p \vdash q$.

PM 2. Logic studies predicates that are relevant for inference. What do you think of "This inference is not valid." ?

2.3.4 Propositions (two-valued logic) and sentences (three-valued logic)

Propositions are special kinds of sentences. A proposition describes some state of the world and it is either true or false. Propositions are used in two-valued logic. A sentence can be any human utterance and its truthvalue might be indeterminate. For example, the exclamation "Boooh !" is neither true nor false. Sentences are used in three-valued logic. Rather than speaking about "sentential logic" we still maintain the general name "propositional logic".

The prerequisites of a rigorous analysis are not only of a technical mathematical nature but also involve the choice of adequate concepts and an insight in the methodology of science. The following subsections discuss this.

2.3.5 Truth

(a1) Our point of departure is Nature. Although Nature shows us only one face at the time, she changes her face over time: and this has taught us that a situation may have an opposite - i.e. that what once was the case need not be the case at another time. In this manner we reach the fundamental logical dichotomy that events *may* or *may not* occur. Having learned the idea of an opposite, our mind takes the freedom to imagine opposites where there is no physical possibility of an opposite. For example, we may imagine a situation where Earth has no gravity, even though that is physically impossible. In itself such hypothetical thoughts don't invalidate logic.

(a2) The fundamental aspect of (a1) is that it concerns the relation of the mind to reality. The wonders of the mind are many, and although researchers investigate those (see Rose (1978) but first Damasio (2003)), we are still far away from understanding how the mind manages to make pictures of reality, supposing that this is an adequate description of what the mind does. But this problem needs not concern us here. A second point is that there are some methods that allow the better acquisition of an understanding of reality than other methods do: the scientific method (supposing that "the" refers to a collection of methods). See for example Popper (1977:369).

(a3) Having the dichotomy of *being* and *not being* in reality and having the dichotomy of *imagining being* and *imagining not being* in the mind - and the latter is required for the mind when it is to be capable of imagining reality - then the following possibilities arise:

Mind (↓) versus reality (→)	Event A occurs	Event A doesn' t occur
Imagine event A	harmony	conflict
Imagine that event A doesn' t occur	conflict	harmony

PM. We don't discuss the possibility that the mind does not imagine anything - then it wouldn't be a mind at all (at least for all practical purposes).

(a4) We have the handicap of our method of communication: since we are using symbols we cannot point to events in reality, and neither can we show images in a mind. Therefor we introduce the phenomenon of *language* and we presume that there is a unique relation between the sentence in our language *and* the event *and* the image that belongs to that event. We will not discuss here how people learn their language but assume that it is by use of language that we are able to point to events and images.

(a5) We thus replace above table with one in which we use language. With the event in reality we associate the sentence in the language that describes the event. With the images in the mind we associate the fact that the mind *asserts* the sentence, or that the person utters the sentence. Let p = "Event A occurs".

Reality (↓) versus reality (→)	p	Not[p]
p is said	truth	falsehood
Not[p] is said	falsehood	truth

(a6) The situation of the table in (a5) is described by Aristotle (384-322 B.C.): "To say of what is that it is not, or of what is not that it is, is false, while to say of what is that it is, or of what is not that it is not, is true." (Tarski (1949:54)) It is actually not just that. Aristotle followed Socrates in the idea that the mind (the soul) does not lie to itself and stated: "(...) all syllogism [reasoning] (...) is addressed not to the spoken word, but to the discourse within the soul (...)" (DeLong (1971:23)).

(a7) The Socratic model of truth and falsehood of statement-thoughts in the soul is a good model but we might as well refer directly to nature. Instead of taking truth and falsehood as a description or qualification of a situation or an event, we will regard truth and falsehood as properties of sentences, i.e. that sentences can be true or false, when they refer to nature. This introduces noise, in that people in groups can lie when uttering sentences, but that is an aspect of pragmatics that we may neglect.

(a8) Attention must be given to the *assertoric use of language.* This is the convention to say only things that are true. When sentences are used (i.e. occur without quotation marks around them) then they are not only stated but it is understood that it is asserted that they are true. For example, this book is not just a bunch of sentences about which you must guess which I believe to be true and which not, but it is all asserted to be true.

(Though of course I might make errors, even if this very statement were the only one.)

(a9) This assertoric usage allows that people are silent, without the implication that they would be without a mind. When it rains I need not say that it rains, and when it doesn't rain then I need not say that it doesn't rain. We do not need to have people speaking all the time trying to express what they imagine about reality. More problematic are mistakes in assertion. I may have said that it rained, while closer inspection showed that it was the neighbour's kid spraying the window. We may distinguish an ideal logic where people don't make mistakes and a pragmatic logic where people make hypotheses and withdraw those upon refutation. From an economic point of view it is interesting to observe that some people or companies strategically manage the truthvalues of their statements.

(a10) With respect to this assertoric convention something special may be noted. When you would assert "*p* is true" then according to that convention you might as well just say *p*. And conversely. Similarly, "*p* is false" could be simply expressed as *not p*. And conversely. Thus the expression "*p* is true" is *equivalent* to *p*. Equivalence, or "if and only if", is expressed, as you may recall from Chapter 1, with the symbol ⇔. Let us distinguish a predicate TruthQ that tests on truth and the Definition of Truth that defines it.

- The definition of truth for an unevaluated variable or expression.

```
DefinitionOfTruth[p]
```

(TruthQ[p] ⇔ p)

- An example application of the definition of truth is:

```
DefinitionOfTruth["1 + 1 == 2"]
```

(TruthQ[1 + 1 == 2] ⇔ 1 + 1 == 2)

- In *Mathematica*, assertion can be modelled with the function ToExpression, that drops the quotes around a String.

```
% /. x_String :→ ToExpression[x]
```

True

- Application of the definition of truth to the Liar shows that we are fortunate that *Mathematica* has a check on a recursive depth, otherwise we would be locked in for eternity. The Liar paradox casts a doubt on our notion of truth.

```
DefinitionOfTruth["Liar"]
```

(TruthQ[Liar] ⇔ Liar)

```
% /. x_String :→ ToExpression[x]
```

$RecursionLimit::reclim : Recursion depth of 256 exceeded. More…

(Indeterminate ⇔ Not[TruthQ[Liar]])

DefinitionOfTruth[*p*]	generates TruthQ[*p*] ⇔ *p* for String *p*. If *p* is a Symbol then ToString[*p*] is used. Evaluate with *x*_String:→ToExpression[*x*]
TruthQ[*p*]	(1) True, False, Indeterminate if *p* has those values respectively, (2) TruthQ[*p*_String] gives If[ToExpression[*p*], True, False, Indeterminate]
Liar	Liar gives the liar sentence "This sentence is not true", formalized as "Not[TruthQ[Liar]]"

The standard *Mathematica* function TrueQ is different. TrueQ[x] gives True iff x === True, and it gives False otherwise.

DefinitionOfTruth uses $Equivalent instead of Equivalent, see §3.3.1.

- Note the following simpler expression of the Liar paradox. The earlier variant is more enlightening since it explicitly refers to the notion of truth. Yet, on structure, the Liar arises by using = opposite to ⇔ which is shown by this variant:

```
liar = ¬ liar
```

General::spell1 : Possible spelling error: new symbol name "liar" is similar to existing symbol "Liar". More…

$RecursionLimit::reclim : Recursion depth of 256 exceeded. More…

¬ Hold[¬ liar]

PM 1. In terms of electrical circuits, perhaps the Liar paradox can be represented by attaching a note to a switch "Turn the switch off to turn on the light" (using the basic diagram). The user will become confused, since, when he follows the instructions and turns the switch off in order to turn on the light, the reverse happens, since turning off the switch causes that the light is off. Such a note is confusing and normally would not be put there.

PM 2. Note that *Mathematica* is not so smart to identify the selfreference that is the cause of the recursion. We must be glad that the program is smart enough to notice that recursion occurs but there may be a dogma amongst mathematicians that issues of selfreference are not investigated, not even in "error handling". (Such tests at input cost time.)

(a11) A pragmatic theory of truth is dissatisfied with the idea that the Definition of Truth merely is dropping quotation marks. The pragmatic motivation may be formulated as, following Quine (1990:93): OK, if to call a sentence true is simply to assert it, then how can we tell whether to assert it ? The pragmatic question is valid, but for each sentence in particular we have to check with the appropriate field of science.

When the sentences concern questions on the validity of reasoning, then we check with logic. See Williams (2002) for a perhaps more pragmatic approach to truth and truthfulness.

(a12) Summarizing, we note three basic aspects of the notion of truth. First, the basic dichotomy of the occurring or not-occurring of events may also be expressed as the dichotomy of truth and falsehood. Secondly, there is the image in the mind and the correspondence between what is said (thought) and what is the case. This correspondence or lack of it can only be experienced by an intelligent mind and the notion of it is fundamental and inexplicable. A command like ToExpression in *Mathematica* mimics that notion. Thirdly, the notion of truth may be appealed upon in debate to give emphasis on what is or has been said, but in essence that only gives emotional or intellectual emphasis, since saying that something is true is equivalent to just saying it. A pragmatic reason for emphasis is that some people sometimes lie so that it might add value if you say that you speak the truth and all but the truth. In legal courts, an oath to state the truth and nothing but the truth is a useful reminder of legal consequences for perjury.

(a13) Apart from propositions that are either true or false, there appear to exist also sentences such as the Liar paradox that apparently are no propositions and that require some three-valued logic. It may be that *not* is a more fundamental notion than *truth*. Reality is just out there, but "not" may have more than one alternative to reality.

2.3.6 Sense and meaning

(b1) The meaning of a sentence is what it says.

(b2) The overall supposition of the former section was that sentences were sensical, i.e. propositions that represent a state of the world and that are true or false. Given the Liar sentence however we must account for sentences that are non-sensical - though they may still have meaning, since even the Liar sentence is not quite without meaning. The following table gives categories. What is not sensical is non-sensical and what is not meaningful is meaningless. We let $\mathbb{P}$ stand for all propositions (i.e. the first row, indicating 'the world') and $\mathbb{S}$ stand for all sentences (i.e. all rows). See Bochenski (1970:20), Frege (1949) and Ayer (1936).

Subject or object	*Example*	*Qualitifcation*	*Truthvalues*
propositions, judgements, theorems	Circles are round	sensical	True | False
phrases, contentions	The Liar	metaphysics – vague – paradoxical	Indeterminate
strings of symbols only	9 ui L 5 happy	meaningless	Indeterminate

PM. Below we will be a bit sloppy in using the words proposition, statement and sentence interchangeably. The prime cause is that a text reads ugly when using only one

term. Variety allows other brain cells to share the load. We presume that you know English and thus also know that the word "sentence" can also mean the verdict by a judge in court. But we also presume that you are not confused by that and think that logic only deals with legal cases. In the same way we presume that you understand above table, so that, once the distinction has been made, we can use the words again with some literary flexibility. Overall, this Part deals with two-valued logic anyway, so there should be little cause for confusion on what we are discussing. PM. In the same way we tend to use "when" a lot where other authors would write "if" to express the conditionality. But a book full of "if"s makes you feel iffy. "When" is more relaxed while you still get the idea.

(b3) The discussion on meaning and sense is a repetition of the discussion on sentences and propositions. Meaning is what a person thinks about something, or, alternatively what people intersubjectively consider. The totality of all meanings forms the memory of a person or group. A meaningless sentence will be a string of symbols for which there are no associations. (Though, the associations might be with those symbols.) Plato wanted meaning (or sense ?) to be some "idea" that we mortals never can know, and Aristotle required that a definition captured the "essence" of a thing without clarifying what an "essence" is (though perhaps an image in the mind / soul). Wittgenstein suggested that the meaning of a term is its use. We ourselves only get so far here that an intersubjective meaning of a sentence is the "complex" given by the individual meanings and the interaction between the individuals.

(b4) We use the meaning of a sentence to determine whether it has any sense.

(b5) Questions and imperatives have another linguistic form than indicative statements, for psychological reasons. Questions can be rendered in the form "It is not known by X whether ...." Or, "If Y knows it, it is hoped that Y says so." Imperatives can be given in the form "X has to or ought to ... ".

(b6) Authors like Ayer (1936) wrote a bit lengthy about declaring metaphysics and religion to be nonsense (or even meaningless). We can cut that discussion short by merely referring to the Liar and clarifying that we apparently require a truthvalue Indeterminate. But the Liar still relies on some meaning since we use that to find its structure.

(b7) Clarity and vagueness depend upon the context. Reduction of context-dependency might create sense - though full freedom of context doesn't seem possible.

(b8) The intimate link between logic and the methodology of science thus is given by the two-valuedness of sensical statements that are produced by science. Scientists must design their concepts such that they are two-valued, and in empirical testing they try to

mold their concepts such that this is achieved. Logic occurs in Nature, outside of us, and forces itself upon our mind. This differs from art and literature, where concepts and words might well be multidimensional.

(b9) With the modern overabundance of passwords for all kinds of internet applications, modern philosophy might become inclined to think that all meaningless strings of symbols might still have meaning, as some password or identification. However, the current focus is on sense.

2.3.7 Symbolics and formalism

(c1) We may take the liberal or formal position that it does not matter to logic why people assert sentences, as long as they assert some. In that case the notions of truth and falsehood become formal labels.

(c2) This liberalism is fed by the observation that people can think about the physical impossible. Though we took our basic point of departure in the dichotomy in Nature, it soon became clear that our minds allow a greater freedom in thought. A similar extension exists in the introduction of metaphysics. A question like "Must the government do something about poverty ?" does not present a physical problem but a moral one, but it can still be imagined that people look for an answer to it. People might want to settle the issue by majority vote, and in that respect the question can be considered to be sensical. It might become nonsensical if wider solutions are sought. But even then, one might hold that some people assert some morals, so that we could proceed with logic in a formal or hypothetical manner.

(c3) Though we can proceed as in (c2), it must be emphasized that a formal approach can never serve as a *foundation* for logic. In the formal approach there is nothing that forces us to accept the idea that statements are either true or false. It is only that such an idea has been ingrained in us by our experience with Nature that we apply it also in our beliefs, political views, and the like, and also in formalism.

(c4) In fact, if we are to give a description of logic then we can say: *logical theories are (successful) scientific theories about the structure of (other) successful scientific theories*. It is by science that we develop successful theories of Nature, and a theory can only be sucessful if it is free of contradiction. By studying the structure of these theories we grow conscious of what we are doing.

(c5) We can successfully predict (with a logical theory) that if a theory is inconsistent then it will not be successful. Namely, such an inconsistent theory will predict an event that will never occur, e.g. that it rains and doesn't rain at the same time. From this it follows that logical theories must be considered to be empirical theories. Empirics is not just a list of events but also their structure (that can be formalized).

2.3.8 Syntax, semantics and pragmatics

We have been using the terms syntax (symbolism or formalism), semantics, and pragmatics. It is useful to define them:

- **Syntax** (symbolism or formalism): investigates the structure in which symbols are used. For example "A is the father of B" is proper syntax.
- **Semantics**: investigates the meaning of objects, and the relationships of objects based upon their meaning. For example when A is the father of B then we know from the meaning of that assertion that B is the son of A.
- **Pragmatics**: investigates the relation of the subject who uses a language (formal or semantic system) to the environment. For example explains different semantics for different groups, or clarifies the differences between promissing, suggesting, threathening, convincing, etcetera.

Clearly, the various areas cannot function without the others. We cannot really judge whether "wizzy woolly wup" is a well-formed sentence when we don't know what the terms stand for. Conversely "A B father is of the" is alphabetically sorted so satisfies some syntax and we can imagine some semantics, yet it may be unclear whether A or B is the farther of the other. Unless there is a pragmatic convention that if B is the father then this is expressed in alphabetical order as "A B is of son the". Of course, pure applications of the various fields try to minimize the dependence of the other fields. The preferred approach is to optimize the combination of the various fields.

2.3.9 Axiomatics and other ways of proof

In their age-old civilisation, say 5000 years ago, the Egyptians developed a system for precise measurements. Complex constructions needed to be built, of which the pyramids were the largest ones. The positions of the stars needed to be tracked. And when the Nile had flooded again and had destroyed some lands and created some new ones, new lots had to be measured out for the displaced. When the Greeks came to visit, they noted this big body of geometric knowledge, and, perhaps not trusting all of it, they wondered: "Can you prove any of this ?" Eventually Euclid posed his axioms, and his textbook has been in use for a bit more than 2200 years now, see Struik (1977).

OK, a long and wonderful story has been simplified here in perhaps too mundane terms. The discoveries of the notion of proof and of the axiomatic method are key events in human history. It is impossible to do them justice in just a few lines. Perhaps we neither should look only to mathematics and look for the source in codes of law, with that notion of proof. Anyhow, the subject of proof will get more attention below. In logic, a main distinction is between (1) on one hand the axiomatic method that relies on

substituting expressions into expressions, and (2) on the other hand the pure enumeration and investigation of all possible cases, which enumeration implies some notion of arithmetic and combinatorics.

In all cases a proof requires an understanding intellect that is willing to see, understand and accept "Yes, this convinces me". Perhaps harder is the "No, this does not convince me" when the proof fails. Often the voice of authority forces people to accept all kinds of statements even though the proof is weak or non-existent. See Aronson (1992ab) on how peer pressure can get a person to say that three lines are equal that aren't.

2.3.10 Outline conclusions

Some early conclusions on the content and relevance of this book are:

- Inference occurs everywhere. Formal decision making might be a rare occasion, but another view is that inference occurs so often that we hardly notice it unless we see a need for a structured approach.
- Logical formats discipline us on the aspects involved in inference. We must decide on the statements that we accept, their structure, on what we want to know, whether our scheme of inference was correct, and whether we can explain the result to others.
- Often, the major result of such a process is that we start thinking about what we really want and what the alternatives could be. Often we already know the conclusion but just want to make sure that premisses are right. Rather than getting at inference we might discover that the situation is totally different than originally thought.
- The properties of the practical inference schemes are quite varied. Some use arithmetic and show the intermediate steps (truthtables), others just show the results (and use a mixture of arithmetic and substitutions), others are plain substitutions. Some verify statements, others allow the selection of "the news". And then there are the many routines of *Mathematica* that allow all kinds of inferences.
- With schemes on the computer, you can quickly run alternative schemes, and judge their properties. This will help you to determine what scheme suits your purposes.
- The results here are limited. This remains an Introduction into Elementary Logic. The field of elementary logic is wider, and then there is Advanced Logic, i.e. anything else not called Elementary.
- *Mathematica* is an excellent environment to discuss logic. It takes away the tedious computation, and it allows you to concentrate on the argument. It is another question whether it is a good environment for actual decision making. Basically *Mathematica* only *supports* inference. There will be occasions where *Mathematica* could be used but you might consider just talking to a wise person.

3. Propositional logic

3.1 Introduction

In this chapter we will define:

1. How to represent the propositions or sentences.
2. How to represent the propositional operators and their truthvalues.
3. How to represent the deductions and conclusions.

These basic concepts are covered in the `Logic`` package:

Economics[Logic, Print → False]

At a later stage we will also consider the `Inference`` package.

This chapter will develop two-valued propositional logic. The subsequent chapter will deal with the predicate calculus. Later we will regard three-valued propositional logic. While discussing propositional logic here it useful to keep these later developments in mind.

3.2 Sentences and propositions

3.2.1 Constants and variables

We will use *p, q, r,* ... as variables that denote sentences or propositions and *A, B, C, ...,* A_0, A_1, ... B_0, B_1, ... as such constants. A variable can temporarily become a constant when it gets an assigned value. An expression in a language, indicated by letters and other symbols put between quotes, is a constant too. Thus "If it rains then the streets are wet." is a sentence and a constant.

3.2.2 Englogish

A special language is "Englogish". This is the simple English-like language that some particular text routines defined below can deal with. Englogish statements still can have upper case letters and punctuation. These are transformed into 'proper sentences' (without first capital and final point), in order to allow those routines to recognize the

same expression in different places in a paragraph.

Sentences[*statement_String*]	transforms a statement into a List of Proper Sentences. A statement is a String that contains English sentences that start with a (Blank &) Capital Letter and that end with a Point (& Blank)

Subroutines not shown here are ProperSentence, JoinStatements and LocateThen. An example analysis is:

sents = Sentences["If it rains then the streets are wet. It rains."]

{ if it rains then the streets are wet, it rains}

3.2.3 Atomic sentences

A sentence is called "atomic" - from the viewpoint of propositional logic - if it doesn't contain propositional operators. The routine ToPropositionalLogic analyzes a paragraph into its atomic sentences. The structure of a statement can be clarified by the use of variables, while keeping track of what each variable means. The meaning of a sentence is what it says.

ToPropositionalLogic[*statement_String*]

transforms an Englogish statement into a List of *Mathematica* Logical Expressions. The routine also generates Propositions = {P[1], … } and PropositionMeaningRule

P	P[i] is an atomic sentence, the ith element in Propositions.
Propositions	the list of atomic sentences P[i]. Enter Propositions /. PropositionMeaningRule to substitute the various meanings
PropositionMeaningRule	This is a rule that gives the meaning of the current atomic sentences

- For example:

ToPropositionalLogic["If it rains then the streets are wet. It rains."]

{If[P_1, P_2], P_1}

Propositions

{P_1, P_2}

PropositionMeaningRule

{$P_1 \rightarrow$ it rains, $P_2 \rightarrow$ the streets are wet}

Propositional logic does not further analyze statements. If you go deeper in analyzing an atomic sentence then you start doing "predicate logic".

3.3 Propositional operators

3.3.1 Definition

The Logic` package extends the standard list in *Mathematica* with the following.

`$Equivalent[p, q]`	means $(p \Rightarrow q) \land (q \Rightarrow p)$, or p if and only if q (p iff q)
`Equivalence[p, ...]`	a conjunction of all possible equivalents, assuming these independent
`Unless[p, q]`	to be read as p ~Unless~ q and translated as $\neg q \Rightarrow p$
`Unless[p, q, r]`	$(\neg q \Rightarrow p) \land (q \Rightarrow r)$
`Imp`	replaces, Rule → Implies. Example: $p \to (p \to q)$ /. Imp The use of Imp allows the use of Rule (→) for logical input readability
`TertiumNonDatur[x]`	gives (x || Not[x])
`Negate[x]`	performs LogicalExpand[!x]
`NotNot[x]`	performs ! Negate[x] or ! LogicalExpand[!x]
`NotpOrq[p, q]`	gives Not[p] || q This function is used to replace If[p, q] and Implies[p, q]
`LogicalVariables[x]`	provides the list of variables for a logical statement x.
`Implications[x]`	gives a list of implications of x. If joined by And, then all implications are equivalent to the original

"Tertium non datur" is Latin for "There is no third possibility". Check the Unless by "I will have an apple unless you pay."

CounterImplies[p, q] = CounterImplies[$p \Rightarrow q$] = $(\neg q \Rightarrow \neg p)$.

Equivalent is new in *Mathematica* since version 7.0. The Logic` package uses $Equivalent to maintain consistency of the text originally written with 5.2, notably for 3-valued logic.

Equivalence is a key notion for doing logic. The following notions are relevant for understanding its functioning in this book. To start with, you might check that $(p \Rightarrow q)$ is equivalent to $\neg p \lor q$ by considering "The patient is operated today, or dies." A check using LogicalExpand does not work since $Equivalent is an operator in the Logic` package and it does not belong to the internal *Mathematica* system. However, the truthvalue shows that the equivalence always holds.

- LogicalExpand does not recognize $Equivalent.

$Equivalent[p ⇒ q, ¬ p ⋁ q] // LogicalExpand

$((p \Rightarrow q) \Leftrightarrow (\neg p \lor q))$

```
% // TruthValue
```

1

- When more statements are equivalent to each other then we can write without pain.

```
Equivalent[p, q, r, s]
```

$p \Leftrightarrow q \Leftrightarrow r \Leftrightarrow s$

- The construction of a truthtable of the latter is undefined, though, for where to put the brackets ? In fact, we should consider all possible combinations, since it might be that only one equivalence does not hold while the other do. The routine Equivalence states all possibilities.

```
Equivalence[p, q, r, s]
```

$((p \Leftrightarrow q) \bigwedge (p \Leftrightarrow r) \bigwedge (p \Leftrightarrow s) \bigwedge (q \Leftrightarrow r) \bigwedge (q \Leftrightarrow s) \bigwedge (r \Leftrightarrow s))$

- Now we can solve it.

```
% // ToAndOrNot // Simplify
```

$((p \bigwedge q \bigwedge r \bigwedge s) \bigvee (\neg p \bigwedge \neg q \bigwedge \neg r \bigwedge \neg s))$

Xor[*p*, *q*] is the negation of Equivalent when we consider just two variables. When more variables are involved, then Xor, as implemented in *Mathematica,* behaves in another way than the negation of Equivalent. For this reason there is no implementation for an easy transformation for Equivalent for more variables.

- This is not generally true or false.

```
Equivalent[Not[Equivalent[p, q, r]] , Xor[p, q, r] ] // TruthValue
```

$\frac{1}{2}$

3.3.2 Truthtables and truth value

Truth tables are an invention of Peirce 1902 and apparently independently Wittgenstein 1921. They enumerate all possible True | False combinations, and then the logical operators are defined in terms of their results. The truth value of the combined statement depends upon the truth values of the components. Truth values are True or False but for presentation they can be 1 or 0 when that is more compact. If we leave two-valued logic and start doing three-valued logic then we may use {True, False, Indeterminate} or {1, 0, $\frac{1}{2}$}.

`TruthValue[x]`	gives the truth share from 0 up to 1 (the average over all states of the world)
`TruthTable[x]`	gives the combinations of True and False for the variables in x. Output is a matrix if the option OutputForm → Table (default) otherwise a list of rules
`TruthTableRule[x:Propositions]`	If x is a list of (logical) variables then a Rule is created. If x is a statement then the statement is evaluated with that Rule
`TruthTableForm[x]`	uses TableForm for possibly more complex expressions
`SquareTruthTable[x]`	presents a binary problem in a square format

Options[TruthTable] also apply to Decide, TruthValue, TruthTableRule and LogicalVariables. Default option is Rule → Implies, meaning that '→' in input is read as 'implies'. Other values disable this option. The Rule format of TruthTableRule is different from TruthTable[x, OutputForm → Rule].

Apart from the square truthtable format and TruthTableForm in Chapter 1, there are these two formats:

- This is the most useful form when we are not interested in all detail and when the square is too compressed. We tend to use this most often.

TruthTable[p ⇒ q]

$$\begin{pmatrix} p & q & (p \Rightarrow q) \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{False} & \text{False} \\ \text{False} & \text{True} & \text{True} \\ \text{False} & \text{False} & \text{True} \end{pmatrix}$$

- This gives rules that can be used for substitutions.

TruthTable[p ⇒ q, OutputForm → Rule] // MatrixForm

$$\begin{pmatrix} \{p, q\} \to \text{True} \\ \{p, \neg q\} \to \text{False} \\ \{\neg p, q\} \to \text{True} \\ \{\neg p, \neg q\} \to \text{True} \end{pmatrix}$$

- If True = 1 and False = 0, and if all possible states of the world are equally likely, then the total truthvalue of the implication is 3 out of 4.

TruthValue[p ⇒ q]

$$\frac{3}{4}$$

PM 1. An expression that receives truthvalue 1 is called a *tautology*. An expression that receives truthvalue 0 is called a *contradiction*. A system of expressions that contains a contradiction (possibly derived) is called *inconsistent*, otherwise *consistent*.

PM 2. The truthfunction w is defined as $w : \mathbb{P} \to$ {True, False} or $w : \mathbb{P} \to \{1, 0\}$ and $W : \mathbb{S} \to$ {True, False , Indeterminate} or $W : \mathbb{S} \to \{1, 0, \frac{1}{2}\}$, with $\mathbb{P}$ the expressively complete set of

propositions and 𝕊 the sentences. Given their overlap we can write $W = w$. Some authors use the word "truthfunction" for other things and then it interferes with the a correct application of the word "function".

PM 3. With the truthfunction w we might define $(p \Rightarrow q) \Leftrightarrow (w(p) \leq w(q))$. Note that the inner equivalence would still be evaluated in terms of True and False, so that the use of 1 and 0 does not really eliminate the fundamental notion of a dichotomy.

3.3.3 Singulary operators

Since we are considering a singulary operator O : {True, False} → {True, False}, there are 4 possibilities.

BinaryTruthTables[1]

	1	2	3	4
p	False	Not	TruthQ	True
1	0	0	1	1
0	0	1	0	1

The operation to always give True and the operation to always give False may be too drastic to be much useful. These two operations thus don't qualify for a separate name of their own. Only "Not" gives a change and not too drastically so. Singular operator 3 that gives True iff p is True and that gives False iff p is False, has already been identified as TruthQ in the Definition of Truth.

3.3.4 Binary operations

When we are considering a binary operator O : {True, False}^{2} → {True, False}, there are 16 possibilities. They can be listed most easily with their truthvalues in columns and using 1 and 0. We don't need a name for all operations since they can be defined in terms of combinations of others. For ease of presentation we use in this table $p \Leftarrow q$ for $q \Rightarrow p$ and also $(p \neg\Rightarrow q)$ for $\neg(p \Rightarrow q)$ even while those are not properly defined and cannot really be used. Note the anti-symmetry down the middle: draw a vertical line between 8 amd 9, and see that 1 to 8 are Not 16 to 9.

BinaryTruthTables[2]

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
p	q	False	Nor	¬⇐	¬p	¬⇒	¬q	Xor	Nand	And	⇔	q	⇒	p	⇐	Or	True
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

The binary propositional operators g_i and their truthtables can be understood as follows, using arrows to denote the link from the domain (input) to the range (output).

$$\begin{array}{ccc} & g_i & \\ \mathbb{P}^2 & \longrightarrow & \mathbb{P} \\ u \downarrow & & \downarrow w \\ \mathrm{I}^2 & \longrightarrow & \mathrm{I} \\ & f_i & \end{array} \quad \text{using} \quad \mathrm{I} = \{1, 0\} \text{ and } u[p, q] = \{w[p], w[q]\}$$

Also $\vdash : \mathbb{P}^n \to \mathbb{P}$ has outcomes for $n = 2$, but then it still differs from $\Rightarrow$. The decision function differs from the operator, since the operator is merely a connector that increases the length of the statement, while the decision function can cause a shorter statement. Hence the table of 16 possibilities refers to operators and not to other possible functions.

PM. There is a school in logic that uses $\mathbb{P} = \mathrm{I}$ so that it considers only binary variables. Frege (1949), Church (1956) and Jeffrey (1967) belong to that school. Church (1956:25): "Therefor with Frege we postulate two abstract objects called truthvalues, one of them being truth (das Wahre) and the other one falsehood (das Falsche). And we declare all true sentences to denote the truthvalue true, and all false sentences to denote the truthvalue falsehood." But denoting means that sentences are in {1, 0} ! Wittgenstein (1921, 1976) Satz 5 says: "Der Satz is eine Wahrheitsfunktion der Elementarsätze." Dopp (1969:42) says: "sagen wir, die aus dem Aussagefunktor und einer Aussagevariablen bestehende Aussageform sei eine Wahrheitsfunction." Again a "sentencefunction" is misnamed as a "truthfunction".

L.E.J. Brouwer made a clear distinction between language and mathematics, and he considered logic the result of the mathematical study of language. Also he conceived that in other times and with another language (and different mathematicians) a different logic could have been created - but mathematics was always the same (even by definition). But above construction shows that these propositional operators have a necessary character, and would have been discovered at any time, though the form of the presentation of course might be arbitrary. It suffices to take the notion of dichotomy (pair of opposites) as primitive and then, with some combinatorics, derive the 16 possibilities. Since we assume the True | False dichtomy for Nature, the propositional operators are necessary and not some invention of language.

PM 1. Let $\mathbb{N}$ be the set of natural numbers. Note the even | uneven dichtomy. Let $n[x] = 1$ if x is even, and 0 when uneven. Then $x + 1$ is a negation, since $n[x + 1] = 1 - n[x]$. Then $x.y$ means a disjunction since $n[x.y] = 0$ only when both x and y are uneven. Similarly $x + y$ is equivalence $n[x + y] = 1$ iff $n[x] = n[y]$. There is a psychological difference with sentences since we have the assertoric convention to only say true things while with numbers we do not have the convention to say or count only even numbers.

PM 2. We already mentioned that Boole[True] gives 1, Boole[False] gives 0, and other values don't evaluate.

- Convert back by Boole[x] == 1 that is True, False or doesn't evaluate, or use Boole[x] === 1 that is True | False. What you use depends upon whether the x already has a determinate truth value or must still get one.

Boole[True]

1

Boole[whatever] == 1

Boole[whatever] = 1

Boole[whatever] === 1

False

3.3.5 A note on *not* - that you might not want to read

The Greek language has two words for "not": $\mu\eta$ (pronounced "mea" where "ea" sounds like in "hair") and $o\dot{\upsilon}$ (pronounced "oo"). "Choice negation ($o\dot{\upsilon}$) supposes a dilemma; the negation of one member of this dilemma is tantamount to the assertion of the other one. Exclusion negation ($\mu\eta$) does not suppose such a dilemma. Hence it is, unlike choice negation, not accessible to a positive interpretation." (Beth (1959:631)).

Consider the truthtable of the 16 binary operators again.

Suppose that someone claims $p \Rightarrow q$ (column 12) but we disagree and think $p \land q$ (column 9). How do we negate what the person says ? We cannot really say $\neg (p \Rightarrow q)$ (column 5) since this would contradict with our own view because $(\neg (p \Rightarrow q)) \land (p \land q)$ is clearly false. At the same time the view of $p \Rightarrow q$ (column 12) actually is not fully false since we know that $(p \land q) \Rightarrow (p \Rightarrow q)$ so that the people we are dealing with are not completely wrong. On the other hand, they might be merely polite, actually think $\neg p \land q$ even more opposite to our own views, but they use $(\neg p \land q) \Rightarrow (p \Rightarrow q)$ and then say $(p \Rightarrow q)$ to be agreeable to us. Apparently, our negation means something to the effect: "What you say is not adequate, not accurate, or not even relevant, though it is true in a certain respect".

These problems seem surmountable when we are precise in what we say. It helps to express whether we discuss single states of the world (rows in the table) or composite possibilities (columns). We can also use Inc[p, *reason*] = "p is incorrect for *reason*". Thus we can distinguish:

1. Say $\neg p$ iff it is meant that p is exactly false (thus from $p \Rightarrow q$ in column 12 to $\neg (p \Rightarrow q)$ in column 5)
2. Say $\hat{p}$ (*contrary*) iff p is an And statement and you think the contrary, or use $\tilde{p}$ (Other, the subcontraries) if that applies

3. Say Inc[p, *other column*] iff it is meant that specifically that other column applies
4. Say Inc[p, q] iff it is meant that a wholly other expression q needs to be considered. Thus the reply to "The world is flat" can be "I'd rather consider the question whether electrons are round".
5. Say NotAtAll[p] or †p iff it is meant that p is nonsensical and has truthvalue Indeterminate (this requires Chapter 7).

3.4 Transformations

3.4.1 Evaluation

In the default situation (of the *Mathematica* kernel) the logical operators do not evaluate. One has to call LogicalExpand or Simplify to achieve evaluation. It then appears that the basic philosophy is two-valuedness.

TertiumNonDatur[p]

$(p \lor \neg p)$

LogicalExpand[%]

True

3.4.2 Algebraic structure

The following routine exploits the isomorphism of {And, Or} with {Times, Plus}. Standard routines in *Mathematica* like Collect and Expand know how to handle Times and Plus and that property can be used for And and Or.

`ToLogic[`*f*`, `*input*`, `*parms*`]`	does *f*[*input* /. `rules`, *parms*] /. reversed`rules` (`rules` taken from Options)

The proper truthvalue of p ∨ q is 1 - (1 - p)(1 - q) but Logic uses the isomorphism of Or with Plus for functions such as Collect and Expand.

- In the following statement, it makes sense to "collect" all terms around q. However, the normal Collect does not work and hence we can use ToLogic.

p && q || ! p && q

$((p \land q) \lor (\neg p \land q))$

Collect[%, q]

$((p \land q) \lor (\neg p \land q))$

ToLogic[Collect, %%, q]

$(q \land (p \lor \neg p))$

The following transformations are more ambitious since they actually allow you to solve a proposition using Reduce. See Chapter 1 for an example. The following merely states what the routine does.

`ToAlgebra[x]`	turns x into equation by replacing And → Times, Or[p, q] → 1 − (1−p) (1−q), Not[p] → 1 − p. If the option All → True is set, then equalities p == 1 || p == 0 are included, so that the output can be offered to Reduce
`ToEquationRule[var_List]`	gives the rules for transforming propositions into equations
`FromEquationRule[var_List]`	gives the rules for transforming equations back into propositional logic

3.4.3 Disjunctive normal form

The disjunctive normal form is defined as: (1) it applies only to propositional variables and / or constants, (2) it contains only ∧, ∨, and ¬, (3) all conjunctions are at the lowest level and the disjunctions are at the highest level. The disjunctive normal form basically tabulates all states of the world in the truth table for which the expression is true.

`ToDNForm[expr]`	changes expr into the disjunctive normal form with only And, Or & Not
`ToAndOrNot[expr]`	only substitutes If, Implies and Equivalent by And, Or & Not

ToDNForm[(p ⇒ q) ⇒ q]

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

LogicalExpand[%]

$(p \vee q)$

- Another way to understand the DNF

Equivalent[(p ⇒ q) ⇒ q, ToDNForm[(p ⇒ q) ⇒ q]]

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

TruthValue[%]

1

- This function merely replaces the “implies” and thus differs from the DNF

ToAndOrNot[(p ⇒ q) ⇒ q]

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

3.4.4 Enhancement of And and Or

You may not like it that you have to evaluate a logical expression with LogicalExpand or Simplify to find a contradiction or truth. In that case you can "enhance" And and Or. This works only for those propositional connectives and not for the other ones. The routine Decide uses this enhancement.

- Examples are:

p ⋀ ¬ p

(p ⋀! p)

AndOrEnhance[True]

AndOrEnhance::State : Enhanced use of And & Or is set to be True

p ⋀ ¬ p

False

p ⋀ q ⋀ ¬ p ⋀ q

False

AndOrEnhance[False]

AndOrEnhance::State : Enhanced use of And & Or is set to be False

`AndOrEnhance[x]`	with x = True| On enhances && and || (otherwise off)
`AndOrEnhance[]`	gives the state of the system
`AndOrRules[]`	rules that enhance And & Or

The AndOrRules[ ] can be used for replacement in standard *Mathematica*. In AndOrEnhanced mode, they are added to the definitions of And & Or, and then are no longer available for Replacement. See also LogicState[ ].

3.5 Logical laws in propositional logic

3.5.1 Definition

A statement is a logical law in two-valued propositional logic iff the truthtable shows only the values True or the total truth value is 1. There isn't much more to it. The value of logic laws doesn't lie in their definition but rather lies in their use. The following may be added though, for proper perspective. The definition of a "logical law" includes a notion how one *proves* such a law. The following aspects reflect this:

- The sense of a "law" may primarily come from the *validity* of an inference or deduction (⊦ rather than ⇒)
- A truthtable proves the law by considering all states of the world (implying some notions of arithmetic)

- A proof can be given with the disjunctive normal form (where it is debatable whether this includes notions of arithmetic or enumeration too)
- A proof can be given with the axiomatic method that assumes some axioms as true and that deduces other laws.

We already have used truthtables in order to check upon tautologies. The following gives some more examples while using truthtables. Inference and the axiomatic method are discussed later.

- Nand[p, Nand[q, q]] is equivalent to $p \Rightarrow q$. In an axiomatic development we could use only Nand to define all other operators. In this case it is handy again that LogicalExpand does not recognize $Equivalent, otherwise it would just show True.

$Equivalent[Nand[p, Nand[q, q]], p ⇒ q] // LogicalExpand

$(p \bar{\wedge} (q \bar{\wedge} q) \Leftrightarrow (p \Rightarrow q))$

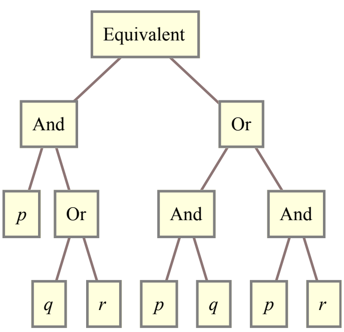
% // TruthTable

p	q	$(p \bar{\wedge} (q \bar{\wedge} q) \Leftrightarrow (p \Rightarrow q))$
True	True	True
True	False	True
False	True	True
False	False	True

- Expansion rules must be logical laws. And and Or satisfy laws of association and communication.

try = p && (q || r); try = Equivalent[try, LogicalExpand[try]]

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



p	q	r	$(p \wedge (q \vee r)) \Leftrightarrow ((p \wedge q) \vee (p \wedge r))$
T	T	T	T
T	T	F	T
T	F	T	T
T	F	F	T
F	T	T	T
F	T	F	T
F	F	T	T
F	F	F	T

3.5.2 Agreement and disagreement

One good way to use logical laws is the following. Suppose that John says P but Mary says Q. Normally Mary would be right of course but it is another question where John and Mary agree and disagree. Using the laws of logic we find that they agree when $P \Leftrightarrow Q$ and that they subsequently disagree for the other cases, thus Xor[P, Q]. Logic also

helps to simplify matters by reducing complex statements. It is not only good to know that Xor is the opposite of equivalence, but also to know what that opposite actually looks like in any particular instance.

Consider this life-threatening situation. John: "Give me the car keys or your parents will have a terrible accident" and Mary: "If my father cannot drive then, ok, I give you the car keys if my mom cannot drive too". When Mary hands John the car keys then their views are in agreement but if she gives them to her parents then they clearly have different views.

- These are the two positions:

```
ToPropositionalLogic["If not give me the car keys then
    your father cannot drive & dies and your mother cannot drive & dies."]
```

$\{\mathrm{If}[\neg P_1, (P_2 \wedge P_3)]\}$

```
John = %[[1]] /. If → Implies
```

$(\neg P_1 \Rightarrow (P_2 \wedge P_3))$

```
PropositionMeaningRule
```

$\{P_1 \to$ give me the car keys, $P_2 \to$ your father cannot drive & dies, $P_3 \to$ your mother cannot drive & dies$\}$

```
Mary = P[2] ⇒ (P[3] ⇒ P[1])
```

$(P_2 \Rightarrow (P_3 \Rightarrow P_1))$

- This solves the equivalence. The result for this example is simple so that we do not have to wonder what the Xor looks like. John and Mary now know what they should focus on.

```
John ~Equivalent~ Mary // ToAndOrNot // Simplify
```

P_1

- The Agreement routine collects these various steps and also reprints the input for a check on the brackets. Agreement selects the conditions under which the equivalence holds (and Disagreement those when it doesn't hold).

```
Agreement[John, Mary] // MatrixForm
```

$$\begin{pmatrix} 1 \to (P_1 \vee (P_2 \wedge P_3)) \\ 2 \to (P_2 \Rightarrow (P_3 \Rightarrow P_1)) \\ \text{Agreement} \to P_1 \\ \text{Disagreement} \to \neg P_1 \end{pmatrix}$$

The routine Agreement thus is not a bureaucratic procedure that merely selects the separate statements of agreement and disagreement. For example in Carl = $p \vee q \vee r$ and Monique = $\neg p \vee q \vee r$ you might clearly guess that they agree on $q \vee r$ and then jump to the conclusion that they disagree on the status of p. Well, this is how a marriage counselor might work. As logicians, we are interested in the conditions that make the

views agree and those that make them disagree.

- Actually, p drops out of the considerations. If Carl and Monique can cause $q \lor r$ to be fulfilled then they can live in agreement.

Agreement[p ⋁ q ⋁ r, ¬ p ⋁ q ⋁ r]

$\{1 \to (p \lor q \lor r),\ 2 \to (\neg p \lor q \lor r),\ \text{Agreement} \to (q \lor r),\ \text{Disagreement} \to (\neg q \land \neg r)\}$

- Alternatively put, if they don't fulfill that condition then there is no basis for agreement.

Agreement[p ⋁ q ⋁ r, ¬ p ⋀ ¬ (q ⋁ r)]

$\{1 \to (p \lor q \lor r),\ 2 \to (\neg p \land \neg (q \lor r)),\ \text{Agreement} \to \text{False},\ \text{Disagreement} \to \text{True}\}$

Agreement can print the result in DNForm with each Or element on a separate line. This suppresses all the other output but this format can be clearer for more complex statements.

Agreement[Matrix, (p ⋁ q ⇒ r) ⋁ s, (¬ p ⇒ s ⋀ (q ⋁ r))]

$$\begin{pmatrix} (p \land (r \lor s)) \\ (s \land (q \lor r)) \\ (q \land \neg p \land \neg r) \end{pmatrix}$$

`Agreement[`*p*`, `*q*`]`	determines where both statements agree, namely \$Equivalent[*p*, *q*], and where they disagree, namely Xor[*p*, *q*]. It uses Simplify instead of AndOrRules. PM. label Disagreement is a String
`Agreement[`*Matrix*`, `*p*`, `*q*`]`	just selects the Agreement and shows the result in MatrixDNForm

3.5.3 Methods to prove something

3.5.3.1 Introduction

Suppose that you have a contingent statement that is no logical law nor a contradiction but that you happen to know to be true, for whatever reason. You want to specify the cases that *make* it true, so that you can present an inference "under these assumptions this conclusion is true". For the following statement, for example, you can identify the rows in the truthtable that give truth.

try = p ⇒ (r ⋀ q)

$(p \Rightarrow (r \land q))$

```
tab = TruthTable[try]
```

$$\begin{pmatrix} p & q & r & (p \Rightarrow (r \wedge q)) \\ \text{True} & \text{True} & \text{True} & \text{True} \\ \text{True} & \text{True} & \text{False} & \text{False} \\ \text{True} & \text{False} & \text{True} & \text{False} \\ \text{True} & \text{False} & \text{False} & \text{False} \\ \text{False} & \text{True} & \text{True} & \text{True} \\ \text{False} & \text{True} & \text{False} & \text{True} \\ \text{False} & \text{False} & \text{True} & \text{True} \\ \text{False} & \text{False} & \text{False} & \text{True} \end{pmatrix}$$

You can easily select the rows that give True.

```
Select[tab, Last[#] === True &]
```

$$\begin{pmatrix} \text{True} & \text{True} & \text{True} & \text{True} \\ \text{False} & \text{True} & \text{True} & \text{True} \\ \text{False} & \text{True} & \text{False} & \text{True} \\ \text{False} & \text{False} & \text{True} & \text{True} \\ \text{False} & \text{False} & \text{False} & \text{True} \end{pmatrix}$$

Are there more systematic ways to prove something ?

3.5.3.2 Agreement

Agreement is also an adequate "method to prove something" (the former subsection). We need only establish the agreement with True. The routine also summarizes the result. (But we already knew this from the implies-form.)

```
Agreement[Matrix, try, True ]
```

$$\begin{pmatrix} (q \wedge r) \\ \neg p \end{pmatrix}$$

3.5.3.3 An algebraic manner

- The algebraic manner reproduces the rows in the truthtable.

```
eqs = ToAlgebra[try, All → True]
```

$\{q\,r\,p - p + 1 = 1,\ (p = 1 \vee p = 0),\ (q = 1 \vee q = 0),\ (r = 1 \vee r = 0)\}$

```
Reduce[eqs, {p, q, r}]
```

$$((p = 0 \wedge q = 0 \wedge r = 0) \vee (p = 0 \wedge q = 0 \wedge r = 1) \vee \\ (p = 0 \wedge q = 1 \wedge r = 0) \vee (p = 0 \wedge q = 1 \wedge r = 1) \vee (p = 1 \wedge q = 1 \wedge r = 1))$$

```
% /. FromEquationRule[{p, q, r}]
```

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

```
% // Simplify
```

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

3.5.3.4 LogicalExpand

try // LogicalExpand

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

There is, in other words, nothing new under the sun. All these approaches are essentially the same as the truthtable selection method. The Agreement routine even calls Simplify (LogicalExpand) so that they must be the same necessarily. The only difference in these methods is that their way of presentation differs: tables, lines, 1 or 0, a transformation of the input.

3.5.4 Contradiction, contrary and subcontrary

Logicians have an appetite for threesomes. The basic dichotomy of True | False tends to be less helpfull when there are more variables involved. Instead of {1, 2, 3, 4, ... } or {1, 2, many}, logicians rather concentrate on {1, 2, 3}. This gives a *trident*.

Thus, instead of focussing on the individual statements, let us now look at *combinations* of them, and in particular to *relations* between those combinations. To ease understanding, we temporarily use capitals to indicate combinations, thus $P = o[p, q, ...]$ for some logical operator o.

We have already seen contrariness in Chapter 2. There is another application now, a different kind. This kind of contrariness is only possible for And-statements and subcontrariness holds only for Or-statements.

- The down arrows represent implication. The other arrows mean that the labels apply.

Contrary[Table, p ∧ q] // Simplify

$$\begin{array}{ccccc}
(p \wedge q) & \longleftrightarrow & \text{Contrary} & \longleftrightarrow & (\neg p \wedge \neg q) \\
 & \nwarrow & & \nearrow & \\
\Downarrow & & \text{Not} & & \Downarrow \\
 & \swarrow & & \searrow & \\
(p \vee q) & \longleftrightarrow & \text{Subcontrary} & \longleftrightarrow & \neg (p \wedge q)
\end{array}$$

The table covers these two definitions:

- Two And statements P and $\hat{P}$ are called *contrary* when they exclude each other, while they might both be false.
- Two Or statements $\neg P$ and $\neg \hat{P}$ are called *subcontrary* when both might be true, while they cannot both be false.

Contrariness is an extreme negation of statement P. It can be denoted as $\hat{P}$. While Not[P] still leaves things to guess, $\hat{P}$ specifically targets what is the case that denies P.

The *contrary* of And-statement P is that statement $\hat{P} \neq \neg P$ such that $\hat{P} \Rightarrow \neg P$. From ($\hat{P} \Rightarrow \neg P$) it also follows ($P \Rightarrow \neg \hat{P}$). The negated statements $\neg P$ and $\neg \hat{P}$ are then called each

other's *subcontrary*.

Given P and $\hat{P}$ there is a remainder Other[p] that gives the rest, and that can be denoted as $\tilde{P}$. The *trident* arises by $P \vee \hat{P} \vee \tilde{P} \Leftrightarrow P \vee \neg P$.

- This gives a treesome or trident. It is equivalent to Tertium Non Datur.

Contrary[3, P]

$(P \vee \hat{P} \vee \tilde{P})$

- The trident shows its power when there are more statements involved. $P = p \wedge q \wedge$ r selects the first row, $\hat{P}$ selects the last row (extreme negation), while $\tilde{P}$ collects all neglected rows inbetween. The extremes give easy interpretable situations while the middle rows give a jumble of True | False.

TruthTable[p ∧ q ∧ r]

p	q	r	$(p \wedge q \wedge r)$
True	True	True	True
True	True	False	False
True	False	True	False
True	False	False	False
False	True	True	False
False	True	False	False
False	False	True	False
False	False	False	False

Contrary[Table, p ∧ q ∧ r] // Simplify

$$\begin{array}{ccccc}
(p \wedge q \wedge r) & \longleftrightarrow & \text{Contrary} & \longleftrightarrow & (\neg p \wedge \neg q \wedge \neg r) \\
 & \nwarrow & & \nearrow & \\
\Downarrow & & \text{Not} & & \Downarrow \\
 & \swarrow & & \searrow & \\
(p \vee q \vee r) & \longleftrightarrow & \text{Subcontrary} & \longleftrightarrow & \neg (p \wedge q \wedge r)
\end{array}$$

You may have seen the logical operator Nor. The following table probably explains it much better than a truthtable. It is the same as the table above but with the columns exchanged.

- This clarifies what the $\overline{\vee}$ operator stands for.

Contrary[Table, Nor[p, q, r]] // Simplify

$$\begin{array}{ccccc}
p \,\overline{\vee}\, q \,\overline{\vee}\, r & \longleftrightarrow & \text{Contrary} & \longleftrightarrow & (p \wedge q \wedge r) \\
 & \nwarrow & & \nearrow & \\
\Downarrow & & \text{Not} & & \Downarrow \\
 & \swarrow & & \searrow & \\
\neg (p \wedge q \wedge r) & \longleftrightarrow & \text{Subcontrary} & \longleftrightarrow & (p \vee q \vee r)
\end{array}$$

`Contrary[p]`	determines the q, such that p & q is False, while possibly Not[p] V Not[q]; just prints with the ^ (OverHat) if it does not evaluate
`Contrary[3, p]`	gives the three possibilities with Other[p] as the remainder
`Contrary[Matrix, p]`	gives {{p, Contrary[p]}, {!Contrary[p], !p}}
`Contrary[Table,` *p* (, *labels*)]	uses TableForm where arrows are included for contradiction and subcontrary; labels must fit the same matrix format
`Other[p]`	comes from the logical law Or[p, Contrary[p], Other[p]]

3.6 The axiomatic method

3.6.1 The system Ᵽ

By abstraction from reality we get a formal system, that differs from its (intended) interpretation in that no longer the semantics apply but only the syntax. The advantage of a formal system is that we are no longer distracted by hidden assumptions from our understanding of the problem area. All that is relevant to make something work is put in schemes that anyone can operate, even someone who does not understand the issue (like a computer). Let we take the subject of propositional logic with all its semantics as discussed above and let us try to create a formal axiomatic system for it. We then get an empty formal structure that we might interprete in various other ways too. In the axiomatic method we not only provide axioms and rules for deduction but we also must state a list of symbols and formation rules for which the axioms must hold. The relations just discussed are depicted in the following diagram. The situation actually is slightly more complex, since what isn't drawn is that we discuss these relations in a meta-language.

$$\text{(Intended) interpretation} \quad \underset{\textit{Interpretation}}{\overset{\textit{Abstraction}}{\rightleftarrows}} \quad \text{Formal system}$$

A full system for propositional logic can be taken from DeLong (1971:107). The "primitive base" Ᵽ is defined as follows. We use "**P**", "**Q**" and "**R**" as variables in our metalanguage to indicate constants, variables and formulas in the object language of the system. We assume that it is further obvious how the parentheses must be used

(DeLong's system includes those but then is less readable). Also, part 1 would actually require quotation marks, e.g. "⇒", but these have been deleted for readability too.

The primitive base Ᵽ ("primary logic" or "propositional calculus") for ℙ is:

1) List of symbols

- **a)** Two logical symbols: ¬, ⇒
- **b)** Two parentheses: (,)
- **c)** An infinite list of propositional constants, A, B, C, A_1, B_1, ...
- **d)** An infinite list of propositional variables, p, q, r, p_1, q_1, ...

2) Formation rules

- **a)** A constant standing alone is a formula
- **b)** A variable standing alone is a formula
- **c)** If **P** is a formula then so is ¬**P**
- **d)** If **P** and **Q** are formulas then so is **P** ⇒ **Q**

3) A list of initial formula schemata (axioms)

- **a)** **P** ⇒ (**Q** ⇒ **P**)
- **b)** (**P** ⇒ (**Q** ⇒ **R**)) ⇒ ((**P** ⇒ **Q**) ⇒ (**P** ⇒ **R**))
- **c)** (¬**P** ⇒ ¬**Q**) ⇒ (**Q** ⇒ **P**)

4) Transformation rule

- **a)** From **P** ⇒ **Q** and **P**, infer **Q**

This is it.

- You may check that the axioms are tautologies. If these would not be tautologies then the "⇒" in this system could not be interpreted as the "⇒" in our own language.

```
TruthValue /@ {P ⇒ (Q ⇒ P), (P ⇒ (Q ⇒ R)) ⇒ ((P ⇒ Q) ⇒ (P ⇒ R)), (¬ P ⇒ ¬ Q) ⇒ (Q ⇒ P)}
{1, 1, 1}
```

Now that we have created this formal system there naturally arise a number of questions such as whether it really is a "good" system. Axiomatic theory has developed a number of criteria to judge on that "goodness".

The traditional method to "prove" the adequacy of an axiomatic system is to provide an existing example in the real world that forms a model for the system. Since the world is assumed to be consistent (there is only one reality), a good fit would show that we have found a good formal model. It appears to be enlightening to analyze what we actually mean by "a good fit", since that generates all kinds of properties of systems that we may not have been aware of before.

DeLong (1971): "Our aim at formalization will be achieved if the informal theory presented above is an interpretation of the formal system." (p 106) and "The

propositional calculus Ꝑ is consistent, correct, independent, expressively and deductively complete, and decidable. It is not categorical, but may be made categorical if we so desire." (p141).

Those properties are meta-systemic that cannot be expressed in Ꝑ itself. They refer to a semantic concept (truth with respect to an intended application) and a formal concept (axiom and theorem proven within the system).

- **Correct**: Only truths are provable. *Proof:* The axioms are tautologies, the transformation rule preserves these, and tautologies are true. (PM. This still allows that some truths cannot be proven by the system.)
- **Consistency**: There is no p such that p and $\neg p$ can be derived. *Proof:* If Ꝑ would be inconsistent then everything can be proven. But next to the tautologies that are proven we find expressions like $p \Rightarrow q$ that are no tautologies and that hence are not proven.
- **Independent**: An axiom is independent if neither it nor its negation can be proven from the other axioms. *Proof:* If it is consistent then the negation cannot be proven, even with the help of the axiom itself. Hence it suffices to show that each axiom cannot be derived from the other axioms. This can be done by showing models such that the other axioms hold but not the one under target, for all targets. See DeLong (1971:135-136).
- **Expressively complete**: For propositional calculus this property means that all 16 binary operations can be expressed. *Proof:* DeLong (1971:137-138) explicitly translates all 16 columns in combinations of $\neg$ and $\Rightarrow$. It might be obvious already from the disjunctive normal form and the translation of And and Or.
- **Deductively complete**: All logical truths in the system (under the intended interpretation, here tautologies) are theorems. The system cannot be enlarged upon without requiring a change in the intended interpretation. *Proof:* All truths can be expressed in a disjunctive normal form. All theorems can be expressed in disjunctive normal form. Ꝑ can prove $p \vee \neg p$ (i.e. $p \Rightarrow p$) for a single variable. Suppose that truth A_k contains k variables or constants then it can be written as $A_{k-1} \wedge (q \vee \neg q)$ for some q. All the way down to the proof for the single variable.
- The following mimics the axiomatic procedure, starting with a truth.

(! p ⇒ (p ⇒ q)) // ToDNForm

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

ToLogic[Collect, %, {p, ¬ p}]

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

% /. ($q \lor \neg q$) → True

$(p \lor \neg p)$

- **Decidability**: There is a procedure do decide in a finite number of steps whether an arbitrary formula is a theorem. The procedure actually gives a proof. *Proof:* This is given by correctness and deductive completeness. The proof used there is constructive so that a proof can be generated when needed. (Compared to other systems where such a proof might be accepted but would require perhaps an infinite number of steps.)
- **Categorical**: All models of the system must be isomorphic. (1) Ᵽ appears to be non-categorical. *Proof:* It is possible to interprete the system in n-valued logic (though losing some of the other properties above). Another point is that there are contingent statements, i.e. of which the truthvalue might not be known. (2) Ᵽ can be made categoric. *Proof:* Use only constants True and False.

3.6.2 A system for ℙ

Having mentioned Ᵽ it is useful to directly develop a wider system for ℙ. A sufficiently rich system of two-valued propositional logic ℙ* contains ℙ, contains Ᵽ, plus the operators ¬, ∧, ∨, ⇒, ⇔ with their truthtables given above, the assignment (=) and identity relation (==) and the Definition of Truth. Next to the transformation rule of Ᵽ also the truthtable method is allowed. We also allow the "hypothetical mode" where a hypothesis is conjectured and can be retracted once a contradiction is arrived at; in that case the implication *hypothesis ⇒ contradiction* is accepted (or simply ¬ *hypothesis*).

- Hypothetical reasoning. Step 1 gives $p \Rightarrow q$.

Ergo2D[p ⇒ q, Hyp : p ∧ ¬ q, q ∧ ¬ q, Retract : p ∧ ¬ q, Not[p ∧ ¬ q]]

1	$(p \Rightarrow q)$
2	Hyp : $(p \land \neg q)$
3	$(q \land \neg q)$
4	Retract : $(p \land \neg q)$
Ergo	———————
5	$\neg (p \land \neg q)$

3.6.3 Information and inference

The axiomatic method differs from the truthtable method. The first uses only rules of substitution, expansion and contraction, that can be applied at liberty and that can deduce individual statements. The truthtable method follows a standard algorithm that specifically identifies tautologies. Nevertheless, there remains a (hidden) structural identity between these two methods, notably where the algorithm uses the same kinds of rules. There may be a difference though with respect to "finding new truths". It

appears to be instructive to compare here the axiomatic method with the truthtable method on the subject of the handling of information.

Note that a statement contains *more* information if its truthtable has *less* True's. A logical law carries no information since you would already know that it is always true. A contradiction carries the information that you should avoid it.

The transformation or reasoning procedure "From **P** $\Rightarrow$ **Q** and **P**, infer **Q**" can be compared to the truthtable of its projection. Consider the truthtable below, and the blocks in it, that each result into a truth. When we assert p then we take the first two blocks of the table. When we assert $p \Rightarrow q$ then we take the first, third and fourth block. When we join them with the "And" then only the first block remains. The other rows are false and no longer relevant. Thus our information can be summarized as $p \wedge q$ and is highly informative. As we now want to know the truthvalue of q, about which we have not asserted anyting yet, then, since $p \wedge q$, we find that q must be true (but this gives two rows again, which is less informative).

- The projection of *modus ponens*.

TruthTableForm[((p ⇒ q) ∧ p) ⇒ q]

Implies	And	Implies	p	q	p	q
True	True	True	True	True	True	True
True	False	False	True	False	True	False
True	False	True	False	True	False	True
True	False	True	False	False	False	False

With an inference $\{p_1, ..., p_n\} \vdash q$ there then are different types of concluding:

a. weakening: where the conclusion contains less information than the premisses

b. determining: where a single statement recieves a truthvalue which until then was not known

c. strengthening: where the conclusion contains exactly as much information as the premisses, so that the result is equivalent and no information is lost, but where the information might be restated perhaps in an easier form

The power of inference may also derive from that people get new information, this needs to be processed, and from the new information some tidbits are stored in permanent memory either as p or as $\neg p$. This process of information processing might

be more complex than purely the $S \vdash p$ format (so that in this case our lack comprehension is blocked by inadequate notation). We will return to the notion of information processing in Chapter 5.

3.6.4 Axiomatics versus deduction in general

Given the (hidden) structural identity of the axiomatic method and the method of the truthtables, it becomes a valid question why mentioning the axiomatic method at all. The point is that the axiomatic method still is the standard in mathematics for a proper definition of a system. Even the truthtable method may be analyzed as being based in axiomatics, as we based $\mathbb{P}^*$ upon Ꝑ.

That being said, this book takes a relaxed attitude towards axiomatics. It appears that the difference between the axiomatic method and a perhaps less formal but still deductive system becomes somewhat fuzzy. If we see the axiomatic method as merely substitution of truths in truths according to a truth-conserving rule then we are right to criticize this for neglecting solution strategies that reduce the time for a proof. Mathematical formalism as a goal in itself has little value as well. The objective of a proof is to convince a critical person and it may suffice that he or she recognizes the proof, as long as the method remains valid. In the methodology of science it appears that a surprising number of issues are not fully defined. Axiomatization of those issues seems overdone, though a bit more formalism sometimes helps. A useful deductive system, even not fully axiomized, still has the main properties of an axiomatic system, in that its terms and transformation rules must be defined somehow.

Hence, while Ꝑ was very formal, $\mathbb{P}^*$ was already less so, and in the remainder we might be even worse. Yet there will remain a deductive backbone and at the same time we will be using *Mathematica* as a logical evaluator.

4. Predicate logic

4.1 Introduction

4.1.1 Reasoning and the inner structure of statements

Predicate logic deals with inference such as: Socrates is a man, men are mortal, hence Socrates is mortal. It is useful to find formal notations for these relations since this allows us to become precise. If we don't formalize then we run the risk of imprecision and making wrong assumptions. The following is a good example how we might go wrong (DeLong (1971:238)):

1. Every dog is mortal.
2. Every animal is mortal.
3. Therefore every dog is an animal.

The premisses are true and the conclusion is. But is it also a good inference scheme ? Substitute "plant" for "dog" and you'll find the scheme invalid because the premisses are still true but the conclusion no longer is. But rather than trying all kinds of schemes and doing all kinds of substitutions, let us go for a structural analysis.

4.1.2 Order of the discussion

We will start with set theory and the diagrams originated by J. Venn since these provide the most elegant introduction into predicate logic.

Aristotle's syllogism already uses predicates but we better discuss it under inference, for which it originally was created. In this chapter we may determine its truthtable though.

Let us load the small application package and discuss the Venn diagrams. PM. The routines in this package are only meant to show the diagrams, it is no use to combine those in order to try for a graphical representation of a logical argument.

```
Economics[Logic`SetGraphics, Print → False]
```

4.2 Predicates and sets

4.2.1 Notation of set theory

"Socrates is a man, men are mortal, hence Socrates is mortal" can be analyzed in this way: Socrates is an *element* of the *set* of all men, that men are a *subset* of all mortal beings, and hence Socrates is an *element* of the *set* of all mortal beings.

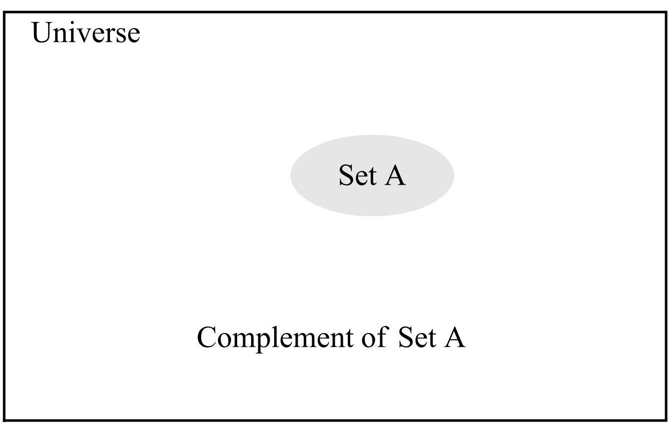
Other words for "set" are collection, aggregate, class. Sets may be denoted as *Men* = {Socrates, John, Charles, ...}. This justs lists the elements of the set. We can regard sets as *wholes*, so that we do not need to refer to the elements they have. Let A, B, C, ... be sets. The following concepts then apply:

1. The symbol $\mathbb{U}$ will be used for some *universal domain*
2. The symbol $\emptyset$ will be used for the *empty set*
3. The difference of sets A and B will be denoted by $A \setminus B$
4. The complement of A (all the points in the universe that do not belong to A) will be noted with a bar across it, thus $\overline{A} = \mathbb{U} \setminus A$

The Venn-diagrams assume a universe, and the sets contain elements of that universe. The sets may be drawn in a shade of gray but sometimes we just draw a boundary, and in other cases the sets may consist of areas and dots that are not connected.

- The basic diagram gives a model of set A and the complement of A. Note the isomorphism with $p \vee \text{Not}[p]$. Note that $\mathbb{U}$ and $\emptyset$ are complements too. (Apparently $\emptyset$ is in the drawing, try to locate it.)

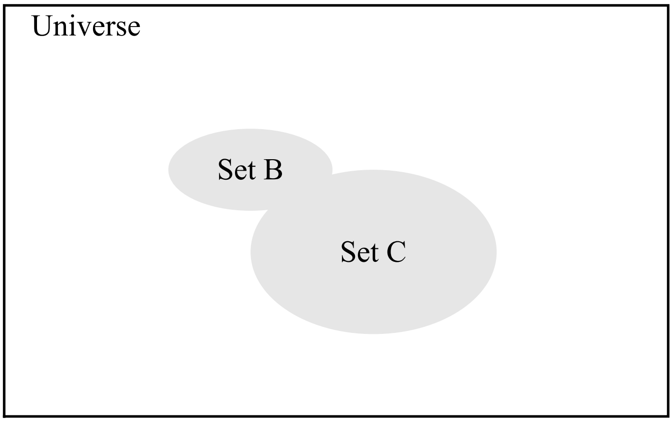
BasicSetGO["Set A"]



5. The union of sets A and B will be denoted by $A \cup B$

- By drawing B and C in the same shade of grey we emphasize that they are united now. Note the isomorphism with $p \vee q$. Note that if B and C wouldn't overlap they still could be united, but then would be disconnected.

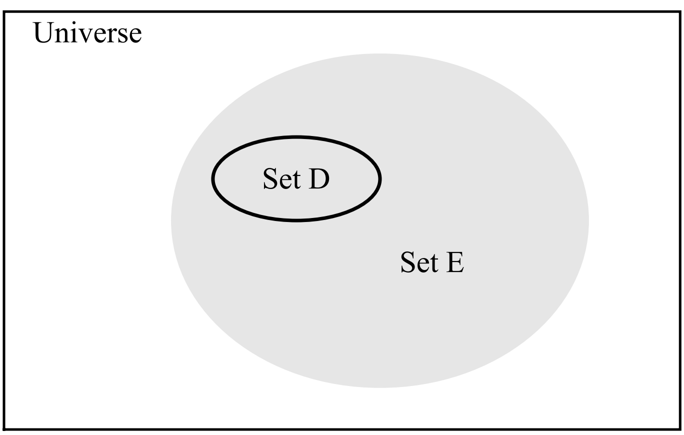
```
UnionGO["Set B", "Set C"]
```



6. A is a subset of B when all elements of A are contained in B, and then $A \subseteq B$
7. Sets A and B are identical iff they contain each other, $(A = B) \Leftrightarrow ((A \subseteq B) \wedge (B \subseteq A))$
8. A is a strict subset of B when all element of A are contained in B but $A \neq B$, and then $A \subset B$

- This shows $D \subseteq E$. We now draw a boundary around D otherwise we would not see the difference in the named areas. Note the isomorphism with $p \Rightarrow q$ and $x \leq y$.

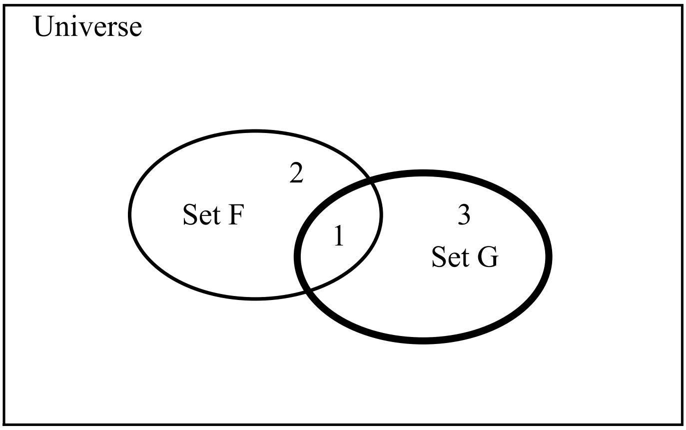
```
SubsetGO["Set D", "Set E"]
```



9. The intersection of sets A and B will be denoted by $A \cap B$
10. When A or B isn't a subset of the other, then $A \setminus B$ can be seen as $A \setminus (A \cap B)$

- This shows $F \cap G$. We now draw just boundaries. The intersection $F \cap G$ is given by area 1. Note the isomorphism with $p \land q$. $F \setminus G$ is given by area 2 and $G \setminus F$ is given by area 3. Using the identity relation we find for example $F = ((F \cap G) \cup (F \setminus G))$.

TwoSetsGO["Set F", "Set G"]



Note that the functions create True | False and that the operations create new sets. These concepts can be summarized in the following table for the notations (and how they can be used in *Mathematica*).

function	*full name*	*alias*	*operation*	*full name*	*alias*
$\subset$	`\[Subset]`	⁝sub⁝	$\bigcup$	`\[Union]`	⁝un⁝
$\supset$	`\[Superset]`	⁝sup⁝	$\bigcap$	`\[Intersection]`	⁝inter⁝
$\subseteq$	`\[SubsetEqual]`	⁝sub=⁝	$\emptyset$	`\[EmptySet]`	⁝es⁝
$\supseteq$	`\[SupersetEqual]`	⁝sup=⁝	$\overline{\text{Set}}$	OverBar[Set]	CTRL [7]
$\in$	`\[Element]`	⁝el⁝	\	\[RawBackSlash]	

- It is a convention to allow for any set X that $\emptyset \subseteq X$. If X is non-empty then we know for sure that $X \neq \emptyset$. Thus we are tempted to write for any set X that $\emptyset \subset X$. However, we cannot do the latter since we would get $\emptyset \subset \emptyset$ and we should at least have $\emptyset \subseteq \emptyset$.
- Check that $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Note that the table still contains the $\in$ symbol. As we regard sets as wholes it might seem that we do not require the notation for 'being an element'. However, it can be that some sets are elements of some other sets.

- One way to define the set of natural numbers is to associate or identify $0 = \emptyset$, $1 = \{\emptyset\}$, $2 = \{\{\emptyset\}\}$, $3 = \{\{\{\emptyset\}\}\}$, ...

- The *powerset* of A is the set of all its subsets, thus $\{x \mid x \subseteq A\}$. If A has n elements then the powerset has 2^n elements, due to the convention that $\emptyset \subseteq A$. For example, the powerset of $\emptyset$ is $\{\emptyset\}$, since only $\emptyset \subseteq \emptyset$. Hence that powerset has $2^0 = 1$ element.
- Russell's set paradox is the following. We can observe that many sets do not contain themselves as an element. Russell's example is the set of teaspoons that is not a teaspoon itself. Also $S = \{1, 2, 3\}$ does not contain itself, and luckily so, otherwise we would get an infinite regression $S = \{1, 2, 3, S\}$ if we were to substitute S in itself. It is natural to regard a set that does not contain itself as a *normal* set. Let R be the set of normal sets and thus we find the definition: $R \equiv \{y \mid y \notin y\}$. For example the empty set has no elements, thus $\emptyset \notin \emptyset$, and thus $\emptyset \in R$. This is in fact an existence proof for R, i.e. that $R \neq \emptyset$. But $R \in R$ gives $R \notin R$, gives $R \in R$ and so on. Contradiction ! Solution: Hence the definition of R cannot really be made. Thus there is no R such that $R = \{y \mid y \notin y\}$ or, alternatively, for any set x we hold that $x \neq \{y \mid y \notin y\}$. It is a bit surprising that such an important and long discussion in the history of philosophy and mathematics can be summarized in such a few lines. If it can. Note that saying $R = \emptyset$ is a standard way to say that the set does not exist. In the same way the set of all square circles is the empty set. But some hold that when we cannot define R then we neither can say that $R = \emptyset$, since the latter would presuppose that R is a *well-defined* concept. If R would be a well-defined concept then, again e.g. $\emptyset \notin \emptyset$, and the paradox resumes again. Thus, it is suggested, there are expressions for sets for which we cannot even say that such sets are empty and thus don't exist. We need another category, namely that such an R is *senseless.* In our terminology it still has meaning, since we understand the components set, element, etcetera; but precisely this meaning makes R senseless. We will return to the issue below.

4.2.2 Predicate calculus and set theory

Predicate logic may be equated to set theory. Opinions differ whether this can be really done. A common and useful effort at distinction is the following:

- Set theory is extensive, thus uses enumerations of the elements that belong to a set. A set is for example denoted by $S = \{x_1, ..., x_n\}$ and set membership is denoted by $x_1 \in S$.
- The predicate calculus is intensive, thus uses properties to determine whether an element belongs to a set. For property P it is denoted by $P[x]$ that some x has the property. Thus $P[x]$ is true for x iff x satisfies the property.

The distinction can be shown with the statement "Socrates is mortal". Set theory would create the set of all mortal beings by *assuming* a list of them all. This would include all bacteria from 5 billion years in the past and possibly some undiscovered aliens, though at some point the set theorist would stop and ask us whether the list was long enough.

Once the set has been determined then we can state "Socrates ∈ Mortals" to express that Socrates is *an element* of that set. In the predicate calculus we would *assume* that the notion of mortality is sufficiently well-known so that the issue can be judged on the properties of mortality and being Socrates. We directly write down *Mortal*[Socrates], where *Mortal*[x] is true iff x has the property of being mortal.

In practice these approaches often merge. For small or heterogeneous sets, the enumeration can be practical. The list {12, 23, 56, 528} does not have a unifying property so that we may be satisfied with the extensive method of merely considering all the listed elements. The Internal Revenue Service of the USA establishes a list of all persons who are supposed to pay taxes; they start of course by considering their properties, but also, given the complexity of these issues, by keeping track of the names and addresses once they are included. On the other hand, visitors to a cinema are admitted when showing the property of "having a ticket", and keepers at the gate normally don't wait with admittance till they have fully listed all elements.

The reason to make the distinction between the extensive and intensive method is not only for practical reasons but also because of philosophy. If we assume that people can make judgements on properties then it is a valid question where the knowledge about those properties comes from. Are these innate, so that babies have capacities beyond their own expectations, or can these be acquired, and, if so, how ? Philosophers have all kinds of ideas on that.

A key issue is the infinite. For infinite sets of numbers, points in geometry, and so on, mathematicians created methods to define sets. These sets use mathematical predicates, like "x is an even number". The idea is that such mathematical predicates would be somehow different from predicates like mortality. Such mathematical predicates can be *constructive* in that they build from small acceptable concepts to more complexity. On the other hand, mortality is awkward to check - or morally unacceptable to check.

From the point of logic these are interesting philosophical questions, but, we are focussed on valid reasoning and decision making. Hence we concentrate on structural forms. To simplify issues, we will allow for the situation that a set can be defined by a predicate, and conversely.

- We can create sets without fully enumerating them. Namely, a set can be *conditioned* by stating that its elements must satisfy a predicate. The condition is expressed with a bar as in {y | *condition* on y}, and this is pronounced as "the set of y given *condition*".

```
Mortal[x] ~$Equivalent~ (x ∈ {y | "y is Mortal"})
```

(Mortal(x) $\Leftrightarrow$ $x \in \{y \mid \text{y is Mortal}\}$)

- Conversely we can define a predicate on any particular set. E.g. define A = {12, 23, 56, 528} and then use:

BelongsToSet[A][x] ~$Equivalent~ (x ∈ A)

$(\text{BelongsToSet}(A)(x) \Leftrightarrow x \in A)$

Henceforth we will sometimes use predicates and sometimes use sets, assuming that the above translation works.

4.2.3 Universal and existential quantifiers

Properties - thus predicates or sets - can be satisfied by a number of objects. How many?

Over the ages, a philosophical and mathematical convention has grown to make a distinction between *all, some* and *none.* If we consider the hairs on Plato's head, then we tend to distinguish baldness, some hair, or a head of hairs. Over the ages there has not grown a tendency to draw the line at 1258 hairs or whatever other specific number. For mathematical problems, it is considered relevant whether a problem (i) is a tautology that applies to all, or (ii) has some solutions, or (iii) is a contradiction and thus has no solution. A driving example is that two lines either overlap (are the same), or cross, or are parallel, so that this problem has limited possibilities indeed. Proving that a problem is soluble (has at least some solutions) can also be called an *existence* proof. Another driving example is the infinite. We cannot know the infinity of all points on a line, but by some methods we still can determine whether all those points satisfy a criterion or not.

Using the negation Not, the threesome *all, some* and *none* can be reduced to a pair *all* and *some*. These two are called *quantifiers.* All will be called the *universal* quantifier, and Some the *existential* quantifier.

- The universal quantifier All is denoted as ∀ (in *Mathematica* ForAll or ESC fa ESC)
- The universal quantifier Some is denoted as ∃ (in *Mathematica* Exists or ESC ex ESC)

  PM. ∀ is the letter A turned upside down. Interestingly, the letter A originally was conceived in that very same way, namely as an image of a cow with its two horns. Similarly the letter B gave the plan of a house with two rooms.

Let us consider objects, subjects or elements x that may satisfy a property P. For example, the elements are the squared millimeters on Plato's head (say at his 50th birthday) while the property is 'being with hair' (having at least one hair). The following diagram has been used since medieval times:

$$\begin{array}{ccccc}
\text{All } x \text{ are } P & \longleftrightarrow & \textit{Contrary} & \longleftrightarrow & \text{All } x \text{ are } \neg P \\
\downarrow & \nwarrow & & \nearrow & \downarrow \\
\textit{Weakening} & & \textit{Contra} - & & \textit{Weakening} \\
 & & \textit{dictionary} & & \\
\downarrow & \swarrow & & \searrow & \downarrow \\
\text{Some } x \text{ are } P & \longleftrightarrow & \textit{Subcontrary} & \longleftrightarrow & \text{Some } x \text{ are } \neg P
\end{array}$$

Note that we have seen the same kind of table already in Chapter 2 and 3, but now there are the quantifiers All and Some. This is actually the original medieval application (with just the words All and Some, not the symbols).

- The table comes about by the four combinations of {All, ¬ All} and {*P*, ¬ *P*}. The human mind interpretes a double negation again in a positive manner, even when the negation doubles in different places.

{{all[x, P], all[x, ¬ P]}, {¬ all[x, ¬ P], ¬ all[x, P]}}

$$\begin{pmatrix} \text{all}(x,\ P) & \text{all}(x,\ \neg\ P) \\ \neg\ \text{all}(x,\ \neg\ P) & \neg\ \text{all}(x,\ P) \end{pmatrix}$$

The table shows that ∀ and ∃ share the nice property that they turn into each other by negation. Thus $\neg\ \forall x\ \ P[x] \Leftrightarrow \exists x\ \neg P[x]$ which may restated again in negative format as $\neg\ \exists\ x\ \ P[x] \Leftrightarrow\ \forall x\ \neg P[x]$.

Aristotle discussed these cases, though did not draw that diagram (DeLong (1971:16). The Scholastics used the memory aid of *affirmo* for the left column (Latin for "I affirm", with vowels A and I) and *nego* for the right column ("I deny", with vowel E and O), and then summarized the square into $\begin{matrix}\mathbf{A} & \mathbf{E} \\ \mathbf{I} & \mathbf{O}\end{matrix}$. The arrows indicate the following relationships.

- **A** ⟷ **O**: You have contradicted that there is a head of hairs iff you find some places that are bald.
- **E** ⟷ **I**: You have contradicted baldness ("All places are without hairs.") iff you find some hairs.
- **A** ⟷ **E**: A head of hairs is contrary to baldness. These are extreme opposites. If one is true then you know that the other is false. When you deny one then certainly the weakened opposite holds, and perhaps even the contrary other.
- **I** ⟷ **O**: "Some places have hair" is subcontrary to "Some places have no hairs". These can be complementary. Note though that *subcontrary* is not *contrary*. Some hairs do not exclude that there is a full head of hairs, unless it has been explicitly stated that this situation is excluded. Similarly, when it is stated that some places are bald, then baldness is still possible unless that has been excluded by finding some places with

hair. These issues are caused by the difference between *some* (⊆) and *only some* (⊂).

The following requires this package:

Economics[Logic`AIOE, Print → False]

With 4 corners and 2 truthvalues we get 8 rows that map "row ⇒ column". The following table lists them. The table gives the same information as already contained in the earlier scheme, but it is another presentation and it might help understanding. Note the symmetry between truth and falsehood, and the anti-symmetry across the subdiagonals. We also distinguish undetermined (U, contingent, True | False but not derivable from the input).

AffirmoNego[TruthTable]

Legend:

(1) True | False are input, and appear on de subdiagonals

(2) C[1] and C[0] are truth and falsehood by contradiction

(3) 1 and 0 are truth and falsehood by weakening

(4) U indicates contigent truth or falsehood (undetermined)

PM. A row input is False iff its contradiction is True

	$\begin{pmatrix} A & E \\ I & O \end{pmatrix}$	T | F	A	I	O	E
All x are P	A	True	True	1	c_0	0
Some x are P	I	True	U	True	U	c_0
Some x are not P	O	True	c_0	U	True	U
No x are P	E	True	0	c_0	1	True
All x are P	A	False	False	U	c_1	U
Some x are P	I	False	0	False	1	c_1
Some x are not P	O	False	c_1	1	False	0
No x are P	E	False	U	c_1	U	False

PM 1. The above (plus section 5.3) somehow sums up the history of logic from Aristotle to Boole, Frege and Peano. It is essentially contained in the writings of Aristotle. DeLong (1971:23): "All in all, Aristotle's logic is a magnificent achievement: he started with virtually no predecessors and invented a theory which today is considered in many respects right and even complete. If from today's vantage point it also seems limited, it must be remembered that the discovery of its limitations is a rather recent achievement, and that it was 2000 years before anyone beside the Stoics made substantial progress in formal logic. Aristotle by no means claimed that his syllogistic theory covered all kinds of arguments. He was aware of others. (....)" To understand this achievement, note: (1) Aristotle used *all*, *some* and *none,* and our symbols ∀, ∃ and ¬ are just fancy mnemonics for people too lazy to think for themselves, (2) Aristotle concentrated on inference, i.e. the

manipulation of combinations of above A, I, E, O statements, to arrive at conclusions. Modern set theory is a bit more subtle, with the difference between $\in$, $\subset$ and =. Yet Aristotle's scheme describes how people arrive at conclusions, and, if asked to the man, he probably would have known very well the difference between a single person and a group. (3) Aristotle also had to cut through the forest of linguistic conventions, as for example "All x are $\neg P$" is commonly expressed as "No x are P", and so on. Perhaps the antique Greek language was even worse. (4) Aristotle had to make a living too. He was the designated teacher to Alexander the Great, had to write some other books, had to run the Peripateic Academy, and so on.

PM 2. Start with a head of hairs. Pull out one hair. Is the head bald ? No ? Pull out another hair. Is the head bald ? No ? Thus, pulling out a single hair does not make a difference and does not cause baldness. Hence, continue this procedure and the person will still not become bald. This problem is called the *Sorites*.

PM 3. A modern approach is e.g. fuzzy logic. This drops the assumption of binary (yes | no) membership and allows for a gradual membership. An example is a family relationship where the family members look alike, but in different grades.

PM 4. Philosophers stopped trying to determine the exact number of hairs on Plato's head when this caused too much hair-splitting. They are not always so wise.

`AffirmoNego[Table]`	presents the medieval square
`AffirmoNego[Ergo[`S`, `p`]]`	creates the square for Ergo[S, p]
`AffirmoNego[f[`S`, `p`]]`	idem for f = Undecided, Unprovable, Undecidable, Consistent
`AffirmoNego[Truthtable]`	evaluate row ⇒ column of the medieval scheme. Execution of the routine gives a legend that explains all, suppress with Print → False
`AffirmoNego[ Truthtable, f[`S`, `p`]]`	e.g. for f = Ergo, Undecided, Unprovable, Undecidable, Consistent. Beware that consistency is not fully correct because of EFSQ

4.2.4 Relation to propositional logic

4.2.4.1 Sets and quantifiers

$P[x]$ not only expresses that x satisfies predicate P, but the whole expression is also a sentence so that it is subject to propositional logic. A predicate (say Big from "John is big") can be seen as a *propositional function* in that its application Big[x] represents a sentence "x is Big". Alternatively put, statements can be functions (and when asserted

then they are applied to reality).

A first consequence is that the quantifiers can be defined as conjunctions and disjunctions in propositional logic.

1. $(\forall x : p[x]) \equiv (p[1] \wedge p[2] \wedge p[3] \wedge ...)$ with some designated ordering
2. $(\exists x : p[x]) \equiv (p[1] \vee p[2] \vee p[3] \vee ...)$

NB. A variable x under a quantifier is called a "bound variable".

Next, what is the predicate itself ? It is that function. Functions are defined from a domain to a range, $f : D \to R$. The notion of the function is that mapping itself. We can use the symbol $Ab[expr]$ to indicate that we abstract from $expr$ and consider it with respect to the wholes of the domain and range, and the mapping involved. This holds by default for the universal quantifier, All. But we can do the same for the parts, Some.

1. Having a head of hair $\equiv Ab[\forall x : p[x]]$ with numbers of the milimeter grid, in this case finite.
2. Having a bald head $\equiv$ $Ab[\forall x : \neg p[x]]$
3. Having none of these $\equiv$ $Ab[\ (\exists x : p[x]) \wedge (\exists x : \neg p[x])\ ]$

Predicates can thus be linked to propositions on the items that satisfy them. Conjunction of predicates, like being big and red, can thus also be expressed in conjunctions of the constituents. Similarly for the disjunctions.

Also sets can be defined using the operators of propositional logic. One format is:

$\forall x$: $(x \in (A \bigcup B)) \Leftrightarrow (x \in A \ \ \vee \ \ x \in B)$

Using the $\{x \mid \ condition\ \}$ notation we need not mention the quantifier and only keep account of the condition. The internal statements are True | False but the result is a set (and thus not True | False):

- $A \bigcup B \ \equiv \ \{x \mid x \in A \ \ \vee \ \ x \in B\}$
- $A \bigcap B \ \equiv \ \{x \mid x \in A \ \ \wedge \ \ x \in B\}$
- $A \setminus B \ \equiv \ \{x \mid x \in A \ \ \wedge \ \ x \notin B\}$
- $\overline{A} \ \equiv \ \{x \mid x \in \mathbb{U} \ \ \wedge \ \ x \notin A\}$

Though the latter are sets, and not True | False, they still can be used in statements that are True | False, e.g. in $(A \bigcap B) \neq \emptyset$ which means that there is an element satisfying both sets (and the properties that define these).

4.2.4.2 Laws of substitution

The time-honoured method to study the structure of statements is to try different substitutions. The following example is taken from Quine (1981:71).

- A true statement is decomposed in a conjunction of two true statements.

"London is big and noisy" ⇔ ("London is big" ∧ "London is noisy");

Let us substitute “noisy” with “small”. The equivalence remains true. By trying various of such substitutions we find that this way of substitution is logically valid.

- A false statement is decomposed in a conjunction of two opposites, again false.

"London is big and small" ⇔ ("London is big" ∧ "London is small");

Let us now substitute “something” for “London”. The equivalence breaks down so that the substitution is invalid.

- A false statement is decomposed in a conjunction of two true statements.

"Something is big and small" ⇔ ("Something is big" ∧ "Something is small");

The last step shows that “something” is not a word just like “London”. Using the quantifiers and predicate logic we can rewrite the last expression and diagnose why the equivalence breaks down.

- The equivalence is still a falsehood but we now better understand that the conjunction runs over predicates and not over the propositions.

"∃ x: (Big[x] ∧ Small[x])" ⇔ ("∃ x: Big[x]" ∧ "∃ x: Small[x]");

Our conclusion is that Big and Small are two non-empty sets, but there is no element in their intersection.

Let us substitute “noisy” back.

- Only the implication is true in general, but because of London the intersection of Big and Noisy is not empty.

"∃ x: (Big[x] ∧ Noisy[x])" ⇔ ("∃ x: Big[x]" ∧ "∃ x: Noisy[x]");

The subsequent conclusion is that the latter equivalence is not a logical law, since we have found a counterexample is Big and Small. The relation may hold contingently for some predicates but may not hold for others.

PM 1. We took this example from Quine (1981). He also remarks: “Those who state that mathematics in general is reducible to logic are counting ‘∈’ in the vocabulary of logic and thus reckoning set theory to logic. This tendency has been encouraged by a confusion of the $F[x]$ of logic with the $x \in A$ of set theory. Properly considered, F is not a quantifiable variable referring to a set or attribute or anything of itself. The importance of this contrast between the schematic predicate F and the quantifable set variable A is overwhelming, once we stop to consider what quantification over sets wrought.” (p125, notation adapted). Comment: (1) Logic is about inference and mathematics about all other formalism, so that “reduction” is a non-issue. (2) Predicates *can* refer to themselves, like *Lying* = *Lying*[¬

Lying] which is lying that you are not lying. This is not a vacuous definition since there is an predicate that satisfies this meta-predicate, $\exists\ x : Lying[x]$, namely *Lying* or ¬*Lying* itself. (3) We hold predicates and sets as equivalent. See §4.6.3 for paradoxes in set theory.

- This quits after 256 substitutions. (*Mathematica* checks recursion not selfreference.)

```
Lying[x_] := Lying[¬ Lying][x]

Lying[Lying];
```

$RecursionLimit::reclim : Recursion depth of 256 exceeded. More…

$RecursionLimit::reclim : Recursion depth of 256 exceeded. More…

PM 2. Quine (1981:124): "A celebrated theorem of Gödel says that no proof procedure can encompass all the truths of number theory, to the exclusion of falsehoods. Since we can express number theory in set theory, it follows that there is no hope of a complete system of set theory." This is a *non sequitur* since if a weak system (number theory a.k.a. arithmetic) is encompassed in a stronger system (set theory) then the stronger system definitively can add features. See Chapter 9.

4.2.5 Review of all notations

We have an abundance of notations for the same issues. Before we construct the table of all the equivalent notations we may observe:

- A key observation of Jaakko Hintikka is that a quantifier always requires a domain. Above we have been sloppy and simply stated "All x are P" while we should have identified some domain D and then have said "All x in D are P". A convention is to take $D = \mathbb{U}$ so that $x \in \mathbb{U}$ becomes a bit *overdone* (since everything is in the universal set) so that people tend not to mention $\mathbb{U}$. However, this is not quite a simple issue and hence it is better to include D. As variables are bound by a quantifier, the quantifier is bound by its domain. Also an expression like $x \in D$ can be called an "open" statement and needs a quantifier or a substitution by a constant for closure.
- When we want to account for the domain then we may get a mixture between predicates and set theory. For predicates we may choose between writing $\forall\ x \in D$: $P[x]$ and $\forall\ x$: $D[x] \Rightarrow P[x]$. If it is clear that $D = \mathbb{U}$ then we might as well write $\forall\ x$: $P[x]$. But if $D \neq \mathbb{U}$ then the implicative format is not quite conventional but still preferable.
- One advantage of introducing a domain is that we can now consider the *contrapositions,* from "All D are P" to "Every non-P is non-D" where we switch domains. This is a new set of questions that logicians and medieval monks delved into. However, using set theory, they are implied by our standard formulas and henceforth do not require special attention.
- If D and P are empty then all elements of D are vacuously also elements of P, so that $D \subseteq P$. From *All Ds are Ps* you would like to conclude that also *Some Ds are Ps*, i.e. the

weakening from **A** to **I** in the medieval square. However, the latter is formalized as $(D \cap P) \neq \emptyset$ implying that the intersection of the empty set with itself is non-empty.

Hence the following True | False statements hold for non-empty domain D:

Language	*Medieval*	*Predicates*	*Quantifiers and sets*	*Sets as wholes*
All x in D are P	**A**	$\forall\ x : D[x] \Rightarrow P[x]$	$\forall\ x \in D : x \in P$	$D \subseteq \mathrm{P}$
Some x in D are P	**I**	$\exists\ x : D[x] \Rightarrow P[x]$	$\exists\ x \in D : x \in P$	$(D \cap P) \neq \emptyset$
Some x in D are $\neg P$	**O**	$\exists\ x : D[x] \Rightarrow \neg P[x]$	$\exists\ x \in D : x \notin P$	$(D \cap \overline{P}) \neq \emptyset$
All x in D are $\neg P$	**E**	$\forall\ x : D[x] \Rightarrow \neg P[x]$	$\forall\ x \in D : x \notin P$	$D \subseteq \overline{P}$

Note that these are statements and not sets. The statement $(A \cap B) \neq \emptyset$ is also a rather ugly and circumstantial expression for something that is rather simple. It is kind of strange that set theory in the last hundred years has not come up with a straightforward symbol to express that two sets overlap. However, it is an useful option now to extend the use of And and Or to sets. Generally we will not become confused between statements and sets, and we already have a precedence in the application of the propositical operators to predicates. Hence we adopt the following definitions:

- $A \wedge B \;\Leftrightarrow\; (\mathrm{A} \cap \mathrm{B}) \neq \emptyset$ is the statement that two sets overlap
- $A \vee B \;\Leftrightarrow\; (\mathrm{A} \cup \mathrm{B}) = \mathbb{U}$ is the statement that two sets form the universe (while not necessarily being opposities)
- $\neg (A \wedge B) \;\Leftrightarrow\; (\overline{A} \vee \overline{B})$ *Proof*: Note that $\overline{A \cap B} = \overline{A} \cup \overline{B}$ and thus $(\mathrm{A} \cap \mathrm{B}) = (\overline{\overline{A} \cup \overline{B}})$. Hence $(\neg (A \wedge B)) \Leftrightarrow ((\mathrm{A} \cap \mathrm{B}) = \emptyset) \Leftrightarrow (\overline{\overline{A} \cup \overline{B}} = \emptyset) \Leftrightarrow ((\mathbb{U} \setminus (\overline{A} \cup \overline{B})) = \emptyset) \Leftrightarrow ((\overline{A} \cup \overline{B}) = \mathbb{U}) \Leftrightarrow (\overline{A} \vee \overline{B})$
- $\neg (A \vee B) \;\Leftrightarrow\; (\overline{A} \wedge \overline{B})$ *Proof:* substitute the complements in the former relation and then use $(\neg p \Leftrightarrow q) \Leftrightarrow (p \Leftrightarrow \neg q)$

Recall that we write $A = B$ when those sets *fully* overlap.

We should beware of existence issues. When two sets overlap then there really should exist an element that they share. When we know that $A \subseteq B$ then it is tempting to conclude $A \wedge B$, but given the convention that $\emptyset \subseteq B$ too we would get $\emptyset \wedge B$, which is a contradiction. Hence the proper formulation is:

- $(A \subseteq B \;\wedge\; A \neq \emptyset) \;\Rightarrow\; A \wedge B$

Let us use $A \mathrel{\overline{\sqsubset}} B \;\Leftrightarrow\; (A \subseteq B \;\wedge\; A \neq \emptyset)$ to express that nonempty sets are involved. This notation is not conventional though and it would be clearest to simply state that the sets are non-empty.

- We defined for any set A that $\emptyset \subseteq A$, which we might translate into $\forall\ x : x \in \emptyset \Rightarrow x \in A$. From weakening we get $\exists\ x : x \in \emptyset \Rightarrow x \in A$. This seems like a strong claim,

asserting the existence of something in the empty set. However, suppose that this would hold for some constant a_0. Then $a_0 \notin \emptyset$ so that the implication holds by falsehood. Hence this a_0 is an example for which the relation holds *ex vacuoso*.

- $\emptyset \subseteq A$ might also be translated as $\forall\, x \in \emptyset$: $x \in A$, so that we select our domain $D = \emptyset$. Weakening results into $\exists\, x \in \emptyset$: $x \in A$, which again seems like a strong claim on the existence of someting in the empty set. When the domain is $\emptyset$ then actually A will be empty too so that we get $\exists\, x \in \emptyset$: $x \in \emptyset$, seeming like a stronger existence claim. Again the relation is only hypothetical. Thus all non-existent things have all properties - due to *ex falso sequitur quodlibet* (EFSQ, see below).
- Note that it seems easier to claim $\neg \exists\, x : P[x]$ than proving $\forall\, x : \neg P[x]$. Good debaters are more inclined to state "I haven't seen a green horse yet" than "I've checked all horses and they are not green" since the latter is a strong claim that likely is false and then undercuts the argument. The pragmatic difference here is between seeing what comes your way and fully checking out. In logic, we don't quite make that difference. That is, from the definitions above.

4.3 Axiomatic developments

DeLong (1971, chapter 3) also presents a "primitive base" for a first order predicate calculus, discusses axioms for set theory by Fraenkel (though Zermelo-Fraenkel is more common) and indicates higher order predicate theory.

For our purposes it is not necessary to repeat these possibilities. Who is interested can find them in the literature, if not DeLong's book itself. For us it suffices (a) that we have seen a "primitive base" for propositional logic, so that we know how such a thing looks like, (b) that we know that similar steps are possible for the predicates and sets. For us, it is more important to continue with logic.

For reference, it remains useful to define the full system of predicate logic as $\mathbb{P}^{**} = \mathbb{P}^{*}$ plus the notions of set theory and the predicate calculus both on the rules of formation of syntaxically correct statements and on valid inference schemes - that we will discuss in next chapter.

Two points remain to note. First, DeLong draws the line between logic and mathematics at the first order predicates. Higher predicates would be mathematics. Such a distinction seems less useful and it would suffice that logic is interested in the subject of inference and thus considers any predicate in that field, of whatever order.

The second point is that an example of a higher order predicate is instructive. We take the example of DeLong (1971:124-5) of the identity (=). We have been using identity but it is good to have a definition. The concept of identity is based upon an equivalence, that

can be decomposed into two implications:

- Two things are identical to each other if and only if they share all properties: $\forall x\ \forall y\ ((x = y) \Leftrightarrow (\forall P\ (P[x] \Leftrightarrow P[y]))$
- The principle of indiscernibility of identicals: If two things are identical to each other then they share all properties: $\forall x\ \forall y\ ((x = y) \Rightarrow (\forall P\ (P[x] \Leftrightarrow P[y]))$
- The principle of identity of indiscernibles: If two things share all properties then they are identical: $\forall x\ \forall y\ ((x = y) \Leftarrow (\forall P\ (P[x] \Leftrightarrow P[y]))$.

The founder of modern symbolic logic Frege wondered about the equality of the Morning Star and the Evening Star. They appear at different moments in the sky whence their names. But astronomy shows that it is just the planet Venus. If you consider it an important property how an object in the sky appears to us on the surface of the planet, then you might hold that the Morning Star is not equal to the Evening Star. To prevent discussions with astronomers you might agree that Venus is identical to itself with the property of having different apparant positions.

4.4 Predicate environment in *Mathematica*

4.4.1 Notations

Since we use *Mathematica*, a discussion of some notations will be useful to understand specific statements in this book.

See *Mathematica* Section 1.8 for a longer discussion of the List object as it has been implemented there. A List is an ordered object. If you want an alphabetical order then use Sort or Union.

- A list can be anything, e.g. numbers, symbols, strings, expressions, other lists, ... The empty list is {} while ∅ is merely the theoretical symbol for the empty set. Thus {{}, 1} contains both {} and 1.

```
lis = {0, b, John^2, "Amsterdam will be flooded in 2060", {}, ∅, {{}, 1}}
```

{0, *b*, John^2, Amsterdam will be flooded in 2060, {}, ∅, {{}, 1}}

- The following definition causes an infinite evaluation and *Mathematica* again saves us from waiting till eternity.

```
selfreferent = {1, 2, 3, selfreferent};
```

$RecursionLimit::reclim : Recursion depth of 256 exceeded. More…

- This creates the list of the integers till 20. To make it a bit more interesting, we square the even numbers and take the square root of the uneven numbers.

```
Table[If[EvenQ[i], i^2, Sqrt[i]], {i, 1, 20}]
```

$\{1, 4, \sqrt{3}, 16, \sqrt{5}, 36, \sqrt{7}, 64, 3, 100, \sqrt{11}, 144, \sqrt{13}, 196, \sqrt{15}, 256, \sqrt{17}, 324, \sqrt{19}, 400\}$

- *Mathematica* also protects us from other infinite procedures.

```
Table[i, {i, 1, Infinity}]
```

Table::iterb : Iterator {i, 1, ∞} does not have appropriate bounds. More…

Table::iterb : Iterator {i, 1, ∞} does not have appropriate bounds. More…

Table[*i*, {*i*, 1, ∞}]

- The procedure Join just links up the lists even when the same elements occur. This is useful when the elements are not merely elements but represent some score or code for which the position and / or the frequency of occurrence is important.

```
Join[{1, 2, 3}, {3, 2, 1}]
```

{1, 2, 3, 3, 2, 1}

Mathematica also has functions that search and test for elements of lists. See *Mathematica* Section 2.3 for rules on selection and pattern recognition.

- Membership is True | False. With respect to the *lis* defined above:

```
MemberQ[lis, ∅]
```

True

- Existence can be tested with ¬FreeQ. FreeQ[*lis,* _Integer] gives True iff *lis* is free of integers. If *lis* is free of them, then there doesn't exist an integer in *lis.* There is at least one integer in *lis* iff FreeQ gives False.

```
¬ FreeQ[lis, _Integer]
```

True

- Whether *lis* is empty can be tested by Length or by a direct test.

```
{Length[lis] === 0, lis === {}}
```

{False, False}

`{`e_1`, `e_2`,…}`	an ordered list of the elements; can also be entered as List[e_1, e_2, …]
`Table[`*expr*`, {`*i*`, `*imax*`}]`	make a list of the values of *expr* with *i* running from `1` to *imax*
`Join[`$list_1$`, `$list_2$`, … ]`	concatenate lists
`MemberQ[`*list*`, `*form*`]`	tests whether *form* is an element of *list*
`FreeQ[`*list*`, `*form*`]`	tests whether *form* doesn't occur in *list.* FreeQ[*expr, form, levelspec*] tests only those parts of *expr* on levels specified by *levelspec*
`Length[`*list*`]`	counts how many elements occur in *list*
`Position[`*list*`, `*form*`]`	gives the positions at which *form* occurs in *list*
`Cases[`*list*`, `*form*`]`	gives the elements of *list* that match *form*
`Select[`*expr*`, `*f*`]`	selects the elements in *expr* for which the function *f* gives `True`
`Count[`*list*`, `*form*`]`	gives the number of times *form* appears as an element of *list*

Definition of the List object, creation, testing and searching for elements of lists.

Mathematica contains some set theoretical functions that apply to lists. They do not necessarily apply to other input forms but perhaps you could extend the routines yourself. The procedures are not fully set theoretic since *Mathematica* adds some features. For instance the universe may be a good theoretical concept but for practical purposes it must be stated.

- Union not only eliminates double occurrences but also sorts the result. Union considers the lists as a set so that the order does not matter, whence it is easier to list the elements aphabetically so that it reads easier.

Union[{2, 3, 1}, {c, 3, b, 2, a, 1}, {b, c, a}, {11, 10, 3}]

{1, 2, 3, 10, 11, *a*, *b*, *c*}

- A set with 3 elements has a powerset with 2^3= 8 elements.

Subsets[{A, B, C}]

{{}, {*A*}, {*B*}, {*C*}, {*A*, *B*}, {*A*, *C*}, {*B*, *C*}, {*A*, *B*, *C*}}

`Union[`$list_1$`, `$list_2$`, … ]`	sorts to a unique list of the distinct elements in the $list_i$
`Intersection[`$list_1$`, `$list_2$`, … ]`	lists the elements that occur in all the $list_i$
`Complement[`*universal*`, `$list_1$`, … ]`	lists the elements that are in *universal,* but not in any of the $list_i$
`Subsets[`*list*`]`	lists the powerset, all subsets of *list*

Set theoretical functions.

The Economics Pack adds some functions. ExistQ differs from FreeQ in: (1) positive thinking rather than negative thinking as in ¬FreeQ, (2) it uses True | False tests rather than pattern recognition so that the mind remains in the testing mode, (3) the default is the first level and not all levels.

- Compare ExistQ and FreeQ.

```
ExistQ[{a, b, c, {1}}, IntegerQ]
```

False

```
¬ FreeQ[{a, b, c, {1}}, _Integer]
```

True

- OrderedUnion is a nice invention of Carl Woll to unite sets but keeping the same order. PM. The routine JoinNews uses Complement to identify the news, but Complement sorts that news.

```
OrderedUnion[{2, 3, 1}, {c, 3, b, 2, a, 1}, {b, c, a}, {11, 10, 3}]
```

{2, 3, 1, *c*, *b*, *a*, 11, 10}

```
JoinNews[{2, 3, 1}, {c, 3, b, 2, a, 1}, {b, c, a}, {11, 10, 3}]
```

{2, 3, 1, *a*, *b*, *c*, 10, 11}

AllQ[*list*, *crit*, *levelspec_*: 1]	gives True if *list* is a nonempty List and all its elements cause crit to be True. The routine uses MapLevel. Levelspec (default 1) and options are for Level
ExistQ[*list*, *crit*, *levelspec_*: 1]	gives True if *list* is a nonempty List and some of its elements cause crit to be True. The routine uses MapLevel. Levelspec (default 1) and options are for Level. Use levelspec Infinity when crit is to apply to any level
OrderedUnion[$list_1$, $list_2$, …]	gives the union of the $list_i$ without sorting as Union would do; no option for SameTest; algorithm of Carl Woll
ForList[*list*, *e*, *var_List*, *body*]	evaluates body for all elements in *list* using formal element *e*. For ex

Some additions in *The Economics Pack*. See ShowPrivate for the specific implementations.

Application of functions or predicates to the elements of lists can be done in the following manner.

- For the operation $\forall\, x \in A : f[x]$, *Mathematica* has the procedure Map[*f*, *A*] and output is the list $\{f[e_1], f[e_2], ...\}$ for the elements in *A*. This can also be denoted as *f* /@ *A*. When *f* is a test then the result is a list of True | False values.

```
NumberQ /@ lis
```

{True, False, False, False, False, False, False}

- ForList has a format that sometimes is more agreeable to set theoretic notation, and that at the same time allows more complicated functions that use named internal variables. If you would use the internal function more often then of course it would

be advisable to separately define it and use Map directly. But for single applications this might work well.

```
ForList[lis, x, {y, z}, y = If[StringQ[x], 100, x]; z = If[ListQ[y], 200, y]]
```

$\{0, b, \text{John}^2, 100, 200, \emptyset, 200\}$

4.4.2 Quantifiers

Mathematica generally supports the standard syntax of mathematical logic. The *Mathematica* environment forces to specific choices though. Notably, variables that appear in the quantifiers ∀ and ∃ must appear as subscripts otherwise there could be a conflict with multiplication. When constructing formulas it seems generally easier to use textual input and let *Mathematica* do all the layout.

- This states that property *P* holds for all pairs that satisfy condition C.

```
ForAll[{x1, x2}, C[x1, x2], P[x1, x2]]
```

$\forall_{\{x1,x2\},C[x1,x2]}\, P(x1, x2)$

- If the properties are not depending upon *x* then we get a mere implication.

```
ForAll[x, C, P]
```

$(C \Rightarrow P)$

Textual input	*formal input*	*meaning*
`ForAll[`*x*`,` *expr*`]`	$\forall_x expr$	*expr* holds for all values of *x*
`ForAll[{`x_1`,` x_2`, … },` *expr*`]`	$\forall_{\{x_1,x_2,\ldots\}} expr$	*expr* holds for all values of all the x_i
`ForAll[`*x*`,` *cond*`,` *expr*`]`	$\forall_{x,cond}\, expr$	*expr* holds for all *x* satisfying *cond*
`ForAll[{`x_1`,` x_2`, … },` *cond*`,` *expr*`]`	$\forall_{\{x_1,x_2,\ldots\},cond}\, expr$	*expr* holds for all x_i satisfying *cond*
`Exists[`*x*`,` *expr*`]`	$\exists_x expr$	there exists a value of *x* for which *expr* holds
`Exists[{`x_1`,` x_2`, … },` *expr*`]`	$\exists_{\{x_1,x_2,\ldots\}} expr$	there exist values of the x_i for which *expr*holds
`Exists[`*x*`,` *cond*`,` *expr*`]`	$\exists_{x,cond}\, expr$	there exist values of *x* satisfying *cond* for which *expr* holds
`Exists[{`x_1`, … },` *cond*`,` *expr*`]`	$\exists_{\{x_1,x_2,\ldots\},cond}\, expr$	there exist values of the x_i satisfying *cond* for which *expr* holds

Two forms for quantifiers in *Mathematica*. NB. AllQ and ExistQ in *The Economics Pack* concern only Lists.

The *Mathematica* guide warns that "in most cases" the quantifiers will not immediately evaluate. But they can be simplified or submitted to routines `Reduce` or `Resolve`.

For more on this, see *Mathematica* Section 3.4.11.

- This asserts that all x squares are positive. This certainly would hold for all negative numbers too. The expression however does not evaluate yet.

```
ForAll[x, x^2 > 0]
```

$\forall_x x^2 > 0$

- The expression gives `False` since the inequality fails when `x` is zero.

```
FullSimplify[%]
```

False

- Resolve can eliminate quantifiers from truth statements. Is there an assignment to the variables that makes the expression true ?

```
Exists[{p, q}, p || (q && ! q)]
```

$\exists_{\{p,q\}} (p \vee (q \wedge \neg q))$

```
Resolve[%, Booleans]
```

True

- Note that *Mathematica* kindly resolves questions on formatting for us. In the following expression the outer x might seem to range over the x in the inner $\exists_x$ yet the convention is that a bound variable looks for its closest bound.

```
Exists[x, x^2 > 0 && Exists[x, x^2 == 0]]
```

$\exists_x \left(x^2 > 0 \bigwedge \exists_x x^2 = 0\right)$

```
FullSimplify[%]
```

True

4.4.3 Propositional form

We want *Mathematica* to be able to recognize a propositional form even when quantifiers occur. This however requires some ingenuity since such constructs can be complex (see the former section).

A statement like "being wet is a predicate, x is a variable, and for all x, x satisfies this predicate" is somewhat dubious, since x occurs both freely and as a bound variable, while "predicate" has different roles. Simply using replacement in *Mathematica* might cause havoc. We still want to be able to manipulate a statement like that, if only to express this dubiousness. But in advance it is not clear what kinds of predicates we will use. A pragmatic solution is to keep a list of predicates that we can test for, and, if we use a new one, then we add that to the list.

- This example is a dubious statement that has small p occurring all over. The routine Statements knows about ForAll and Exists and thus can find the variables for propositional logic.

example = p ⋀ ForAll[p, P[p]] ⋁ q ⇒ ¬ ForAll[x, P[x]]

$(((p \wedge \forall_p P_p) \vee q) \Rightarrow \exists_x (\neg P_x))$

Statements[example]

$\{p, q, \exists_x (\neg P_x), \forall_p P_p\}$

- The propositional form can become clearer by making substitutions with formal parameters (while keeping the ones that do not cause problems).

PropositionalForm[example]

$(((p \wedge P_2) \vee q) \Rightarrow P_1)$

- The PropositionMeaningRule in this case only lists the terms that are hard to recognize.

PropositionMeaningRule

$\{P_1 \to \exists_x (\neg P_x), P_2 \to \forall_p P_p\}$

`Statements[x]`	gives the list of variables in x assuming the possibility of predicates. When Show → False then just formal P[i] are shown, see PropositionMeaningRule
`PropositionalForm[x]`	gives the form of the proposition where all predicates known in PredicatesQ[List] are replaced with formal parameters P[i]. The routine sets Propositions and the PropositionMeaningRule, see also Results[PropositionalForm]. The routine uses Add in Options[Predicates] to enlarge on the list
`PredicatesQ[x]`	returns True when at least one of the `PredicatesQ[List]` occurs in x, at any level (but as a head of an expression and not just as a symbol). The positions are stored in Results[PredicatesQ]. It uses Add in Options[PredicatesQ] to enlarge on the list
`PredicatesQ[List]`	contains a list of predicates to test for in Statements

Thanks to these routines, we can make useful truthtables. Note that TruthTable is not so smart to recognize when ∀ and ∃ might be contradicting.

- This is a simpler example (for *example* we would get many useless rows).

```
TruthTable[ForAll[p, P[p]] && ¬ ForAll[x, P[x]]]
```

$$\begin{pmatrix} \exists_x (\neg P_x) & \forall_p P_p & (\forall_p P_p \wedge \exists_x (\neg P_x)) \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{False} & \text{False} \\ \text{False} & \text{True} & \text{False} \\ \text{False} & \text{False} & \text{False} \end{pmatrix}$$

4.4.4 Element and NotElement in *Mathematica*

When $\neg (a \in b)$ is evaluated in *Mathematica* then it uses a separate symbol $\notin$ that LogicalExpand does not recognize. Just great.

```
Not[a ∈ b] ⋀ (a ∈ b)
```

$(a \notin b \wedge a \in b)$

```
% // LogicalExpand
```

$(a \in b \wedge a \notin b)$

A solution is to transform the expression into something that is recognized. *Belongs* is a symbol in *The Economics Pack* that also shows as $\in$ but that can be recognized by LogicalExpand. You can see the difference with 'true' $\in$ from the brackets and a bit more space around it.

- The functions ToBelongs and FromBelongs do the transformations.

```
ToBelongs[%]
```

$((a \in b) \wedge (a \notin b))$

```
% // LogicalExpand
```

False

- NoteElement puts a shell around a statement, first replacing $\in$, then evaluating, and replacing them back. This is useful when there remain set relational statements that you want to continue working with.

```
NoteElement[(a ∉ c) ⇒ (a ∉ b) ⋀ (a ∈ b) // LogicalExpand]
```

$a \in c$

PM. Another strange aspect in *Mathematica* is: "Element[{x1, x2, ... }, dom] asserts that all the xi are elements of dom." But when we use { } to denote set membership then the set {x1, x2, ... } would be the element of dom, not the elements of {x1, x2, ... }. *Mathematica* is not perfect.

`NoteElement[expr]`	first replaces Element with Belongs, and NonElement with ! Belongs, then evaluates *expr*, then substitutes back. For example NoteElement[$p \in P \wedge p \notin P$ // LogicalExpand] will now evaluate to False
`Belongs[x, y]`	like Element, so that Not[Belongs[x, y]] can be recognized by LogicalExpand. In TraditionalForm it prints exactly the same
`ToBelongs[expr]`	replaces Element with Belongs, and NotElement with Not[Belongs[…]]
`FromBelongs[expr]`	replaces Belongs with Element

4.5 Theorem that the Liar has no truthvalue

4.5.1 The theorem and its proof

Theorem on the Liar: In a sufficiently rich system of two-valued logic (see above for ℙ**) there does not exist a Liar. Seen from that system the Liar has no truthvalue (True or False).

Proof: The proof has the following structure. First we pose the hypothesis that a Liar exists. Then we derive the contradiction, causing us to retract the hypothesis. That part of the deduction is maintained as an implication, however. Subsequently, we use that implication to derive the main result.

Let us *suppose* that there is a L saying that it is not true. Then the following deduction can be made.

Ergo2D["L = Not[TruthQ[L]]", $L \Rightarrow \neg L$, $\neg L \Rightarrow L$, $L \wedge \neg L$]

1	L = Not[TruthQ[L]]
2	$(L \Rightarrow \neg L)$
3	$(\neg L \Rightarrow L)$
Ergo	———————
4	$(L \wedge \neg L)$

- Check this part of the deduction

$(L \Rightarrow \neg L) \wedge (\neg L \Rightarrow L) \Rightarrow (L \wedge \neg L)$ // TruthValue

1

The first hypothesis causes a contradiction so that we retract the hypothesis, while keeping the implication of the step.

- Hence "L = Not[TruthQ[L]]" ⇒ (L ∧ Not[L]). However L ∨ ¬L.

Ergo2D["L = Not[TruthQ[L]]" ⇒ (L ∧ Not[L]), L || Not[L], "L ≠ Not[TruthQ[L]]"]

1	(L = Not[TruthQ[L]] ⇒ ($L \wedge \neg L$))
2	($L \vee \neg L$)
Ergo	————
3	L ≠ Not[TruthQ[L]]

- Check this part of the deduction

((p ⇒ q) ∧ ¬ q) ⇒ ¬ p // TruthValue

1

Overlooking the whole, we find that we have reached a definite point that proves the theorem.

- Final conclusion

Ergo["L ≠ Not[TruthQ[L]]"]

(⊢ L ≠ Not[TruthQ[L]])

Q.E.D.

PM. This proof seems to use the counterfactual "Suppose there is ...". ℙ** allows for counterfactual reasoning and only allows the substitution of truths into truths, since ℙ* does. However, above proof still works since L can be used in the sense of All L so that the assumption holds for all variables. For ℙ** it indeed is immaterial whether we write p or L for the Liar construction. The proof thus also shows that for all p no such construction can be given. That being said, it still is useful to state the proof in terms of "Suppose there is ..." since that reads better for psychological reasons (while it also is valid for systems that accept hypothetical reasoning).

By *reductio ad absurdum* it thus follows that there is no sentence in ℙ that satisfies the "definition" of the Liar. In equivalent words:

(1) There is no p in ℙ such that p = Not[TruthQ[p]]

(2) For all p in ℙ it holds that p ≠ Not[TruthQ[p]]

In the methodology of science, Nature is the (intended) interpretation of ℙ and we can observe that the interpretation still stands, since we do not observe that Nature forms 'statements' such as the Liar.

The only problem that we have now is to properly interprete this result. We took ℙ as a collection such that we felt that any sentence could be constructed but the *Theorem on the Liar* can be read as saying "do not form the Liar". It must be admitted that if the theorem is added to the formation rules of ℙ then this is a smart way to prevent the occurrence of the Liar outright instead of having to find this out by means of deduction. However, we

feel frustrated by this since we can form a Liar in our natural language and we had hoped that $\mathbb{P}$ would cover our language.

We conclude that natural language used for human communication is some $\mathbb{S}$ larger than $\mathbb{P}$. Below we will develop three-valued logic to allow for sentences like the Liar - with an Indeterminate truthvalue. Thus we have $\mathbb{S}$ with normal syntax for English and we let logic determine whether a sentence is in $\mathbb{P}$ or $\mathbb{S} \setminus \mathbb{P}$, without relying on "formation rules" for $\mathbb{P}$.

Corollary: Three-valued logic is required for natural languages. *Proof:* The Liar is a sentence that cannot be formed with True | False and since we want to be able to form it, it must have a third value. Q.E.D.

Hypothesis: For language a three-valued logic is also sufficient.

Seen positively, the Liar provides an existence proof for three-valued logic. It is like the introduction of the 0, where people originally didn't know what finger to use for it. There is no logical distinction between restricting a two-valued logic to $\mathbb{P}$ and the introduction of a third value to $\mathbb{S}$ (i.e. when $\mathbb{S} \neq \mathbb{P}$).

We will say NotAtAll[p] or †p iff it is meant that p is nonsensical and has truthvalue Indeterminate.

```
NotAtAll[Liar]
```

```
† Not[TruthQ[Liar]]
```

PM 1. † is called "dagger" by some (indeed in the official *Mathematica* list of symbols) and by others it is used in publications as a "cross" to indicate a person's decease. There is thus a crime or a natural death involved but anyhow with †p the ghost of p lingers on.

PM 2. We don't need †($L = \neg L$) since we already have ($L \neq \neg L$). Thus the † only applies to an L if it were defined such.

PM 3. † is a useful general notion. Consider the statement "All round squares are round." Though this is nonsensical you might hold that it is syntactically correct using "All X Y are X". If you are right then of course also "All X Y are Y" whence "All round squares are square". Thus "All round squares are round and all round squares are not round": contradiction ! A solution might be to say in the first place that "All round squares are not at all round or square since they don't exist in the first place".

4.5.2 A short history of the Liar paradox

4.5.2.1 Introduction

The Liar has been a problem for over 2000 years, not only by itself, but also, via self-reference, as the root of many other paradoxes, antinomies and vicious circles (see Hughes & Brecht (1979)).

The state of the art of solving the Liar paradox is rather disappointing, see Martin (1970). There is a great diversity in opinion about what actually is *the* solution of the paradox. Moreover, when authors are close to the solution then it is not recognized as such or not put forward in a convincing manner. For example, Hodges (1978:164-165) presents sufficient material to deal with the Liar but he does not mention its solution. Also the contribution of Fitch in Martin (1970) is important since it points at the solution - but points only. Kripke (1975) expounds many good observations too, yet in the end opts for another variant of the theory of types.

The following short history selects the elements that contribute to the solution.

4.5.2.2 Antiquity

At the beginning of Abstract Reflections on the Scientific Method, and in the course of founding Logic, Aristotle held that propositions are either true or false: “In the case of that which is, or which has taken place, propositions, whether positive or negative, must be true or false. Again, in the case of a pair of contradictionaries (...) one of the two must be true and the other false.” (Bochenski (1970:62))

If we accept the theory of Beth (1959:23) about the rivalry and resentment between the schools of Athens and Megara then it was in a mood of mockery that Eubulides of Miletus, member of this school of Megara, questioned this Aristotelian point of view by presenting a powerful counterexample: “Whoso says “I lie”, lies and speaks the truth at the same time.” (although these are not Eubulides’ own words but a good approximation of them - see Bochenski (1970:131-132)). In the light of the evolving events it would have been more appropriate when Eubulides had stated this in a more solemn mood than mockery. It is a pity that he didn’t develop an alternative system.

Aristotle didn’t have a good reply to this attack. He argued weakly that the sentence “I lie” (Greek *pseudomenai*) might be true in some particular aspects, but false in an absolute manner. In itself still a potentially fruitful approach yet still undeveloped. Bochenski (1970:132): “Aristotle deals with the Liar summarily in that part of his Sophistic Refutations in which he discusses fallacies dependent on what is said “absolutely and in a particular respect”.” Aristotle: “The argument is similar, also, as regards the problem whether the same man can at the same time say what is both false

and true: but it appears to be a troublesome question because it is not easy to see in which of the two connections the word "absolutely" is to be rendered - with True or with False. There is, however, nothing to prevent it from being False absolutely, though True in some particular respect (...)" One might interprete this as three-valued logic. But apparently that needs more development to become acceptable.

The latter approach was certainly not accepted by Aristotle's contemporaries, and the apparent insolvability of the problem attracted the attention of many. Theophrastus, a disciple of Aristotle, wrote three books on the subject, and Chrysippus of Soli perhaps 28. We already remembered the sad passing of Philetas of Cos.

But it were the Stoics, and among them Chrysippus of Soli (ca. 281-208 BC) who indeed brought the problem to some solution. To start with, they accepted a basic notion of Aristotle about the subject matter of logic: "Yet not every sentence states something, but only those in which there is truth or falsity, and not all are of that kind. Thus a prayer is a sentence, but is neither true nor false ... the present theory is concerned with such sentences as are statements." (Bochenski (1970:49))

These *propositions* (our vocabulary) must, according to Aristotle, appeal to the soul (the origin of reason): "Certainly as sight is in the body, so is reason in the soul (...)" (Bochenski (1970:48)) and "All syllogism and therefore *a fortiori* demonstration is addressed not to outward speech but to that within the soul." (Bochenski (1970:46)). Aristotle was via Plato a grand-pupil of Socrates.

It does not seem incorrect to say that the soul (mind) is the center of the meaning of concepts and thus at the same time the faculty that distinguishes sense (propositions) from nonsense (merely sentences). (See Bochenski (1970:46) and Beth (1959:24))

Aristotle however did not universally apply his intention. It are the Stoics who make a start with its general application. Chrysippus says: "The (fallacy) about the truth-speaker and similar ones are to be (solved in a similar way). One should not say that they say true and (also) false; nor should one conjecture in another way, that the same (statement) is expressive of true and false simultaneously, *but that they have no meaning at all*. And he rejects the aforementioned proposition and also the proposition that one can see true and false simultaneously, and that in all such (matters) the sentence is sometimes simple, sometimes expressive of more." (Bochenski (1970:133), my italics) There is also another fragment of Chrysippus: "they (who state the Liar) completely stray from word meaning; they only produce sounds, but they don't express anything" (Beth (1959: 24)).

What has been translated here as "meaningless" would be translated in our vocabulary as "senseless". We see that a clear distinction between propositions and sentences is

being made, in such a manner that only propositions are (either) true or false by virtue of the fact that only they carry sense; and since the Liar causes a contradiction it must be senseless and henceforth it must be merely a sentence and not a proposition.

Beth (1959:24) suggests that the majority of the Stoics did not accept Chrysippus' view but when I regard Beth's theory about that majority then I am inclined to conclude that they in fact did accept it. First there is the general Stoic definition of a proposition ("statement") as "a sentence that is either true or false"; secondly the equally generally accepted definition of truth and falsehood: "true is that which happens to be and is opposite to something, and false is that which does not happen to be and is opposite to something". The Liar has no clear opposite (most likely not "I speak the truth") and hence it follows that the Liar is not a proposition ("statement"). This approach is identical to that of Chrysippus once you start to look into the property of "being opposite to something" and see that this implicitly appeals to some meaning or sense. It can best be translated as "having sense" since it is only for sensical statements that we can imagine an opposite.

Now, concerning these historical remarks, a word of caution is required. For one thing, only limited information about the opinions of Aristotle and other exists, and the material that exists is often fragmentary and only partially preserved. Experts may disagree about translations and filling of the gaps in fragments. For example, there are many possible words for the concept of 'sentence', e.g. statement, proposition, assertion, expression, theorem, but in a theoretical development these words get a specialized meaning, and then translation becomes a critical issue, with the risk of imposing later theory upon earlier times. Reading Beth (1959) and Bochenski (1970) one may wonder whether these two authors have applied the same translation code. We find the same problem with respect to the translation of "meaning" or "sense". The proper conclusion is that these historical quotes, how valuable they may be, cannot be used to reach a final judgement on the opinions of the ancient Greek on the subject of the Liar. They only generate tentative questions and answers, and thus provide some historical insight and a starting point for further analysis.

Continuing this short historical review, it may be mentioned that the Liar also occurs int he Bible, namely where Paul says: "One of themselves, a prophet of their own, said: "Cretans are always liars, wily beasts, lazy gluttons."" (Titus I:12-13, see DeLong (1971:30)) It is generally assumed that this prophet was Epimenides, who lived at the beginning of the 6th century BC. As a result the Liar paradox is sometimes ascribed to Epimenides: but this must be wrong, since Epimenides certainly would not have been worrying about the logical problem. If he would have uttered that phrase, it could only have happened because he did not see that it should be false by virtue of the fact that he

himself was a Cretan - and thus a liar, wily beast and lazy glutton himself. Also apostle Paul did not see this, since he adds the comment: “This testimony is true.” (Titus, idem)

The “Church Father” Augustine can be mentioned to have studied the paradox.

4.5.2.3 Middle Ages and Renaissance

By the time of the Middle Ages the works of the Stoics were completely forgotten and only the works of Aristotle became gradually known, where the latter eventually caused the Renaissance. The Medieval Scholastics thus had to remake the same development from Aristotle as the Stoics had.

Perhaps one way to test a logical theory is to cause a downfall of civilization and then observe what the new upstarts will invent. About Scholastic logic Bochenski notes: “the problem of semantical antinomies was faced in really enormous treatises. Numerous antinomies of this kind were posited and (...) more than a dozen different solutions attempted. Between them they contain nearly every essential feature of what we know today on this subject.” (Bochenski (1970:251))

Indeed, Paul of Venice (ob. 1429) gives about 15 solutions, of which are interesting (Bochenski (1970:242)): “The fifth opinion states that when Socrates says that he himself says what is false, he says nothing ...” (reminding us of Chrysippus), and “The sixth opinion states that the insoluble is neither true or false but something intermediate, indifferent to each” (a three-valued logic). It is not useful here to repeat why Paul of Venice thought these approaches to be inadequate, and why he thought them to be *different*. (Our development takes these as being the same.) Instead it is useful to note three points:

1. Instead of using the sentence “I lie” the Scholastics start using other forms and notable the sentence “This sentence is false”. Buridan even uses “The sentence written on this page is false” - which sentence must be written on a page that is further blank. The latter procedure helps to eliminate vagueness about words like “lie” and “this”. The word “lie” namely may not be purely logical but might have connotations of deceit, untrustworthyness or plain badness, which might confuse the purely logical issue. Also the selfreference is not simple anymore but sufficiently complex to convince you to accept it.
2. The Scholastics start noticing the composition of the Liar. When L = “This sentence L is false” then we note the occurence of L on two sides, and here it is noted by Pseudo-Scotus: “It is to be said that a part as a part cannot stand for the whole proposition.” (Bochenski (1970:238)) Elaboration on the latter by William of Ockham (ob. 1349) - also known from “Occam’s razor” (“essentialia non sunt multiplicanda”) - brings us very close to a solution. An extensive quote from Styazhkin (1969:44) is

relevant, and indicates Ockham's awareness of the distinction between equivalence (⇔) and identity or definition (=).

"According to Ockham, the source of the antinomy lies in thet fact that the terms required for notation of propositions are sometimes used for notation of the same propositions in which they are used as constituents. More plainly, Ockham meant that part of a proposition (the predicate "is false") must not (in order to eliminate an antinomy) refer to the entire assumption in which it appears. (The problem of this type of restriction did not appear in connection with the predicate "is true", because, in defining truth, Ockham, following Aristotle, assumed that the propositions (1) p and (2) "p is true" are equivalent.) Ockham's view therefor reduces to an interdictum of circular definitions. In other words, it is not permissible to require linguistic constructions in which, for example, a given proposition appeals directly to falseness proper. (...) By eliminating circular arguments in his studies of paradoxes, Ockham was led to his famous 'razor' ("essentials should not be multiplied more than necessary"). He believed that his solution to the problem was the most general possible (in the sense that if his proscription is followed, antinomies do not appear."

3. There is the development of systems of sentences that give rise to paradoxes. Styazhkin reports that Buridan, pupil of Ockham, found a counterexample "in which Ockham's proscription is not violated (there is no direct appeal to falseness), but an antinomy nonetheless appears. The system S consists of the following statements: (A) "man is an animal", (B) "only A is true", (C) "A and B are the only available statements". (...) The system S therefore contains a paradoxical statement, B. In view of this type of expression, Buridan distinguished two types of paradoxical statements: The first type contains 'direct reference' (...) while the second type contains 'indirect reference'." (Styazhkin (1969:44-45)) Having established the concept of indirect reference, Buridan did not simply forbid its use (as Occam had done for direct reference), which is sensible since such indirect reference might be caused by more people and it is rather difficult to control their actions. Buridan's proposed solutions need not be considered here, however.

In the flowering of the Renaissance most of the works of the Scholastics were put aside, as old and dusty, including the works on logic, although the clergymen formed a small group in which the ideas were kept alive.

4.5.2.4 Since Leibniz, Boole, Frege, Peano and Russell

With the work of Leibniz (1646-1716), Boole (1815-1864), Frege (1848-1925) and Peano (1858-1932) a new interest in Logic awoke, in which logic developed distinct mathematical features. The interest in the Liar paradox however only returned after 1900 when Russell linked the Liar to his own paradoxes of set theory. He proposed a

theory of types as a general method for solving and evading paradoxes, see Russell (1967).

To form the Liar paradox, one needs to construct p = "p is false" and counting the quotation marks on either side of the equality sign we see that the right hand side has more quotation marks than the left hand side, so that it might be said to be 'of a higher type'. By the use of the equality sign it however is expressed that they would be of the same 'type'. The theory of types is that every sentence has a unique type. Thus, either the theory of types is incorrect for language and one has to accept the contradiction created by the Liar paradox, or one adopts the theory of types and then it is impossible to construct the Liar.

At first Russell expressed his doubts about the philosophical implications of a type structure of language. Russell & Whitehead (1910-1913, reprint 1964) state in their Principia Mathematica, p vii:

> "A very large part of the labour involved in writing the present work has been expended on the contradictions and paradoxes which have infected logic and the theory of aggregates. We have examined a great number of hypotheses for dealing with these contradictions; many such hypotheses have been advanced by others, and about as many have been invented by ourselves. Sometimes it has cost us several months' work to convince ourselves that a hypothesis was untenable. In the course of such a prolonged study, we have been led, as was to be expected, to modify our views from time tot time; but it gradually became evident to us that some form of the doctrine of types must be adopted if the contradictions were to be avoided. The particular form of the doctrine of types advocated in the present work is not logically indispensable, and there are various other forms equally compatible with the truth of our deductions. We have particularized, both because the form of the doctrine which we advocate appears to us the most probable, and because it was necessary to give at least on perfectly definite theory which avoids the contradictions. But hardly anything in our book would be changed by the adoption of a different form of the doctrine of types. In fact, we may go farther, and say that, supposing some other way of avoiding the contradictions to exist, not very much of our book, except what explicitly deals with types, is dependent upon the adoption of the doctrine of types in any form, so soon as it has been shown (as we claim that we have shown) that it is possible to construct a mathematical logic which does not lead to contradictions. It should be observed that the whole effect of the doctrine of types is negative: it forbids certain inferences which would otherwise be valid, but does not permit any which would otherwise be invalid. Hence we may reasonably expect that the inferences which the doctrine of types permits would remain valid even if the doctrine should be found to be invalid."

It seems however that Russell's association with Wittgenstein brought him to firm acceptation. Wittgenstein struggled at that time with the problem how it could be

possible to discuss the relation of language to reality, when language is presupposed when one discusses anything, see Wittgenstein (1921, 1976:8). Russell proposed the following solution. The distinction of mention and use of a sentence also created the distinction between an object language and a meta-language. Such as one can discuss Dutch and German in English. All sentences of the same type can be said to have a same level, so that there are levels in a language, with a lower level being the object of a meta-level. Thus the relation of an object-language to reality can be discussed in a meta-language - so that Wittgenstein's problem was non-existent.

Though Wittgenstein did not accept this solution (who probably perceived that he could also use German to discuss English), many others did, notably Tarski (1949). In fact, according to Beth (1959:493): "the final disentangling of the knot is due to Tarski". Let us suppose that a definition of truth is possible for a language, and of course we require for such a definition that it does not lead to contradictions. Then however the Liar is formed, resulting into a contradiction. Therefor, no such definition of truth is possible. Beth (1959:511): "It follows that, if an adequate definition can be given of truth and falsehood for sentences in any formal system A (...) then this definition cannot be reproduced within that system A itself; for otherwise we could again derive the liar paradox within the system A. This last remark shows what exactly is wrong with ordinary language from our present point of view; for it will be clear that, if an adequate definition of truth and falsehood could be given for sentences contained in ordinary language, then this definition could always be restated within ordinary language."

Thus the views of Russell and Tarski merge here, holding that an adequate definition of truth is not possible in ordinary language as we commonly understand it but only in a formalized restructured language, created by logical and mathematical means, with a well-defined structure of types: in which the lowest level only discusses material objects and in which the concept of truth of one particular level only applies to sentences of the level just below (or perhaps all lower levels). Users of ordinary language might not be aware of this, just as a person might ride a bike without being able to explain how. In so far as ordinary language appears useful that can only be since there might be hidden processes in the mind that we are not aware of that cause us to respect the levels of application.

Beth (1959:510): "The Liar's statement (...) must be accepted as existing; it is printed in the present book and the reader may read it aloud, if he should wish." Since the Liar exists (in natural language) the applied concept of truth must be inadequate. Therefor it is necessary to formulate another (artificial) language (with types) in which an adequate definition of truth can be given. And when one supposes that there is a Liar statement then it can be shown that there isn't any.

PM. Paradoxical instances apparently can be resolved. A level-Liar $L[n]$ = "$L[n$-1] of level n-1 is false" might be prevented by holding that the zero level that discusses reality has no lower language level. When various people refer to each other so that the level is undetermined then they must be required to use the artificial language and specify the appropriate level.

The views of Russell and Tarski coincide with the Stoic view, most forcefully expressed by Chrysippus, that the Liar is a nonsensical statement, for either it violates the type structure or it requires an inadequate definition of truth. Contrary to the Stoics Russell and Tarski give more specific reasons how the nonsense is caused. The Stoics gave some reasons but Russell and Tarski are more specific.

PM. The Stoic view is not entirely without problems either. If the sentence L = "This sentence is false" is nonsensical then it is neither true nor false. That implies that it must be not true, at least. But L says that it is not true. Hence it is true and thus not nonsensical. Etcetera. See DeLong (1971:242). But also see Chapter 7 to resolve this.

Russell and Tarski solved the contradiction of the Liar, but at some costs:

- The declaration of a type structure basically means the exclusion of self-reference. A sentence cannot refer to itself since it cannot be at two levels at the same time. Why not simply forbid self-reference ? Using levels obscures the destruction of self-reference. To destroy self-reference is a high price since it might be a nice property. In fact, language is self-referent at times, so that the theory of types is inadequate for natural language.

PM. Wittgensteins problem can be judged non-existent because of self-reference. When a sentence discusses the relation between itself and reality then it is no problem that language is presupposed when that language can be self-referent.

- With Russell and Tarski we may adopt the notion that ordinary or natural language does not rigidly recognize a theory of types. But the conclusion that a definition of truth then becomes impossible for mathematical logic may be too strong. Such a conclusion violates our basic intuitions about the generality of the concept of truth. Kripke (1975:694-695) expresses that unease: "Thus far the only approach to the semantic paradoxes that has been worked out in any detail is what I will call the 'orthodox approach', which leads to the celebrated hierarchy of languages of Tarski. (...) Philosophers have been suspicious of the orthodox approach as an analysis of our intuitions. Surely our language contains just one word 'true', not a sequence of distinct phrases "true_n", applying to sequences of higher and higher levels."

The theory of types holds that the Liar paradox is caused by an inadequate definition of truth. Some reject that view, see Martin et al. (1970) and notably Kripke (1975:698): "Almost all the extensive recent literature seeking alternatives to the orthodox approach

- I would mention especially the writings of Bas van Fraassen en Robert L. Martin (1970) - agrees on a single basic idea: there is to be only one truth predicate, applicable to sentences containing the predicate itself; but paradox is to be avoided by allowing truth-value gaps and by declaring that paradoxical sentences in particular suffer from such a gap."

Thus a third value ("gap" or "paradoxical" or Undetermined or Indeterminate) is introduced, offsetting the dichotomy of True | False. Indeed, if the Liar cannot be formulated in a theory of types *T*, but we *can* formulate it in ordinary language, where do you leave it ? Normally you would have a box on the attic to store stuff you don't use.

However, for example Kripke again accepts some types and limitations thereby, see Kripke (1975:714): "(...) there are assertions we can make about the object language which we cannot make in the object language. For example, Liar sentences are *not true* in the object language, in the sense that the inductive process never makes them true; but we are precluded from saying this in the object language by our interpretation of negation and the truth predicate. (...) The necessity to ascent to a metalanguage may be one of the weaknesses of the present theory. The ghost of the Tarski hierarchy is still with us." Kripke also refers to "natural language in its pristine purity, before philosophers reflect on its semantics (in particular, the semantic paradoxes)" and thus suggests that natural language might not even be able to discuss and solve the paradoxes ...

Quine (1990:80) states: "Ascription of truth just cancels the quotation marks. Truth is disquotation." (In *Mathematica* the opposite procedures are ToString and ToExpression.) However, for a language he also requires: "Its truth predicate (...) must be incompletely disquotational. Specifically, it must not disquote all the sentences that contain it." (p83) and "And even these excluded applications can be accomodated by a hierarchy of truth predicates. (...) definability of truth for a language within the language would be an embarrassment." Thus the theory of types is now limited to sentences referring to truth. A sentence "This sentence is not true$_n$" either couldn't be formed or may be judged false$_{n+1}$.

4.5.2.5 Summing up

The following points may be noted:

- The level (type) structure may not apply to the whole language since there exist cases of self-reference (not all paradoxical).
- The distinction between levels in languages may not be of a logical but a linguistic nature, meaning that you might do it, for example as discussing English in Chinese, but in all likelihood there is no logical necessity to claim levels.

- The adoption of a three-valued logic seems most general. Yet, we must show that contradictions can be avoided. In itself this is a difficult concept since a "contradiction" seems only defined in two-valued logic, so that it might not be clear what we should avoid when we adopt three-valued logic.

These notitions are inherent in the writings of Aristotle. Obviously he was conscious of neglecting meaningless and disinformative sentences when he dared to say that all sentences are true or false. Regrettably he was not explicit and complete on the assumption and nor were his followers. The adoption of a three-valued logic hence can be experienced as very un-Aristotelian. But it need not really be so.

This historical outline has been useful to pose the problem, introduce some concepts, and arrive at tentative answers and research questions. We now wonder what the definition of truth really is, how three-valued logic looks like, and so on. In proceeding, we should heed the warning by Beth (1959:518): "various divergent opinions on this topic have been defended by a number of authors; these heterodox opinions, though frequently very acute, result in most cases from an underestimation of the methodological importance of the paradoxes and from a neglect of the standards of rigour which are characteristic of contemporary formal logic."

4.6 A logic of exceptions

4.6.1 The concept of exception

Human history abounds with exceptions to rules. Regard the following examples (except the fourth):

- A law like "All murderers will be hanged". Well, of course, except that they may ask pardon.
- There can be exceptions to a rule, that prove a rule.
- Our life is ruled by doctrines. Except that life is stronger than any doctrine.
- Catch 22: Soldiers remain in battle, except those who lost their sanity. If you apply for the exception then that is taken as a proof of sanity.

There are different ways to express exception. In the format of set theory we might define for a set A:

A == {x | (conditions[x]) ~Unless~ (exceptions[x])}

$$A = \{x \mid (\neg \text{ exceptions}(x) \Rightarrow \text{ conditions}(x))\}$$

- For example:

A == {x | Even[Sqrt[x]] ~Unless~ (x ≤ 0)}

$A = \left\{x \middle| \left(x > 0 \Rightarrow \text{Even}\left(\sqrt{x}\right)\right)\right\}$

- In case you are worried about areas where A is undefined, you can use the *p* unless *q*, then *r* construction, like: "I'll take an apple, unless you pay, for then I'll take a big chocolate fudge".

A == {x | Unless[conditions[x], exceptions[x], then[x]]}

$A = \{x \mid ((\neg \text{exceptions}(x) \Rightarrow \text{conditions}(x)) \bigwedge (\text{exceptions}(x) \Rightarrow \text{then}(x)))\}$

The difference between conditions and exceptions is of a psychological nature, in that the exceptions tend to occur less frequently, even so much less that people tend to forget about them. But in a formal sense they are conditions like all the other. Except, that exceptions tend to dominate the other conditions. Normal conditions don't apply when the exception strikes.

Mathematica has well-developed features for pattern testing and keeping account of rules of exception.

- The function Condition (symbol /;) tests the input of a function and only allows it to evaluate when the condition is satisfied. The function thus does not evaluate except when the conditions are satisfied.

testedSqrt[x_] := Sqrt[x] /; x > 0

{testedSqrt[10], testedSqrt[−10]}

$\left\{\sqrt{10}, \text{testedSqrt}(-10)\right\}$

- The very symbol Except represents a pattern object that fits anything except those that match it.

Cases[{a, b, c, d, e, f, 1}, Except[_Integer]]

$\{a, b, c, d, e, f\}$

We may take the concept of exception to be well-established.

4.6.2 The liar

Our historical review of the Liar gave various cases, with Aristotle, the Stoics, the medievals, where part of the proposed solution was to introduce an exception. The main exception proposed was that Tertium Non Datur holds unless a sentence is not a proposition.

(p ⋁ ¬ p) ~Unless~ NotAProposition[p]

$(\neg\ \text{NotAProposition}(p) \Rightarrow (p \vee \neg p))$

We will develop this in Chapter 7 as "three-valued logic".

4.6.3 Selfreference in set theory

The problems of the Liar and Russell's set paradox hinge upon our freedom to define concepts at liberty. There is nothing in the concepts of linguistics or aggregation that suggests, at first glance, that we cannot compose all kinds of combinations of the concepts and symbols, and then test their truthvalue. Except that we now have instances for which we cannot find such a truthvalue. The approach since Russell has been his theory of types, such that well-defined concepts satisfy a hierarchical order. Syntax then replaces inference. Set A or sentence B only apply to sets or sentences of a lower level. A problem with this approach is that is runs counter to the natural feeling that language and aggregation are very flexible. Levels exclude selfreference while it is an attractive property.

It can also be shown that we rather need selfreference so that we should allow it.

The original definition of a set came from Cantor: "Unter einer 'Menge' verstehen wir jede Zusammenfassung M von bestimmten wohlunderschiedenen Objecten m unsrer Anschauung oder unseres Denkens (...) zo einem Ganzen" (Zermelo (1908:200)). Recall our notation of before, with $R \equiv \{y \mid y \notin y\}$ the Russell set of "normal" sets that do not contain themselves. In that period it seemed that R satisfied Cantor's definition. Because of his paradox, Russell restricted the kind of membership. Zermelo, who worked in the same field, and actually discovered the paradox independently, decided to question Cantor's definition. He suggested the notion of *definiteness*: "A question or assertion (...) is said to be *definite* if the fundamental relations of the domain, by means of the axioms and the universally valid laws of logic, determine without arbitrariness whether it holds or not." (p201). Subsequently he designed the axioms of set theory that are basically still in use today. There are different comments on this: (a) the notion of definiteness actually implies a three-valued logic, namely for sentences with terms that are indefinite, (b) we may wonder whether definiteness really differs from Cantor's "wohlunderschieden", my bet is that it doesn't, (c) Zermelo's axioms reflect a variant of the theory of types, though he doesn't call it that way, (d) we still seem to lose selfreference somewhere.

Historically, we may observe that Russell's struggle was different than Zermelo's. At that time he seems to have had a Platonic conception of mathematics, in which indeed a thought would have to 'exist' in some way or other, and thus would by itself be "wohlunterschieden" (definite) instead of possibly nonsensical. His development of the

theory of types was part of his struggle to reduce his Platonic tendencies. At the end of the Principia Mathematica he writes: "it appeared that in practice the doctrine of types is never relevant except where existence theorems are concerned or where applications are to be made to some particular case" (Russell & Whitehead (1964:182)). For us it is less worrying that nonsensical things can 'exist' in a box in the attic where they are no longer used.

Reconsider the Russell set paradox, and consider that we might allow for normal sets, that do not contain themselves, to be contained in some *R*, even if the are "of higher type" than *R*.

A cousin of *R* is *Z*, called the Zermelo set of normal sets, that includes a small consistency condition that actually imposes a very useful form of selfreference.

$$Z = \{ y \mid y \notin y \wedge y \in Z \}$$

Clearly, if we consider the set of teaspoons then this holds without problem. When we try $Z \in Z$ then the condition for membership appears to be unsatisfied (a contradiction) whence we conclude that Z is not a member of itself. This approach reminds of the one of Paul of Venice to solve the Liar paradox (see Bochenski (1970) and Beth (1959)).

When you think that requiring $y \in Z$ for the set of teaspoons is overdone, then we can do, which indeed is a more general approach:

- If $y \neq Z$ then $y \notin y$
- If $y = Z$ then $y \notin y$ but for consistency we require $y \in Z$.

This is formulated jointly as that $y \notin y$ applies unless $y = Z$, for then the consistency requirement is explicitly imposed. Of course, consistency holds in general, but when we limit our attention to a single statement then we must translate overall consistency to the particulars required for the expression under scrutiny.

- Solving Russell's paradox with Unless.

```
def = $Equivalent[y ∈ Z, Unless[(y ∉ y), y == Z, (y ∉ y) ⋀ (y ∈ Z)]]
```

$(y \in Z \Leftrightarrow ((y \neq Z \Rightarrow y \notin y) \bigwedge (y = Z \Rightarrow (y \notin y \bigwedge y \in Z))))$

- Replace some symbols that LogicalExpand does not recognize.

```
def2 = % // ToBelongs // ToAndOrNot // LogicalExpand
```

$(((y \in y) \bigwedge (y \notin Z)) \bigvee ((y \in Z) \bigwedge (y \notin y)) \bigvee (Z = y \bigwedge (y \notin Z)))$

- For the set of teaspoons that is not a member of itself, we find that it belongs to this Z.

```
def2 /. (Belongs[y, y] → False) /. (Z == y) → False
```

$(y \in Z)$

- Application to itself gives the conclusion that Z ∉ Z, and this holds without contradiction.

def2 /. y → Z

$(((Z \in Z) \wedge (Z \notin Z)) \vee ((Z \in Z) \wedge (Z \notin Z)) \vee (Z \notin Z))$

% // LogicalExpand

$(Z \notin Z)$

Thus there is a way out of Russell's paradox without resorting to a theory of types. When our ancestors invented the 0, this caused paradoxes for them, and some of them really had a hard time, but eventually a decent set of concepts was found. In the same way the "set of all normal sets" apparently must be defined a bit smarter. The ghost of paradox is still with us because of the need to add a consistency requirement. And perhaps the set *Z* is not entirely what you naively had in mind for a "set of all normal sets", but at least consistency has been arrived at.

Another example may help. The definition of a perfect bike is that it has round wheels to ride around on and square wheels to park it. Since there are no round squares, this perfect bike does not exist. Is that something to worry about ? It only shows that our idea of bike perfection was inconsistent in the first place, so that we construct a bike with a standard. This other bike is perfect in that it works. In the same way someone who really wants to work with a "set of all normal sets" will be happy with above definition of *Z*, even when it might not be as originally conceived.

4.6.4 Still requiring three-valuedness

That being said, and having indicated (not proven of course) the possible consistency of a set theory with selfreference, there remains the other observation that *R* has not gone away. It is easy to conclude "do not form it" (on pain of inconsistency) but that still causes some uneasiness. Rules restrict our freedom so there must be a good reason for them.

The crux is: why not allow *all* freedom of definition, and let logic tell us what to do when we reach a contradiction ? The basic thing logic should tell us is: sometimes words may not refer to anything, and, sometimes circumstances are not like we thought.

But even when you hold on to formation rules, then you still would need a place to put all the things that you do not form. Saying "don't define *R* or the Liar paradox" is equivalent to saying "they don't have a value in True | False or 1 | 0". If we assign a value 1/2 then we can construct the following table:

$x \in y$	$x \notin y$	NotAtAll[$x \in y$]
1	0	0
0	1	0
$\frac{1}{2}$	$\frac{1}{2}$	1

When we submit R to our deductive facilities, we derive a contradiction, hence we conclude that R is nonsense, and then we assign a truthvalue 1/2. Or, to show that a set exists, we not only have to show that it has an element (which can always be done by EFSQ), but also that its definition is not inconsistent, so that the concepts are well defined in the first place.

As said, the key issue here is freedom of expression. The world is True | False but our concepts can be nonsensical. It is useful, not only to have the freedom to be creative, but also to have the freedom to err in sense. The best example here is the theory of types: it creates a rigid structure that forbids the elegant construction of the Zermelo Z cousin of R. Rigidity can be too high a price to pay. The Zermelo Z is not only an elegant construction but it also contains all kinds of elements that otherwise would not be contained in that way.

Formation rules may help to restrict nonsense, but they may also create nonsense, as seen from a logical point of view.

PM 1. It is not useful to replace the value 1/2 by 0 since Not[x $\notin$ y] would generate a 1 in the lowest row, and similarly for using 1. Thus three values are required. Perhaps you don't like 1/2 and prefer U or $\mathfrak{U}$ for *undetermined*. It appears that the best choice is *Indeterminate.*

PM 2. A theory of types might still be useful for sets. Cantor proved that the power set is always greater than the set itself. A "set of all sets" would have to contain its power set as a subset, which it thus cannot do. Hence such a "set of all sets" does not exists. Some hierarchies indeed might exist. See Chapter 11, Reading Notes, for a discussion of this argument.

PM 3. Much of this discussion derives from mathematicians who want everything to be really neat. We can only respect this and actually be happy that there are such people. The achievements of people like Euclid are mind staggering. Yet a bit of flexibility would help. Mathematicians tend to be motived by the desire to understand things and to do so they tend to eliminate all exceptions, while to understand the world you must allow for exceptions. Mankind tries to solve the problem of bureaucracy since the times of the pyramids and it would help when the bureaucratic attitude finds no fertile ground in the realm of the mind.

4.6.5 Exceptions in computers

Computer programs have to deal with undefined situations too. When they have a rigid structure then this might cause a lot of errors and thus a great deal of error handling. Programs would be more flexible when they could guess what was intended (or perhaps simply leave things as they are).

- This expression does not simplify to True | False when the variables don't have values. What is its truthvalue ?

p ∧ q ∨ r // LogicalExpand

$(r \lor (p \land q))$

- And what is the truthvalue of this statement ?

p ∧ q ∨

Syntax::tsntxi : "\"p ∧ q ∨\" is incomplete; more input is needed. More… "

Another good example is string concatenation. It requires strings and at least two of them. Forgetting to type one quote differs from forgetting to type two quotes. In the first case the cryptic message prints just on the page you are working with and in the second case there also appears a separate message notebook with another cryptic message.

- Apart from guessing the error, the other question is what its truthvalue is.

"ab" <> "cde

Syntax::tsntxi : "\" ab <> \" cde" is incomplete; more input is needed. More… "

"ab" <> cde

StringJoin::string : String expected at position 2 in ab <> cde. More…

ab <> cde

Note the difference in output (quotes and underscore). The difference between "*cde* and *cde* is in the syntax of *Mathematica* that the first still is waiting to be processed for becoming proper input for the kernel while the latter has already passed that test so that the kernel assumes that *cde* would be a variable. But that interpretation assumes for the first case that the quotation mark in "*cde* was no typing error. When the quotation mark in "*cde* was a typing error, in the sense that the user tried to follow the syntax and tried to put in a variable *cde*, then the user got the wrong message. NB. These cells are locked since later versions of *Mathematica* might handle the situations differently. Also, in the first error message a single quote has been changed into \" since otherwise *Mathematica* gives a third error message (an error on an error message) when opening this notebook at this place.

In both cases the computer fortunately doesn't crash and even reproduces the input. It produces the input so well that you even start to think that it should be able to guess

what was intended. Really, the computer user already typed *some* strings and the *actual symbol* for concatenation. Shouldn't a smart computer be able to guess (as a default solution) in the first case that concatenation of "abc" and "cde" is intended ? (It might test whether cde is a variable.) What now is programmed seems inefficient and confusing, and a good rule for programmers is that you shouldn't program for others what you wouldn't want to be programmed to yourself. A more user friendly approach would be to create that concatenation and report "OK, your input wasn't all up to standards as we perceive them, but we made a guess: if it isn't right, please correct it, and, if you follow our syntax then you don't get messages like this."

Actually, it is already a step ahead in politeness that these messages aren't called "error" messages. An "error" in this case is only a difference of opinion on what syntax is acceptable.

This is just an example. What counts is the philosophy. (a) Dealing with exceptions to systems that you design. (b) Truthvalues of nonsense (given some syntax).

Of course, in the concatenation example, we require context-dependency. If this entering of code was your last try at the password that you forgot for your banking account, then perhaps you would be happy with some degree of strictness. And of course, hoping for user-friendliness is also a form of strictness. It would perhaps be too much to demand that there be whole company departments staffed with people trying to guess what would be the hidden meaning of input "errors" (i.e. caused by syntax of other departments). (This becomes complex. Perhaps it is better to become completely cynical and not care about anything anymore. Let us quickly move on to *inference.*)

5. Inference

5.1 Introduction

5.1.1 Introduction

Inference focusses on the process of handling information. A collection of individual observations gives rise, via *induction*, to the conjecture of a general relationship. A collection of general relationships gives, via *deduction*, new general relationships, or, with some particulars, other particular conclusions. The information that is processed must be there otherwise it cannot be processed. Inference essentially assumes the handicap that the information is not recognized immediately. It must first be processed before it is recognized in the form that it can be used. And the final response should be "Ah, this is *news*."

Logic and mathematics are the deductive sciences. The proper ways of induction are generally left to the other sciences. Yet it will be useful to include a section on induction to clarify where the general laws come from that are used in logic and mathematics. Other books start with "self-evident truths" but it is useful to be specific.

The analysis of inference requires a taxonomy of human mental processes. For the lovers of the medieval square this taxonomy presents a Land of Cockaigne.

- First load the package.

Economics[Logic`AIOE, Print → False]

PM. The notation ⊢ of valid and ↧ of invalid inferences might suggest to the TruthTable routine that there are two different forms involved, a P and an independent Q. However, we can recover the proper propositional form that shows their connection.

- TruthTable recogizes some propositional forms.

Ergo[S, p] ⋀ NonSequitur[S, p]

$((S \vdash p) \bigwedge (S \mathrel{↧} p))$

PropositionalForm[%]

$(P_1 \land \neg P_1)$

5.1.2 Validity

Some deductions may be formally valid but for irrelevant reasons. From a falsehood you can derive anything you want, which in Latin is the "ex falso sequitur quodlibet" (EFSQ) argument. An example is the statement (truth): "If it rains and does not rain at the same time, then you'll get a new car." Hence the following distinction is useful:

a. An argument is formally valid if its static projection is a tautology

b. An argument is materially valid if it is formally valid and if its premisses are free of contradictions and (known) falsehoods

- The EFSQ argument structure

TruthTableForm[(¬ p ⇒ (p ⇒ q))]

Implies	Not *p*	Implies *p* *q*
True	False True	True True True
True	False True	False True False
True	True False	True False True
True	True False	True False False

- A more drastic EFSQ example

TruthTable[(p ⋀ ¬ p) ⇒ q]

$$\begin{pmatrix} p & q & ((p \land \neg p) \Rightarrow q) \\ \text{True} & \text{True} & \text{True} \\ \text{True} & \text{False} & \text{True} \\ \text{False} & \text{True} & \text{True} \\ \text{False} & \text{False} & \text{True} \end{pmatrix}$$

5.2 Induction

5.2.1 Introduction

Part of life is the processing of information. Organisms apparently have sorts of memories. Some memory is built into the hardware and then called instinct. The more

complex organisms may have memory banks and rules to store information. But even lifeless objects have some memory, as a screwdriver by its shape "remembers" that it can handle only some types of screws. Summarizing a list of events into a general rule is called induction - as opposed to deduction that goes from the general rule to the particular.

- Mathematical induction or recursion is a formal scheme to come from the particular to the general, without apparently referring to reality.
- Empirical induction are the ways of empirical science, to determine empirical laws.
- The definition & reality methodology is the approach to turn empirical laws into truths by definition.

Deduction relies on induction. Without induction there would be no general rules, and without general rules there can be no deduction. Deduction may start with "self-evident truths" but these came about from induction or definition.

It would have been preferable to introduce induction in the introduction, but it requires some notation that is only developed in the course of this book.

5.2.2 Mathematical induction or recursion

We will take recursion and mathematical induction as equivalent terms. The term "recursion" unfortunately does not explicitly refer to "induction".

Let A[x] be the property under consideration and let ℕ be the set of natural numbers, thus ℕ = {0, 1, 2, 3, ...}. We want to know whether property A holds for all natural numbers. Mathematical induction consists of this argument:

Ergo2D["Mathematical Induction", A]

1	It can be shown that property A holds for n = 0, thus A[0]
2	It can be shown for arbitrary n that if A[n] then also A[n + 1]
Ergo	————
3	A[n] for all n ∈ ℕ

The above starts with n = 0, but it might also start with any other number and prove a property for all numbers from there onwards. You might take bigger steps than one, considering only a subset, and you might use more variables.

This kind of scheme is called "procedural" or "constructive". You don't have to work through all natural numbers but merely write a decision procedure or "algorithm" to check things out, and, you can check that because the procedure is *finite*. A useful term thus is "effective finite procedure". Of course, for complex issues computation times even for "finite" cases might be long.

Sometimes properties are defined in this precise manner. Precisely the set of natural numbers can be defined by using the idea of "the successor" or "the next". Let 0 be the

first element, 1 is next to 0, 2 next to 1, etcetera. The best way to understand mathematical induction is to see that it mimics this kind of definition. The "for all" in row 3 of the scheme above actually does not refer to all natural numbers as a completed set, but uses that if you are at number *n* then you can take the next one (actually defining "for all").

An example of a definition by means of recursion is: when we know what addition is, then we can define multiplication as follows. Let us not use the word "multiplication" yet but a vague term "operator". Then define: (1) x *operator* 0 = 0, (2) x *operator* (y + 1) = x *operator* y + x. Henceforth we will call this *operator* "times".

Other examples of recursion are the procedures FoldList or FixedPoint in *Mathematica*:

FoldList[f, x, {a, b, c}]

$\{x, f(x, a), f(f(x, a), b), f(f(f(x, a), b), c)\}$

The principle of mathematical induction would hold for all possible predicates *P*. In this case the "for all" might be vague since there doesn't seem to be a natural order for predicates.

- Definition of recursion. *Mathematica*'s routine ForAll does not directly recognize *P* when it is used only as a predicate. Therefor we mention it explicitly.

ForAll[P, Ergo[P, P[0], ForAll[n, (P[n] ⇒ P[n + 1])], ForAll[n, P[n]]]]

$\forall_P (\{P, P_0, \forall_n (P_n \Rightarrow P_{n+1})\} \vdash \forall_n P_n)$

PM 1. Mathematical induction is defined for the natural numbers. Real numbers require the continuum (see e.g. L.E.J. Brouwer).

PM 2. Though mathematical induction has been presented here in the context of induction, where it must be catalogued to start with, later we will see it in the context of deduction again. When concepts have been defined in terms of recursion then one would use recursion again for proofs.

5.2.3 Empirical induction

Economics[Probability, Print → False]

Suppose that you note that sometimes it rains and sometimes it doesn't rain. You also observe that sometimes the streets are wet and sometimes they aren't. You become interested in the combined occurrences. You collect some 100 cases. In all the 25 cases that it rains the streets are wet. There are 3 cases where it didn't rain but the streets are still wet (e.g. because the road cleaning car came by). The data can be collected in the following table.

- The observation matrix has 9 cells but only 4 degrees of freedom, because of the summing of rows and columns. In the Disease-Test matrix in epidemiology the

disease (cause) standardly is in the columns and the test-result (effect) in the rows. We assume that rain is the cause. Is the observation that the streets are wet a good predictor for what is the cause ?

mat =

"Observation count"	"It rains"	"It doesn't rain"	"Total"
"The streets are wet"	25	3	□
"The streets are not wet"	0	□	□
"Total"	□	□	100

;

```
Headed2DTableSolve[mat]
```

Observation count	It rains	It doesn't rain	Total
The streets are wet	25	3	28
The streets are not wet	0	72	72
Total	25	75	100

Instead of remembering all these 100 cases either individually or by frequency distribution, the memory processing unit might save on storage and retrieval costs by adopting a general rule that "If it rains then the streets are wet".

This can become a general rule for which we can use a truthtable. The truthtable tests the condition whether the frequency is zero or non-zero.

- The SquareTruthTable procedure sorts the texts in alphabetical order, unaware of our hypothesis on causality. Thus we transpose its result.

```
SquareTruthTable["It rains" ⇒ "The streets are wet"] // Transpose
```

(It rains ⇒ The streets are wet)

	It rains	¬ It rains
The streets are wet	True	True
¬ The streets are wet	False	True

Clearly, such general rules must be treated with care since you might run into cases where the rule is refuted. For example, a street running under a bridge or through a covered shopping mall might appear to be dry even while it is raining.

At this point it is useful to note that e.g. an implication is *said* to be a binary operator while it *in fact* relies on other "unobserved" variables or an "error term". If it doesn't rain then another event must explain the street's possible wetness, see Colignatus (2007e).

5.2.4 *Definition & Reality* methodology

One way to insulate a rule against refutation is to turn it into a definition. The concepts of "It rains" and "The streets are wet" can be defined such that "If it rains then the streets are wet" becomes a certainty by definition. For the odd case when you encounter a dry street when it rains then you look for the cause of the exception, and you decide that this is not a "real street". It depends upon the costs for memory management what

is the cheapest solution. You might decide to adopt the rule "If it rains then streets in the open air are wet" so that the general notion of "street" is adapted, or you might keep the general notion of "street" and create the notion of the non-streets as "covered pathways". Anything goes as long as you remain consistent and stick to the facts in your original list of 100 observations. Note that such covered pathways may still be wet when it rains, because of simultaneous cleaning.

The procedure to insulate notions against refutation by turning them into definitions can be called the "definition & reality methodology". While reality is contingent and every moment is new, it is a cost-effective method of memory management to use definitions that allow certain (as opposed to probabilitistic) conclusions.

See Colignatus (2005) for a chapter that explains more and an application of this methodology to economics.

PM 1. There is the example of Wittgenstein who kept on searching for a hippopotamus in the classroom because nobody had given an adequate definition so that he could identify that there was none. This might be the proper attitude for mathematics but in normal science definitions might be rather pragmatic.

PM 2. You would still benefit from a course on probability and applied statistics.

5.3 Aristotle's syllogism

5.3.1 Introduction

Aristotle not only recognized the A, I, O and E forms and that they allow an *immediate* conclusion just by themselves, but he also created forms of *mediate* inference where two premisses are combined to support the conclusion.

Greece (Hellas) at that time suffered from sophists (lawyers and consultants) who designed all kinds of argumentations and advocacy that reeked of non sequiturs. (This is very different nowadays. Mankind has made real progress.) DeLong (1971:12) relates: "many examples of purely verbal contradictions and quibbling are given in Plato's dialogs, expecially in the *Euthydemus*". He gives this example from Dionysodoros (normal letters) and Ctesippos (italics):

".... You say that you have a dog. *Yes, a villain of one*, said Ctesippos. And has he puppies ? *Yes, and they are very like himself.* And the dog is the father of them ? *Yes*, he said, *I certainly saw him and the mother of the puppies come together.* And is he yours ? *To be sure he is.* Then he is a father, and he is yours ; ergo, he is your father, and the puppies are your brothers."

Another problem was geometry, where some authors simply begged the question (if P and Q and R then P), or where A assumed what B proved and the other way around,

creating an intellectual atmosphere that demonstration was impossible or that you could assume anything *anyway*.

Aristotle ambitiously set out to determine forms that were valid.

5.3.2 The four basic figures

When we consider a conclusion like "Thus, Socrates is mortal" then there is a subject (Socrates) and a predicate (Mortal), abstractly S and P. Aristotle did not use the labels *subject* and *predicate,* but used the labels *minor term* and *major term*. On each term we can introduce a separate premiss. The premiss on the major term P is called "the major" and stands first. The premiss on the minor term S is "the minor" and comes next. The conclusion follows as third. Aristotle understood the argument as "If major and minor, ergo conclusion", i.e. as a linear inference and not as 2-dimensional table. Of course, the premisses must be correct statements and thus require another term, called the *middle term M.* Aristotle then considered taking *P, M, S* from the 4 possibilities of the AffirmoNego (AIEO) scheme. These are called categorical syllogisms.

Maintaining the order of first *subject*, then *predicate*, there are two possibilities for the major premiss: {M, P} or {P, M}. There are two possibilities for the minor premiss too: {S, M} or {M, S}. Hence there are four combinations in total, given by the following table. Aristotle used the term *figure* for these combinations, though didn't consider the fourth one. NB. Minor and maior are Latin now.

Syllogism[All, P, M, S]

$$\begin{pmatrix} \text{Maior}(M, P) & \text{Maior}(P, M) & \text{Maior}(M, P) & \text{Maior}(P, M) \\ \text{Minor}(S, M) & \text{Minor}(S, M) & \text{Minor}(M, S) & \text{Minor}(M, S) \\ \text{Conclusion}(S, P) & \text{Conclusion}(S, P) & \text{Conclusion}(S, P) & \text{Conclusion}(S, P) \end{pmatrix}$$

For a categorical syllogism there are 4 figures and 4 possibilities for each line, thus 4 * (4 * 4 * 4) = 256 possibilities. Only 24 combinations are valid schemes for inference, see DeLong (1971:19-24).

The important properties then are:

- You have to assume something. Some points are without demonstration and are self-evident truths. Aristotle: "Our own doctrine is that not all knowledge is demonstrative." (DeLong (1971:20))
- You must use the valid inference schemes. This makes for a valid inference.
- But you must also make sure that the assumptions are True. Thus EFSQ is excluded. Only then there is a proper *demonstration*.

Aristotle: "Syllogism should be discussed before demonstration because syllogism is the more general: the demonstration is a sort of syllogism, but not every syllogism is a

demonstration" (DeLong (1971:20)).

He emphasized that inference concerns issues that make sense, as different from mere expressions: "(...) all syllogism, and therefor *a fortiori* demonstration, is addressed not to the spoken word, but to the discourse within the soul (...)" (DeLong (1971:23)).

Finally (for us) he wrote on giving proper definitions, in order to identify those self-evident truths. According to Aristotle Socrates was "the first to raise the problem of universal definition" and: "it was natural that Socrates should be seeking the essence, for he was seeking to syllogize, and 'what a thing is' is the starting point of syllogism (...)" (DeLong (1971:23)). A mere definition does not cause that the defined term really exists. Also, a good definition requires that its terms are better understood than what is defined.

PM. This is not the only logical scheme that Aristotle used. In his writing he for example also used $p \Rightarrow p$, $\neg p \Rightarrow p$, ergo p, at least in words though perhaps not in formulas. Also all the math and geometry of the Greeks dependent upon reasoning. People generally use all kinds of schemes. At issue here is the formalization of methods.

5.3.3 Application

The following gives some categorical syllogisms.

- This gives the AIOE schemes again, in case you forgot.

Syllogism[AIOE, S, P, Text]

$$\begin{pmatrix} \text{All S are P} & \text{No S are P} \\ \text{Some S are P} & \text{Some S are not P} \end{pmatrix}$$

- This gives the standard syllogism, figure 1, {A, I, I}.

Syllogism[1, {A, I, I}, {mortal, humans, Socrates, Text}]

A	All humans are mortal
I	Some Socrates are humans
Ergo	——————
I	Some Socrates are mortal

- Default is the use of set membership for I.

Syllogism[1, {A, I, I}]

A	$M \subseteq P$
I	$S \in M$
Ergo	——————
I	$S \in P$

- When the conclusion concerns sets then we use the definition $A \wedge B \Leftrightarrow (A \cap B) \neq \emptyset$ when expressing that two sets overlap.

Syllogism[1, {A, I, I}, {Mortal, Human, Men, And}]

A	Human ⊆ Mortal
I	(Men ∧ Human)
Ergo	—————
I	(Men ⋀ Mortal)

- When you prefer the quantifiers then you must mention the variable that is bounded by them.

Syllogism[AIOE, S, P, x]

$$\begin{pmatrix} \forall_x (S(x) \Rightarrow P_x) & \forall_x (S(x) \Rightarrow \neg P_x) \\ \exists_x (S(x) \bigwedge P_x) & \exists_x (S(x) \bigwedge \neg P_x) \end{pmatrix}$$

Syllogism[1, {A, I, I}, {Mortal, Human, Socrates, x}]

A	$\forall_x (\text{Human}(x) \Rightarrow \text{Mortal}(x))$
I	$\exists_x (\text{Socrates}(x) \bigwedge \text{Human}(x))$
Ergo	—————
I	$\exists_x (\text{Socrates}(x) \bigwedge \text{Mortal}(x))$

- Or the latter when it is evident that x is in the domain:

% /. Socrates[x] → True

$$\begin{pmatrix} \text{A} & \forall_x (\text{Human}(x) \Rightarrow \text{Mortal}(x)) \\ \text{I} & \exists_x \text{Human}(x) \\ \text{Ergo} & \text{—————} \\ \text{I} & \exists_x \text{Mortal}(x) \end{pmatrix}$$

The syllogism routine also can give linear output, when you specify that Ergo is to be used instead of Ergo2D.

Syllogism[1, {A, I, I}, Ergo]

({M ⊆ P, S ∈ M} ⊢ S ∈ P)

`Syllogism`[*figure*, {*major*, *minor*, *conclusion*}, {*P*, *M*, *S* (, *x*)}, *d*:Ergo2D]
prints the syllogism of *figure*, where the premisses must be in A, I, E or O, where *P*, *M* and *S* are the expressions for the predicate, middle and subject; *x* may be needed for the quantifier, but may be Text, can be omitted for set membership or can be And for non empty intersections; *d* can also be Ergo or your own function
`Syllogism`[*figure*, "XYZ", {*P*, *M*, *S* (, *x*)}, *d*:Ergo2D]
same as above but with letters X, Y, Z as letters A, I, E or O instead of symbols

`Syllogism[All, P, M, S]`	gives the four possible figures of a syllogism, with P the major term, M the middle term, and S the minor term (S is the subject of the conclusion and P its predicate). The Maior is the premisse on the major term, Minor the premisse on the minor term, and the conclusion concerns the relation of both. A categorical syllogism arises when P, M, S are taken from the 4 possibilities of the AffirmoNego (AIEO) scheme, giving $4 * (4 * 4 * 4)$ or 256 possibilities. Only 24 combinations are valid schemes for inference, Aristotle (384−322 B.C.) and DeLong (1971:19−24)
`Syllogism[AIOE, x, s, p]`	prints the AIOE square

Syllogism has option Messages. Value None gives no messages, False (default) only when a figure and mode is invalid, True only when a figure and mode is valid, All gives always a message.

Note that the predicate is "monadic" *P*[*x*]. Syllogism does not deal with relations such as Brother[*x*, *y*] ⇔ Brother[*y*, *x*].

Next to the universal "all" and the particular "some" there is the singular, e.g. Socrates. It is arbitrary to take the latter as "all Socrates" or "some Socrates".

`Maior[x, y]`	a label for the major premisse in a syllogism. Thus either x or y should be the maior, i.e. the predicate of the conclusion (term P), so that the other is the middle term M
`Minor[x, y]`	a label for the minor premisse in a syllogism. Thus either x or y should be the minor, i.e. the subject of the conclusion, (term S), so that the other is the middle term M

The symbol Maior is Latin that has no j.

`AIOE[A | I | O | E, s, P, Text]`	gives a text, e.g. for A: all s are P
`AIOE[A | I | O | E, s, P]`	uses sets
`AIOE[A | I | O | E, s, P, x]`	uses ForAll x and Exists x
`AIOE[x, S, p]`	gives labels for AffirmoNego, for x is Ergo, Undecided, Unprovable, Undecidable, Consistent

5.3.4 Comments

5.3.4.1 Progress since Chapter 2

The AII scheme was mentioned in the introduction, and from that perspective we haven't achieved much. There are lots of equivalent notations but that seems to add little compared to the sharp reasoning in natural language as we already did. On top of that, the syllogism still allows lots of nonsense, i.e. when we don't ask the routine to test

on what a valid inference scheme is. It may well be that the word 'silly' originated from the normal folk in the village overhearing the monks doing their syllogisms.

Syllogism[4, {A, E, O}, {dogs, fathers, puppies, Text}]

A	All dogs are fathers
E	No fathers are puppies
Ergo	――――――
O	Some puppies are not dogs

It is a suggestion to go to the library, get DeLong (1971) and look on page 21 for the table of Aristotle's 24 valid schemes. You might want to read some more in this good book. Alternatively, you might draw Venn diagrams, and find that various schemes can be reduced to others. The axiomatic method will help us to find the independent axioms that are sufficient for these deductions.

PM. Since we can find a dog that is not a father - e.g. a puppy itself - the inference is true by EFSQ, what explains why Aristotle insisted that the premisses should be true. This scheme is actually valid but would require another example with premisses that are true.

- One direct way to tackle the AIOE schemes is to consider the Exists statement for the element that makes it true, and substitute this same element in the ForAll statement, whereup the quantifiers drop out, and the problem can be tackled by propositional logic.

Syllogism[4, {A, E, O}, {dogs, fathers, puppies, x}, Ergo] /. Ergo["Projection"]

$$((\forall_x (\text{dogs}(x) \Rightarrow \text{fathers}(x)) \wedge \forall_x (\text{fathers}(x) \Rightarrow \neg\, \text{puppies}(x))) \Rightarrow \exists_x (\text{puppies}(x) \wedge \neg\, \text{dogs}(x)))$$

SubstituteQuantors[%, x → case]

$$(((\text{dogs}(\text{case}) \Rightarrow \text{fathers}(\text{case})) \wedge (\text{fathers}(\text{case}) \Rightarrow \neg\, \text{puppies}(\text{case}))) \Rightarrow (\text{puppies}(\text{case}) \wedge \neg\, \text{dogs}(\text{case})))$$

% // LogicalExpand // MatrixDNForm

$$\begin{pmatrix} (\text{dogs}(\text{case}) \wedge \neg\, \text{fathers}(\text{case})) \\ (\text{fathers}(\text{case}) \wedge \text{puppies}(\text{case})) \\ (\neg\, \text{dogs}(\text{case}) \wedge \text{puppies}(\text{case})) \end{pmatrix}$$

`SubstituteQuantors[`*expr*`,` *rules* | `List` *of rules*`]`

is a quick and dirty routine to replace quantors with one single instance so that one can apply propositional logic. The user must make sure that the replacement is correct, there is no formal test on that. Results[SubstituteQuantors] give some diagnostics: the collection of the bound variables in the ForAll and Exists respectively, the union of those variables, the intersection of those of Exists with those that are replaced. Best strategy is to substitute the bound variables of the Exists into the ForAll. But there can be list of variables, and conditions, so … it is at your risk

5.3.4.2 Critique from epistemology

The ghost of circularity is with us. In set theory we would define $M \subseteq P \Leftrightarrow \forall x : (x \in M \Rightarrow x \in P)$ so that entering this in the first line of a syllogism shows that we actually presume (a begging of the question) that if Socrates $\in M$ then definitely Socrates $\in P$. So that there is no *real* inference. That being said, it can be admitted that $p \Rightarrow p$ is a valid scheme. In other words, it might well be that Aristotle's original syllogisms *define* the words All and Some, and don't really reflect *inference*, the issue that we are interested in. In modern days we have better definitions for All and Some, so that the syllogism only remains interesting as a historical item.

It may thus well be that the whole epistemological value of the inference scheme is different. Aristotle's invention is a magnificent achievement but might perhaps not be the true model or the best model of human inference. Perhaps we should research more the issues of fuzzy sets, pattern recognition, intensive predicates, statistics and the like.

Let us take a more interesting example than "being human" and "being mortal". Let us consider two more specific propositions: (a) Socrates is at risk of dying from some illness. (b) Some people can be cured with medicine M but others then actually become more ill and die sooner. Now, the truth of those statements is not established by listing all elements who belong to the various sets. Everything is established with uncertainty. The diagnosis of the illness contains an element of probability - he might actually not be ill. The prognosis that determines whether the medicine will work must be based upon properties that are determined with some error. Well, this is an epistemologically much more relevant situation, and also seen from the point of view of logic, since there will be no possibility to beg the question. Possibly the doctors would reason "Socrates is old, and other old people reacted badly, so we better don't give him the medicine". In this case the inference is based upon similarity and not upon the full certainty that Socrates belongs to the set (that we defined with him already belonging to it).

That being said, the following: (1) The observation of the "begging of the question" is in itself correct. Inference can only show what there was already. The whole point is that the information is not yet accessible in the form that is useful. (2) The fuzzy replacement of "Socrates is old" with "Socrates is similar to other old people" just replaces the predicate "old" with "similar to other old people" so that the format remains the same. OK, there are differences but in this case there is value in noting the same structure too. (3) Perhaps also because of the fact that western civilization has been trained to think in terms of Aristotle's syllogism, his schemes reflect how people actually reason. (4) Yes, of course, at one stage we should continue with probability and statistics.

PM. Compare this with a situation of a witch doctor in an undeveloped land. He has it soft-wired in his brain that magic relies on similarities. His patients are convinced by

similarities too. As sickness comes along naturally with the release of bodily fluids, *hence* the witch doctor battles sickness by bleeding, *because* it is similar. The witch doctor has seen that they use injection needles in hospitals and he decides that these needles have great magical power. Thus he uses needles too and injects his patients with his own cocktails, *because* this is similar. Similarity can be dangerous. Let us first find out about certainty before we continue with fuzziness.

5.4 Taxonomy of inference

5.4.1 Proof and being decided

We have a whole list to look at: proven, refuted, non-sequitur, decided, undecided, untenable, provable, decidable, and so on. We can devour these in small chunks using the medieval square of contrariness.

Let S_0 be a specific (constant) set of statements. We may even call it a system but you might also think of a book. Let us consider a statement p that (i) can already be contained in the system, (ii) contradicts it, or (iii) might be added without contradiction. The following gives a review of the possibilities.

Let us first recall what we already saw in Chapter 2.

- This table uses the concepts of *contradiction* and *contrariness*. Instead of one statement we now use a list of statements S_0. Note that there are no quantifiers so that we (must) use constants. These concepts apply to a single conclusion *p.*

AffirmoNego[Ergo[S_0, p]]

Proven[p]				Refuted[p]
$(S_0 \vdash p)$	$\longleftrightarrow$	Contrary	$\longleftrightarrow$	$(S_0 \vdash \neg p)$
	$\nwarrow$		$\nearrow$	
$\Downarrow$		Not		$\Downarrow$
	$\swarrow$		$\searrow$	
$\left(S_0 \mathrel{\overline{\downarrow}} \neg p\right)$	$\longleftrightarrow$	Subcontrary	$\longleftrightarrow$	$\left(S_0 \mathrel{\overline{\downarrow}} p\right)$
NonSequitur[¬p]				NonSequitur[p]

We should recall the threesome $\left(P \vee \hat{P} \vee \tilde{P}\right)$. When S_0 does not prove or refute p then there is the third possibility $\tilde{P} = (S_0 \mathrel{\tilde{\vdash}} p) = \left(S_0 \mathrel{\overline{\downarrow}} p\right) \wedge \left(S_0 \mathrel{\overline{\downarrow}} \neg p\right)$. Let us call this "undecided".

- A system proves *p,* refutes *p,* or leaves it undecided.

Contrary[3, Ergo[S, p]]

$\left((S \vdash p) \bigvee (S \vdash \neg p) \bigvee (S \mathrel{\tilde{\vdash}} p)\right)$

A system may help us to decide upon a statement like p = "The sun rose over the Himalayas". The statement is *decided* within the system iff the system proves p or it proves $\neg p$. When the system proves both then p is untenable. When the system allows $(S \vdash p) \wedge (S \vdash \neg p) \Rightarrow (S \vdash p \wedge \neg p)$ then there is a proof of a contradiction. We use the term *contradiction* for $p \wedge \neg p$, and the term *untenable* for the p such that $(S \vdash p) \wedge (S \vdash \neg p)$.

- Contrariness of Undecided.

AffirmoNego[Undecided[S_0, p]]

$$
\begin{array}{lcccl}
\text{Undecided[p]} & & & & \text{Untenable[p]} \\
\left(\left(S_0 \underset{\downarrow}{\top} p\right) \bigwedge \left(S_0 \underset{\downarrow}{\top} \neg p\right)\right) & \longleftrightarrow & \text{Contrary} & \longleftrightarrow & ((S_0 \vdash p) \bigwedge (S_0 \vdash \neg p)) \\
 & \nwarrow & & \nearrow & \\
\Downarrow & & \text{Not} & & \Downarrow \\
 & \swarrow & & \searrow & \\
\left(\left(S_0 \underset{\downarrow}{\top} p\right) \bigvee \left(S_0 \underset{\downarrow}{\top} \neg p\right)\right) & \longleftrightarrow & \text{Subcontrary} & \longleftrightarrow & ((S_0 \vdash p) \bigvee (S_0 \vdash \neg p)) \\
\text{Tenable[p]} & & & & \text{Decided[p]}
\end{array}
$$

`Decided[`S`:, `y`___, `p`]`	Ergo[S, y, p] $\bigvee$ Ergo[S, y, Not[p]]
`Undecided[`S`:, `y`___, `p`]`	Not[Decided[(S, y,) p]]
`Tenable[`S`:, `y`___, `p`]`	Not[Untenable[(S,y,) p]]
`Untenable[`S`:, `y`___, `p`]`	Ergo[S, y, p] $\bigwedge$ Ergo[S, y, Not[p]]

To prevent a possible disappointment: these are just objects and do not evaluate the truth.

That some statement is undecided is an occasion that we encounter frequently in life. The expression above for undecided is a bit unwieldly. Given the importance of deduction for logic it might also be preferable to introduce a separate notation such as (S ♆ p) where we use Neptune's trident. We will call this trident "unresolved", and use it as purely equivalent with the other expressions. It is just a label, though.

Unresolved[Definition, S, p]

$$\left((S ♆ p) \;\Leftrightarrow\; (S \tilde{\vdash} p) \;\Leftrightarrow\; \left(\left(S \underset{\downarrow}{\top} p\right) \bigwedge \left(S \underset{\downarrow}{\top} \neg p\right)\right) \;\Leftrightarrow\; \neg ((S \vdash p) \bigvee (S \vdash \neg p))\right)$$

`Unresolved[`(S,) p]	shows as Neptune' s trident ♆ to reflect that the deduction is undecided. This is an alternative way to show Other[Ergo[S, p]], given that ~ might not read well for Ergo, especially in more complex statements, while it also may be recommendable that there is a separate symbol for the application of Other to Ergo. Unresolved does not evaluate, as neither Other[Ergo[…]] does. See Undecided[S, p]
`Unresolved[Definition,` (S,) p]	shows these definitions

`Unresolved[Rule, a, b]`

gives rules for replacement from a to b,
for a and b in {Unresolved, Other, Other[Ergo[##]]&, Unresolved},
where just Other can be used as shorthand for the proper more complex expression

`Independent[({S2 ___},) S ___, axiom]`

expresses that an axiom is independent of S when it is
not decided from S. Being an axiom means Ergo[(S2,) axiom]

5.4.2 Provable and decidable

S_0 is a specific system or statement. Let us now suppose that we have various systems, for example the subsets S of S_0, with S thus an element of the powerset of S_0. If S stands for a statement in a book then we can take paragraphs or chapters. If S_0 is a library then S can be a collection of books. When we use the phrase $\exists$ S then "proven" becomes "provable" and "decide" becomes "decidable".

Statement p is provable iff we find at least one S that supports it. Using quantifiers we now fully adopt the AIOE scheme. The opposites and contraries don't need to say much but it is good to have them seen once, if only because you might wonder whether they do say much.

- Contrariness for unprovable p.

AffirmoNego[Unprovable[S, p]]

Unprovable[p]				Axiomatic[p] or Inconsistent[S0]
$\forall_S \left(S ↧ p\right)$	$\longleftrightarrow$	Contrary	$\longleftrightarrow$	$\forall_S (S \vdash p)$
	$\nwarrow$		$\nearrow$	
$\Downarrow$		Not		$\Downarrow$
	$\swarrow$		$\searrow$	
$\exists_S \left(S ↧ p\right)$	$\longleftrightarrow$	Subcontrary	$\longleftrightarrow$	$\exists_S (S \vdash p)$
Some S are consistent for p				Provable[p]

A statement p is undecidable if there is no S in S_0 that decides it. Conversily it is decidable, so that p or $\neg\, p$ are provable. The result in the lower left corner might also be seen as independence though has been taken here as a consistent statement.

- Contrariness for undecidable *p*.

AffirmoNego[Undecidable[S, p]] /. Unresolved[Rule, Undecided, Unresolved]

				Axiomatic[p] or
Undecidable[p]				Inconsistent[S0]
$\forall_S\,(S$ ♆ $p)$	$\longleftrightarrow$	Contrary	$\longleftrightarrow$	$\forall_S\,((S \vdash p) \bigvee (S \vdash \neg p))$
	$\nwarrow$		$\nearrow$	
$\Downarrow$		Not		$\Downarrow$
	$\swarrow$		$\searrow$	
$\exists_S\,(S$ ♆ $p)$	$\longleftrightarrow$	Subcontrary	$\longleftrightarrow$	$\exists_S\,((S \vdash p) \bigvee (S \vdash \neg p))$
Some S are consistent for p				Decidable[p]

The squares for provability and decidability contain a statement in the upper right hand corner that shows that consistency issues pop up. If there is one inconsistent statement then the whole system should become inconsistent. The label given there is not quite consistency, since the quantifier runs over *S* while the concept of consistency has a quantifier running over *p*. This brings us to the next section.

Conventional textbooks tend to present these concepts as definitions with lists of proofs how they relate. The AIOE squares however seem a much better explanation how they are related. The concepts are now rather self-evident, and we might run a program to translate the one into the other.

`Decidable[(S, y___,) p]`	Exists[{S, y}, Decided[S, y, p]]
`Undecidable[(S, y___,) p]`	ForAll[{S, y}, Undecided[S, y, p]]
`Provable[(S, y___,) p]`	Exists[{S, y}, Ergo[S, y, p]]
`Unprovable[(S, y___,) p]`	Not[Provable[(statements,) p]]

To prevent a possible disappointment: these are just objects and do not evaluate the truth. Ergo is the label for Proven.

`ContraryLike[Last]`	a list of predicates for which ContraryLike only negates the last element (e.g. ForAll). Change the list by ContraryLike[Last] = {….}
`ContraryLike[f[p, …, q]]`	gives f[!p, …, !q] unless f is in ContraryLike[Last], in which case f[p, …, !q] (such as ForAll)

5.4.3 Consistency

We now switch focus, are less interested in individual conclusions, and wonder about all conclusions. This causes that the quantifier runs over p instead of S.

- A deductive system is inconsistent when it both proves and refutes a statement. Because of EFSQ, one inconsistency implies that all statements get proven.

AffirmoNego[Consistent[S_0, p]]

Consistent[S[0]]				Inconsistent[S[0]]
$\forall_p \left(\left(S_0 \mathrel{\top\!\!\!\downarrow} p\right) \bigvee \left(S_0 \mathrel{\top\!\!\!\downarrow} \neg p\right)\right)$	$\longleftrightarrow$	Contrary	$\longleftrightarrow$	$\forall_p\ ((S_0 \vdash p) \bigwedge (S_0 \vdash \neg p))$
	$\nwarrow$		$\nearrow$	
$\Downarrow$		Not		$\Downarrow$
	$\swarrow$		$\searrow$	
$\exists_p \left(\left(S_0 \mathrel{\top\!\!\!\downarrow} p\right) \bigvee \left(S_0 \mathrel{\top\!\!\!\downarrow} \neg p\right)\right)$	$\longleftrightarrow$	Subcontrary	$\longleftrightarrow$	$\exists_p\ ((S_0 \vdash p) \bigwedge (S_0 \vdash \neg p))$
Consistent[p]				Inconsistent for p but EFSQ implies inconsistent S[0]

We might use the Affirmo Nego Truthtable to show how the truthvalues are related. However, this time the table is not correct. If there is an inconsistency then we can derive *all* statements so that in this case O ⇒ E. In the same way, having a single consistent statement (neither it nor its opposite proven) implies that all are consistent, since if there was an inconsistency then you could prove anything.

- This table now is incorrect.

AffirmoNego[TruthTable, Consistent[S, p], Print → False]

	$\begin{pmatrix} A & E \\ I & O \end{pmatrix}$	T | F	A	I	O	E
$\forall_p \left(\left(S \mathrel{\top\!\!\!\downarrow} p\right) \bigvee \left(S \mathrel{\top\!\!\!\downarrow} \neg p\right)\right)$	A	True	True	1	c_0	0
$\exists_p \left(\left(S \mathrel{\top\!\!\!\downarrow} p\right) \bigvee \left(S \mathrel{\top\!\!\!\downarrow} \neg p\right)\right)$	I	True	U	True	U	c_0
$\exists_p\ ((S \vdash p) \bigwedge (S \vdash \neg p))$	O	True	c_0	U	True	U
$\forall_p\ ((S \vdash p) \bigwedge (S \vdash \neg p))$	E	True	0	c_0	1	True
$\forall_p \left(\left(S \mathrel{\top\!\!\!\downarrow} p\right) \bigvee \left(S \mathrel{\top\!\!\!\downarrow} \neg p\right)\right)$	A	False	False	U	c_1	U
$\exists_p \left(\left(S \mathrel{\top\!\!\!\downarrow} p\right) \bigvee \left(S \mathrel{\top\!\!\!\downarrow} \neg p\right)\right)$	I	False	0	False	1	c_1
$\exists_p\ ((S \vdash p) \bigwedge (S \vdash \neg p))$	O	False	c_1	1	False	0
$\forall_p\ ((S \vdash p) \bigwedge (S \vdash \neg p))$	E	False	U	c_1	U	False

`Consistent[` `S_:, y___, p_]`	ForAll[{p}, NonSequitur[S, y, p] || NonSequitur[S, y, Not[p]]]
`Inconsistent[` `S_:, y___, p_]`	Exists[{p}, Untenable[S, y, p]]

5.4.4 Decidability and consistency

It might be useful to define *strong decidability* with Xor. If we had defined normal decidability with Xor instead of Or then we wouldn't have required consistency, since this is implied by Xor. Considering just one variable, the statement $p \veebar \neg p$ excludes $p \wedge \neg p$ and thus is more accurate than $p \vee \neg p$. The battle cry *tertium non datur* however is translated with $p \vee \neg p$ merely because Or appears so useful for the binary situation of p and some q. The implied weakness for the concept of decidability (that now also includes the possibility that a system is inconsistent and thus proves anything) is corrected by also requiring consistency. The choice of concepts further is immaterial since all has to be established anyway.

5.4.5 Semantic interpretation

5.4.5.1 Relation of a system to the world

When we build deductive systems, we normally have an intended application. Pure mathematicians might do without, just going for the joy and esthetics of patterns, though perhaps such activity might be called an application too. The intended application is called the *semantic interpretation* of the system, since, next to the symbols and formal relations within the system, persons using those normally think about them in terms of their meaning and how they understand them for applications to the real world.

Thus when we have a system that allows us to arrive at conclusions ($S \vdash p$) then we may compare this to the world (the actual p). We saw this already with the definition of Truth, where Truth is a predicate, and we compared Truth[p] with p. We can do the same with a deductive system S. This S may be very general, such as for example a diagnostic test, like taking a blood sample to determine whether a patient has a disease or not. The test result basically derives from a deductive system based upon human knowledge. Using the threesome P, $\hat{P}$ and $\tilde{P}$ that we encountered we now may question how our deductions relate to the world. To do this we can construct a table with all the combinations of the states of the world with the possible decision states.

- The world is *always* represented in the columns and our decisions are *always* in the rows. It would be awkward if you don't do that. The reason for this presentational convention is that the leading opposites are easier recognized in columns. (Perhaps because of an ape-like preference for trees.) In the following table the core is given by columns of "Ergo, *p*, ¬*p*" while the first three columns support the explanation.

SemanticTable[All, Ergo[S, p]]

P	P̃	P̂	Ergo	p	$\neg p$
$(S \vdash p)$	$\neg(S \Psi p)$	$(S ↧ \neg p)$	$(S \vdash p)$	Right[1]	Error[2]
$(S ↧ p)$	$(S \Psi p)$	$(S ↧ \neg p)$	$(S \Psi p)$	Error[1] and Unresolved[1]	Error[2] and Unresolved[2]
$(S ↧ p)$	$\neg(S \Psi p)$	$(S \vdash \neg p)$	$(S \vdash \neg p)$	Error[1]	Right[2]

The table allows the following observations:

1. The system conforms to reality when (1) if *p* is the true situation, it decides on *p*, thus $(S \vdash p)$, *and*, (2) if ¬*p* is the true situation, it decides on ¬*p*, thus $(S \vdash \neg p)$. This is like with the Definition of Truth.
2. The system makes plain errors when (1) if *p* is the true situation, it decides on ¬*p*, thus $(S \vdash \neg p)$, *or*, (2) if ¬*p* is the true situation, it decides on *p*, thus $(S \vdash p)$.
3. The system leaves questions to ask when it cannot decide on the state of the world. The intended application fails because of undecidability.
4. Rejecting a decision, like (S ↧ p), still leaves two possible courses of action, either the contrary or the undecidedness.

a) To reject *p* while *p* actually is the case, is called the *error of the first kind.*

b) To reject ¬*p* while *p* actually is not the case, is called the *error of the second kind.* This can be phrased as "failing to reject a false null-hypothesis *p*".

From a formal point of view there is no difference between how we define *p* or ¬*p*. We might take *p* = "The sun shines" or *p* = "It is clouded". Therefore there is the convention that *p* should reflect the "status quo" (the "current situation" (or the one closest to it)). In this way the irresolution of the errors 1 and 2 get more meaning. When a surgeon has to decide on operating Mr. A or Ms. B, and *p* = "With an operation Mr. A has more chance to survive longer and better than Ms. B", then it makes a difference who gets the benefit of the doubt when that question cannot be fully settled. Medical ethics frequently is reduced to a *first come first served* base, so you better get into the hospital as quickly as possible.

`SemanticTable`[f[S___, p]]	gives a table with columns {p, $!p$} and rows {f[S, p], unresolved, contrary}
`SemanticTable[Number, f[`S`, `p`]]`	identifies the cells with numbers
`SemanticTable[Matrix, f[`S`, `p`]]`	just gives the matrix
`SemanticTable[Text, f[`S`, `p`]]`	
	explains the errors of the first and second kind. This is the only case where you might use option Matrix → {…} to give your own inner explanation
`SemanticTable[All, Ergo[`S`, `p`]]`	fuller explanation for Ergo (not available for other f)

5.4.5.2 Truth and provability

In the definition of truth (Truth[p] ⇔ p) we can distinguish Truth[p] ⇒ p and conversely p ⇒ Truth[p]. We can do the same with ($S \vdash p$) instead of Truth[p]. In the semantic table above, $p \Rightarrow (S \vdash p)$ takes the perspective from the columns, while $(S \vdash p) \Rightarrow p$ takes the perspective from the rows.

For a single p we call $(S \vdash p) \Rightarrow p$ that p is grounded. If it holds for all p then S is semantically correct.

For a single p we call $p \Rightarrow (S \vdash p)$ that p is lifted. If it holds for all p then S is deductively complete.

When the sets of statements of system S and its interpretation overlap and S is both semantically correct and deductively complete, then we have essentially reproduced the definition of truth, i.e. that p is true iff S proves it. In that case there are no undecidables in S and its interpretion any more. The other properties mentioned above, notably expressive completeness and consistency, may be seen as supportive properties, compared to this decomposition of the ⇔ in the definition of truth. PM. Categoricalness refers to other areas of application.

5.4.5.3 Semantical correctness or truthfulness

This section considers $(S \vdash p) \Rightarrow p$. A single p is called semantically grounded (for system S) when proof (in S) implies that p is true. Truth can of course only be established for an (intended) implementation. If this holds for all p under an intended interpretation then system S itself is called *semantically correct* for that interpretation. This term may be a bit pedantic but it is the one used in the literature. A better English word is "truthful", as we regard a person as truthful when, if he or she asserts something, we can rely on it being true. A more standard term is "scientific". When a formal system S is applied to a body of scientific knowledge then the theorems of S would also be true in that science.

- Generally we use this simple form. Note that $\forall\, p$ also covers instances of $\neg p$.

  SemanticallyCorrect[S, p]

  $\forall_p\, ((S \vdash p) \Rightarrow p)$

Normally the intended application of S is clear. For example in logic, an axiomatic system could be intended to cover propositional logic. Or an axiomatic system could be intended to cover the use of sets. When there is no clear interpretation then the "for all" quantifiier may not have a domain and there is the risk that we are not guided by our knowledge of a practical situation. In that case it helps to include the condition that the "for all" refers to all statements in S that must also be interpreted.

- The full definition of semantically correctness is:

  SemanticallyCorrect[{Interpretation}, S, p]

  $\forall_{p, p \in S}\, (p \in \text{Interpretation} \bigwedge ((S \vdash p) \Rightarrow p))$

- When $\forall\, p$: $(S \vdash p) \Rightarrow p$ holds then we can deduce that a proof for p means that there is no proof for $\neg p$. When we give a model in reality, then the system will be consistent (recall that definition).

  Ergo2D[Grounded[S, p], Grounded[S, ¬ p], CounterImplies[Grounded[S, ¬ p]], Ergo[S, p] ⇒ NonSequitur[S, ¬ p]]

  1 $((S \vdash p) \Rightarrow p)$
  2 $((S \vdash \neg p) \Rightarrow \neg p)$
  3 $\left(p \Rightarrow \left(S \underset{\downarrow}{\top} \neg p\right)\right)$
  Ergo ————————
  4 $\left((S \vdash p) \Rightarrow \left(S \underset{\downarrow}{\top} \neg p\right)\right)$

  Consistent[S, p]

  $\forall_p \left(\left(S \underset{\downarrow}{\top} p\right) \bigvee \left(S \underset{\downarrow}{\top} \neg p\right)\right)$

When $\forall\, p$: $(S \vdash p) \Rightarrow p$ holds then we can deduce the following "inverse truthtable", where we now assign truthvalues to the decision states and then see what this implies for reality. It $(S \vdash p)$ then p and there can't be a proof for $\neg p$, since from $(S \vdash \neg p)$ it would follow that $\neg p$, whence would follow a contradiction. Similarly for $(S \vdash \neg p)$. Finally, it can be that p is undecided and for whole S undecidable, and in that case $p \vee \neg p$. Note that the left hand side $(S \vdash p) \vee (S \underset{\downarrow}{\top} p) \vee (S \vdash \neg p)$ is equivalent to the right hand side $p \vee \neg p$ so that we can read the table both ways.

TruthTable[SemanticallyCorrect, S, p]

$(S \vdash p)$	$(S \Psi p)$	$(S \vdash \neg p)$	p	$\neg p$
1	0	0	1	0
0	1	0	1	0
0	1	0	0	1
0	0	1	0	1

- This table derives from that both p and $\neg p$ are grounded.

Grounded[S, p] && Grounded[S, ¬ p] // LogicalExpand

$$\left(\left(p \bigwedge \left(S ↧ \neg p\right)\right) \bigvee \left(\neg p \bigwedge \left(S ↧ p\right)\right) \bigvee \left(\left(S ↧ p\right) \bigwedge \left(S ↧ \neg p\right)\right)\right)$$

- Presenting the parts in the Or statement on separate lines makes for easier reading.

Grounded[S, p] && Grounded[S, ¬ p] // MatrixDNForm

$$\begin{pmatrix} \left(p \bigwedge \left(S ↧ \neg p\right)\right) \\ \left(\neg p \bigwedge \left(S ↧ p\right)\right) \\ \left(\left(S ↧ p\right) \bigwedge \left(S ↧ \neg p\right)\right) \end{pmatrix}$$

PM. There is a pitfall that we should not fall in. Consider the statement that some p is undecided. If we interprete the $\forall\, p{:}\;\; (S \vdash p) \Rightarrow p$ as the possibility in *Mathematica* to do replacements, then undecidedness causes a contradiction (i.e. in combination with this interpretation of semantical completeness).

- This is how a contradiction would be caused.

Undecided[S, p]

$$\left(\left(S ↧ p\right) \bigwedge \left(S ↧ \neg p\right)\right)$$

% /. Ergo[S, p_] :→ p

$(\neg p \wedge p)$

% // LogicalExpand

False

The answer to this is that $\forall\, p{:}\;\; (S \vdash p) \Rightarrow p$ is not quite a scheme for replacement. In particular, such a replacement rule apparently replaces $\neg\,(S \vdash p)$ with $\neg p$. The error is that this assumes equivalence ($\Leftrightarrow$) instead of just implication. You cannot replace both $(S \vdash p) \rightarrow p$ and $(S \vdash \neg p) \rightarrow \neg p$ when it holds that $((S \vdash p) \Rightarrow (S ↧ \neg p))$. So if you substitute an inconsistency then you should not be surprised that one turns up.

`Grounded[`S`:, `y`___, `p`]`	is `Ergo`[S, y, p] ⇒ p
`SemanticallyCorrect[`S`:, `y`___, `p`]`	
	expresses that for all p: Grounded[S, y, p]
`SemanticallyCorrect[{`$S2$`, `$y2$`___}, `S`:, `y`___, `p`]`	
	states this for all p in {S, y} with the now clarified condition that p must also be in the intended interpretation {S2, y2}. Thus {S, y} must also be a subset of {S2, y2}
`TruthTable[SemanticallyCorrect, `S`, `p`]`	
	gives the truthtable when Ergo[S, p] ⇒ p holds

5.4.5.4 Deductive completeness

This section considers $p \Rightarrow (S \vdash p)$. System S is deductively capable for some p iff, when p is true in the intended interpretation S can prove it. As the opposite to the above "grounded" we can use the term "lifted" here. When all p are lifted then system S is called *deductively complete*.

- Again we will prefer the first expression that is simpler.

DeductivelyComplete[S, p]

$\forall_p \, (p \Rightarrow (S \vdash p))$

DeductivelyComplete[{Interpretation}, S, p]

$\forall_{p, p \in \text{Interpretation}} \, (p \in S \bigwedge (p \Rightarrow (S \vdash p)))$

- In a deductively complete system both the statement and its opposite have been lifted so that it is no longer undecided.

Ergo2D[Lifted[S, p], Lifted[S, ¬p], p ∨ ¬p, Decided[S, p]]

1	$(p \Rightarrow (S \vdash p))$
2	$(\neg p \Rightarrow (S \vdash \neg p))$
3	$(p \bigvee \neg p)$
Ergo	———
4	$((S \vdash p) \bigvee (S \vdash \neg p))$

Hence:

ForAll[p, Decided[S, p]]

$\forall_p \, ((S \vdash p) \bigvee (S \vdash \neg p))$

Deductive completeness not merely implies that all statements are decidable but it also excludes a jumbled-up system (where all statements are decidable but some in mirror

form):

ForAll[p, Decided[S, p]] ⋀ Exists[p, p ⇒ Ergo[S, ¬ p]]

$(\forall_p ((S \vdash p) \vee (S \vdash \neg p)) \wedge \exists_p (p \Rightarrow (S \vdash \neg p)))$

The following truth table basically is not needed since it is the same as that of $p \vee \neg p$. Just when p is a nonsensical statement such that it has an indeterminate truthvalue, then deductive completeness allows that the system derives an inconsistency. To express this, we need three-valued logic.

TruthTable[DeductivelyComplete, S, p]

p	$\neg p$	$(S \vdash p)$	$(S \vdash \neg p)$	$((S \vdash p) \wedge (S \vdash \neg p))$
1	0	1	0	0
Indeterminate	Indeterminate	1	1	1
0	1	0	1	0

- These are the possible situations when both p and $\neg p$ are lifted. Parts in the Or statement are on separate lines.

Lifted[S, p] && Lifted[S, ¬ p] // MatrixDNForm

$$\begin{pmatrix} (p \wedge (S \vdash p)) \\ ((S \vdash p) \wedge (S \vdash \neg p)) \\ ((S \vdash \neg p) \wedge \neg p) \end{pmatrix}$$

`DeductivelyComplete[`*S*`:, `*y*`___, `*p*`]`	expresses that all *p* that are true under the intended interpretation are theorems
`DeductivelyComplete[{`*S2*`, `*y2*`___}, `*S*`:, `*y*`___, `*p*`]`	clarifies that this actually means that for all *p* in the interpretation {S2, y2} it must hold that they are also in {S, y} and can be deduced
`Lifted[`*S*`:, `*y*`___, `*p*`]`	is *p* ⇒ Ergo[*S*, *y*, *p*]

5.5 Inference with the axiomatic method

5.5.1 Introduction

This section investigates rule based inference for propositional logic. **Infer** is a large array of rules that has been composed automatically by InferenceMachine. Repeated replacement using Infer gives a logical conclusion. The method still suffers from the general weakness of the axiomatic method, that any truth can be derived from false premisses.

Economics[Inference, Print → False]

5.5.2 Pattern recognition

We now investigate the axiomatic method. Some relations are posed as axioms, and conjectures may be proven by repeated application of these axioms. The truthtable method that we discussed above is strong for cases where one has a definite idea about the relation to be proven. The axiomatic method on the other hand has the advantage that forms may be created that one had not thought of before.

In the `Logic`` package we already encoutered the set of rules **AndOrRules**. Here we meet **Infer**, a large array of rules that has been composed automatically by InferenceMachine.

Though *Mathematica* is a rule based program by itself and has remarkable capabilities in pattern recognition, it still is not as simple as it might seem to develop an inference machine. The problem resides in the fact that pattern recognition remains a complicated affair.

For example it turns out that Not[p] is not always recognised as a proposition on a par with p.

- When we have an axiom (p_ ⇒ !p_) :> !p, then *p* should stand for any *q* and thus also ¬ *q:* but replacement does not recognize that.

axiom = (p_ ⇒ !p_) :> !p

(p_ ⇒ ¬ p_) :→ ¬ *p*

q ⇒ !q /. axiom

¬ *q*

(!q ⇒ q) /. axiom

(¬ *q* ⇒ *q*)

Similar pattern recognition problems seem to exist for other basic properties, like the antisymmetry of If.

Given this problem with pattern recognition, we best distinguish between (1) the axioms proper, i.e. the axioms as we wish them to hold, as expressed by a replacement pattern, and (2) the metarules, that allow *Mathematica* to recognise and apply the axioms.

With respect to these metarules, we still have levels of complexity. The first level of metarules in pattern recognition is given for example by patterns defined by hand, as is the case for AndOrRules or SelfImplication. A second level of metarules arises when we use properties like symmetry and embedding within BlankNullSequences, and use brute force by having *Mathematica* generate all combinations. For straightforward patterns both approaches may come down to the same. Of course it is most elegant - a third level - when a general pattern can be defined, for example that Not[p] comes on a

par with p, but when the developer has not yet solved his pattern problem, brute force may be a good alternative.

The InferenceMachine provides some brute force, and allows you to formulate your own rules and to expand them.

A problem that is inherent in the axiomatic method is that any truth can be derived from false premisses. In Latin, this is the Ex Falso Sequitur Quodlibet situation. This EFSQ problem is not solved here.

5.5.3 Infer

Infer is composed by InferenceMachine when the `Inference`` package is loaded by *Mathematica*. Infer is standardly based on the axioms SelfImplication, ModusPonens, EFSQ, IfTransitive, and IfToAnd, while using metarules AndSymmetric, OrSymmetric and IfAntiSymmetric.

`x //. Infer`	tries to solve a logical statement x
`InferQ`	gives suggestions on the use of the package
`InferAndOr`	AndOrRules ~Join~ Infer

InferQ

The ⇒ will here have the function of implication within the object language.
On terms: Object language constants are P[1], P[2], ... and variables are p1, p2, ... Commonly, the term 'axiom' is used for an (object language) expression like p1 ⇒ p1 which one accepts as true, for variable p1. In other words (p1 ⇒ p1) :> True (concludes to True) for every substitution of a constant. Commonly, next, there are rules that allow manipulation of those axioms. Here however, those replacement rules are much more in focus. It becomes rather natural to use the term 'axiom' for p_ ⇒ p_ :> True. Our somewhat deviant use of terms is likely caused by the fact that patterns and variables are different.
a) To allow better manipulation of patterns, define axioms in terms of symbols p, q, ... Of course, use logical operators And, Or, Xor, Not, ⇒ (Implies), True and False. Use RuleDelayed for the result. E.g. (p ⇒ p) :> True.
You can use patterns, so that your axioms are replacement rules. Note that (p_ ⇒ p_) gives a warning message. Then use LogicalPattern[p ⇒ p]. Perhaps, for the distinction with the metarules, you don't want to use patterns for the axioms, and have these automatically inserted later.
Note: since you would like to use repeated replacement, your axioms should simplify rather than expand.
b) Give metarules (e.g. on symmetry) with normal patterns x_, y_ , ... in the premisses. These metarules must be single, named and in MetaRuleForm.
c) Each axiom can be inputted in ExpandAxiom with the metarules for its operators. Options allow replacement of symbols p, q, ... with patterns (in the premiss), and If with Rule.
d) Joining these expansions gives YourSetOfRules.
e) Analysis of object language logical expressions like x = (p1 ⇒ q1) works like this:
x /. YourSetOfRules or x //. YourSetOfRules
f) The default inference rule is Infer. It presumes AndOrEnhancement. Otherwise use InferAndOr.
g) Infer has a preference for both using ⇒ and solving by substitution.

- The situation that we want to represent is:

Ergo2D[(p ⇒ q), (q ⇒ r), p, "find this"]

1	$(p \Rightarrow q)$
2	$(q \Rightarrow r)$
3	p
Ergo	———
4	find this

- A simple success is:

(p ⇒ q) ∧ (q ⇒ r) ∧ p //. Infer

r

- Perhaps more complex is:

(!r ⇒ !q) ∧ (!r ⇒ p) ∧ (p ⇒ q) //. Infer

r

- By comparison, Simplify correctly collects all information. But this does not represent the situation that we want to represent.

(p ⇒ q) ∧ (q ⇒ r) ∧ p // Simplify

$(p \wedge q \wedge r)$

5.5.4 Axioms and metarules

In mathematics, all axioms are treated the same. *Mathematica* forces us to be more specific. Metarules will here be rules that will be applied to the axioms in order to increase the likelihood that the axiomatic pattern is recognised.

`ModusPonens`	axiom	((p_ ⇒ q_) && p_) :> q
`IfToIf`	axiom	(p_ ⇒ (q_ ⇒ m_)) :> ((p ⇒ q) ⇒ (p ⇒ m))
`IfToAnd`	axiom	((p_ ⇒ q_) ⇒ m_) :> ((!p ⇒ m) && (q ⇒ m))
`EFSQ`	axiom	(!p_ ⇒ (p_ ⇒ q_)) :> True
`IfTransitive`	axiom	((p_ ⇒ q_) && (q_ ⇒ r_)) :> (p ⇒ r)
`SelfImplication`	axiom	(!p_ ⇒ p_) :> p and (p_ ⇒ p_) :> True

Note: EFSQ = Ex Falso Sequitur Quodlibet: From False follows anything you want.

`AndSymmetric`	metarule	(x_ && y_) :> (y && x)
`OrSymmetric`	metarule	(x_ || y_) :> (y || x)
`IfAntiSymmetric`	metarule	(x_ ⇒ y_) :> (!y ⇒ !x)

The following routines are used to get Not[p] on a par with p:

`Deny[x]`	gives a rule for denial of x: x → Not[x]; x can be a list
`DenyPattern[pattern, x, prin:True]`	
replaces x_ with ! (x_) within pattern and x with ! x (for conclusions). Default prints	

5.5.5 InferenceMachine

The routine that puts everything together is InferenceMachine. By 'expanding' an axiom we will mean the application of the metarules so that an axiom is replicated in all its patterns.

`InferenceMachine[opts]`	expands the axioms using the metarules, creating a list of replacement rules

InferenceMachine is directed by various options. The input options are: HoldAll → {axioms taken as they are}, Hold → {axioms that already conclude to True; expandable}, AxiomToTrue → {axioms that don't conclude to True, and you want to add 'axiom :> True'; expandable}, ExpandAxiom → {axioms which will be expanded}, PlacedProperties → {And → {…}, If → {…}, …} are the metarules (entered as Strings so that they will not evaluate at the call) that should hold for any instance of the operators.

- An example is the definition of Infer at start up:

```
Infer = InferenceMachine[
        HoldAll → {SelfImplication},
   Hold → {EFSQ},
   AxiomToTrue → {ModusPonens, IfTransitive, IfToAnd},
   ExpandAxiom → {ModusPonens, IfTransitive, IfToAnd},
   PlacedProperties →
        {Or → "OrSymmetric", And → "AndSymmetric",
        Implies → "IfAntiSymmetric"}];
```

5.5.6 The axiomatic method and EFSQ

It turns out that Infer is not without a paradox. We can trace this paradox to the Ex Falso Sequitur Quodlibet situation, or, that from a falsehood anything can be derived. It is useful to be aware of this, for otherwise we might conclude that there is an error in our InferenceMachine. The following is a crucial example.

- The following applies the transitivity of If:

```
(x ⇒ !x) ∧ (!x ⇒ x) /. Infer
```

$(x \Rightarrow x)$

% /. Infer

True

- While SelfImplication gives (now using //. rather than /.):

(x ⇒ !x) ∧ (!x ⇒ x) //. SelfImplication

$(\neg x \land x)$

% //. InferAndOr

False

The analysis of the situation is as follows. The proper answer is provided by SelfImplication, and the statement is False. Given that there is a falsehood, anything can be derived, such as the results True (the first approach).

Above problem seems to be related to the fact that IfTransitive dominates SelfImplication. But the true problem is of a more general nature. If we would change the order, we would run into another case where EFSQ causes problems.

It thus turns out that Infer is rather risky for application to real world questions. It may generate True, by *valid* deduction, while the proper answer is False. For this reason there has been no effort to extend Decide[ ] with Infer, and to create an InferEnhance[ ] mode as with AndOrEnhance[ ]. This does not seem too bad, as one can always fall back on the method with the truth tables.

In fact, if one wishes to save the axiomatic method, then the proper approach would seem to be to have various deductions parallel to each other. Eventually, this amounts to the same as the truthtable method.

5.5.7 Expansion subroutines

AndEmbedding[*r*]	embeds p_ && q_ within BlankNullSequences, while the pattern−names are added to the last part (assumed to be the conclusion)
AxiomToTrue[*x*, *y*:Implies, *adjustable*:True]	
	gives (x /. RuleDelayed → y) :> True. Constructs the rule: whole axiom :> True. If adjustable, then the conclusion is patterned if the premiss was
ExpandAxiom[*axiom*, *opts*]	
	applies the metarules for Implies, And, Or & Xor, for all separate occurrences of these operators. The metarules must be given as operator → name or operator → {name1, name2, …} where namei are Strings. ($namei will be the copy for the occurrences.)

Default options for ExpandAxiom are: MachineForm → True applies MachineForm to the axiom first, First → True means that the metarules only apply to the premiss, LogicalPattern → True gives patterns in output, Rule → True replaces Implies with Rule, AndEmbedding → True embeds p && q within BlankNullSequences, Union → True flattens the result and thereby removes possible duplications. Possible options are default rules for Implies, And, Or & Xor.

5.5.8 Different forms

The analysis gives rise to various forms, namely MachineForm, ObjectForm and MetaRuleForm. Actually, Rule and Pattern are not explicitly called 'Form', but still are forms of a logical statement.

`MachineForm[x]`	does UnPattern[x] /. Rule → Implies. Axioms are internally set to MachineForm, to facilitate matches
`ObjectForm[x]`	does UnPattern[x] /. RuleDelayed → Rule. Transforms into object language form, including :>
`MetaRuleForm[x]`	does LogicalPattern[MachineForm[x]]
`ToRule[x]`	does LogicalPattern[x] /. Implies → Rule. Transforms object language expressions into rules
`LogicalPattern[x]`	changes a logical expression x into a logical pattern. Default option First → True applies pattern only to the premiss

5.5.9 Accounting

The InferenceMachine works by assigning a marker to each occurrence of a logical operator. The metarules then are applied to each operator individually. After this, the markers are removed again.

`PlacedMetaRule[` *name_String* `]`	subroutine for ExpandAxiom and PlaceProperties, derives a placed metarule from the general metarule
`PlacedProperties[`*len*`, `*x*`]`	subroutine for ExpandAxiom, gives each placed operator its own list of metarules; len is the number of occurrences of the operator, and x gives the list of general metarules for that operator
`Place[`*Operator*`, `*i*`]`	marks the place of the ith operator
`PlaceOperator[`*expr*`]`	gives each operator it place mark
`PlaceUndo`	= {*Place*[*x*_, *y*_] → *x*} which removes Place marks

5.6 A note on inference

The similarity of "," and "∧" in respectively $\{p_1, \ldots, p_n\} \vdash q$ and $(p_1 \wedge \ldots \wedge p_n) \Rightarrow q$ is innocuous but we can also observe a similarity between ⊢ and ⇒ that is more unsettling.

- Earlier we wrote:

  Ergo$[(P[1] \Rightarrow P[2])_t,\ P[1]_{t+1},\ P[2]_{t+3}]$

  $(\{(P_1 \Rightarrow P_2)_t, (P_1)_{t+1}\} \vdash (P_2)_{t+3})$

- But it is a more tantalizing to write the following since it clarifies that "⊢" can also be formalized with $\Rightarrow_t$ so that it is only a small leap of thought to abstract from t.

  Implies$[(P[1] \Rightarrow P[2])_t \wedge P[1]_{t+1},\ P[2]_{t+3}]$

  $(((P_1 \Rightarrow P_2)_t \wedge (P_1)_{t+1}) \Rightarrow (P_2)_{t+3})$

In other words, the difference between a statement and an argument seems only to be how the ∧ and ⇒ symbols are read and interpreted. Reading statically gives a statement, reading dynamically gives an argument.

The dynamic notation might also help to formalize hypothetical reasoning, i.e. when one supposes something but later has to reject it as false. In standard inference notation that becomes $(p \Rightarrow \neg p) \vdash \neg p$ yet in practice texts read as "p_t ; $(\neg\, p)_{t+1}$; thus $(\neg\, p)_{t+2}$" where the word "thus" only has the psychological value of emphasis.

Having established the difference between implication and inference, the question arises whether a further investigation in inference is useful. It seems that most characteristics of inference can be captured by the study of statements. The dynamic aspect of inference does not add anything to its validity (or tautological character of the associated statement). By describing inference, logic has made room for the very human activity of

deduction, but the application of "⊢" itself does not seem to lead to general results that cannot be described otherwise.

Thus, one is left with a feeling of unease whether the difference (and even distinction) between Decide (⊢) and Implies (⇒), shorter versus longer expressions, is really that relevant (except for psychology).

- PM. It appears that *Mathematica* also has a symbol Therefore, that does not evaluate but shows with three dots. It is conceivable to restrict the use of $S \vdash p$ to "system S proves p" and then this $S \therefore p$ to Socrates, i.e. human inference. But given the structural identity this causes a multitude of symbols without really adding much, so we skip this.

Therefore[p, p ∧ q]

$p \therefore (p \wedge q)$

Because[p ∨ q, p ∧ q]

$(p \vee q) \because (p \wedge q)$

6. Applications

6.1 Introduction

The issues of morals, knowledge, probability and modal logic are applications of both predicates and inference.

In these areas of application the notion of contrariness is much used. The following reviews these concepts, where the term “deontic logic” is used for the logic of morality.

Statement [p]	**Deontic**	**Knowledge**	**Probability**	**Modal logic**
P	Ought p	Know p	Certain p	Necessary p
$\tilde{P}$	Freedom p	Ignorance p	Uncertain p	Contingent p
$\hat{P}$	NotAllowed p	Know ¬ p	Certain ¬ p	Impossible p

Interesting to observe is how these concepts relate, and also that the p might belong to the domain of mathematics or empirics. Mathematical truths are called necessarily true. The state of knowledge about an empirical state might be compared to the decision state of a formal system. Probability generally concerns empirical states, but when an event is 100% certain then some might call it necessary. A 0% probability for p means that $\neg p$ is 100% certain. Uncertainty is the key notion, since when you are “not uncertain” about p, then it still might be p or $\neg p$. Certainty may be qualified as to what it is that is certain. Events that are 100% certain might also be called necessary.

A final application that must be mentioned here is the application of logic and inference to larger issues in the world.

6.2 Morals and deontic logic

6.2.1 Introduction

Morals have the same structural form as Preferences:

Preferences	**Morals**
Better	Ought
Indifference	Freedom
Worse	Not Allowed

Morals and preferences are an example application of the predicate calculus. Morals are preferences in a strong form, such that people are unwilling to consider other aspects before some principles have been accepted first. Such an ordering is also called *lexicographic* - taken from the analogy of a dictionary where words are ordered such that for example a *p* is always before a *u*.

The prime subject of the theory of morals is that there is a gap between Is and Ought. This principle is not self-evident. People tend to confuse reality with what should be. Once you are aware of the distinction, it seems pretty obvious - yet confusion creeps up at unexpected moments anyway. Some countries in the world for example have a death penalty, and the citizens of those states are used to the idea - which may cause some of them to think that this is how it *should* be. But a 'should' can never be derived from an 'is'.

The logic of morals is called deontic logic. The most important axiom is that if something is morally imperative, then also all its implications are morally imperative. If a person drowns, and if accidental deaths ought to be prevented, then we should try to save that person.

Economics[Logic`Deontic]

Cool`Logic`Deontic`

Allowed	FreedomQ	NotAllowedQ	SetDeontic	ToOught
AllowedQ	MoralConclude	NotOught	ToAllowed	
DeonticAxiom	MoralSelect	Ought	ToFreedom	
Freedom	NotAllowed	OughtQ	ToNotAllowed	

SetDeontic["Explain"]

SetDeontic[u, o, na] symplifies the following steps:
The user has to set
  Universe[] = {p, ¬p, q, ...} where each p has a ¬p (not p)
  Ought[] = Ought[{...}] with a selection from the universe, for the Op
  NotAllowed[] = NotAllowed[{...}] with another selection, for the ¬Ap
Then ToAllowed[] and ToFreedom[] give what is allowed and what is free to choose
The crucial idea is that Op ⇔ ¬A¬p.
The universe consists of three disjoint sets: Ought, Freedom and NotAllowed. The
  Universe, Allowed and Freedom objects read as Or[ ], the Ought and NotAllowed objects read
  as And[ ]. Ought, Freedom and NotAllowed may also be seen as Better, Indifferent and Worse.
SetDeontic[Universe] creates the universe from the binary states, and selects the Ought[Universe] cases

6.2.2 Setting values manually

By first setting some values manually, we will better understand the components.

- Required are some undeclared Symbols. Each represents some statement, like p = "This person drowns", q = "I help".

symbs = {p, q, r, s, t, v}

$\{p, q, r, s, t, v\}$

- The elements of the universe should also contain the negations - like $\neg p$ = "This person does not drown".

u = Universe[] = Flatten[FromEvent /@ symbs]

$\{p, \neg p, q, \neg q, r, \neg r, s, \neg s, t, \neg t, v, \neg v\}$

- $O\neg p$ means "This person should not drown". Let us also take Or for some r.

o = Ought[] = Ought[{p, r}]

Ought($\{p, r\}$)

- Let us declare that t and v are not allowed: $\neg At$ & $\neg Av$.

na = NotAllowed[] = NotAllowed[{t, v}]

NotAllowed($\{t, v\}$)

The key concept is $O(\neg p) \Leftrightarrow \neg Ap$. For example: You should not smoke $\Leftrightarrow$ It is not allowed that you smoke. (An ethical principle is stronger than a health warning !)

- It turns out that we did not properly state what ought to happen. We forgot $\neg t$ and $\neg v$.

too = ToOught[na]

Ought($\{\neg t, \neg v\}$)

- And neither were we specific on what is not allowed. We forgot $\neg p$ and $\neg r$.

ToNotAllowed[o]

NotAllowed($\{\neg p, \neg r\}$)

`ToAllowed[]`	derives what is allowed from what is not allowed
`ToFreedom[]`	derives what is subject to free choice from Ought[] and NotAllowed[]
`ToNotAllowed[`*x_Ought*`]`	derives what is not allowed if x Ought
`ToOught[`*x_NotAllowed*`]`	derives what Ought if x is NotAllowed

Note that only the last two require an input. They must be called before the first two can be called.

6.2.3 Using SetDeontic

The routine SetDeontic helps us to consistently define the realms of the discussion. Hence, properly redefining Ought and NotAllowed.

SetDeontic[symbs, {p, r}, {t, v}]

$\{\{p, \neg p, q, \neg q, r, \neg r, s, \neg s, t, \neg t, v, \neg v\}$, Ought($\{p, r, \neg t, \neg v\}$),
NotAllowed($\{\neg p, \neg r, t, v\}$), Allowed($\{p, q, r, s, \neg q, \neg s, \neg t, \neg v\}$), Freedom($\{q, s, \neg q, \neg s\}$)$\}$

`SetDeontic[`*U_List*`,` *O_List*`,` *NA_List*`]`	
	The universe elements are defined as the elements in U and their negations. What ought is defined as the elements in O and the negations in NA. What is NotAllowed is defined from the elements in NA and the negations of O
`SetDeontic[Universe]`	
	sets Universe[Universe] to the outer product of {p, ¬p} for the elements in U, and sets Ought[Universe] to the list of possibilities that satisfy what ought

SetDeontic has also defined the objects Allowed and Freedom.

- Allowed is what is not NotAllowed. What ought, is also allowed. (It would be strange to say "You ought to help, but you are not allowed to help.")

Allowed[]

Allowed($\{p, q, r, s, \neg q, \neg s, \neg t, \neg v\}$)

- Freedom exists where we are allowed to do things that we do not ought to do.

Freedom[]

Freedom($\{q, s, \neg q, \neg s\}$)

6.2.4 Objects and Q's with the same structure

`Allowed[]`	should refer to an Allowed[{...}] object
`Allowed[{...}]`	is the object that contains what is allowed
`AllowedQ[p]`	is True iff p is an element of Allowed[]
`AllowedQ[p_List]`	is True iff all elements in p are in Allowed[]
`AllowedQ[Universe, p_List]`	is the same as AllowedQ

`Freedom[]`	should refer to a Freedom[{...}] object
`Freedom[{...}]`	is the object that contains what is free to choose from
`FreedomQ[p]`	is True iff p is an element of Freedom[]
`FreedomQ[p_List]`	is True iff all elements in p are in Freedom[]
`FreedomQ[Universe, p_List]`	is True iff all elements in p that are not– ought are in Freedom[]

`NotAllowed[]`	should refer to a NotAllowed[{...}] object
`NotAllowed[{...}]`	is the object that contains what is not allowed
`NotAllowedQ[p]`	is True iff p is an element of NotAllowed[]
`NotAllowedQ[p_List]`	is True iff all elements in p are in NotAllowed[]
`NotAllowedQ[Universe, p_List]`	is True iff some elements in NotAllowed[] also occur in p

`Ought[]`	should refer to an Ought[{...}] object
`Ought[{...}]`	is the object that contains what ought
`OughtQ[p]`	is True iff p is an element of Ought[]
`OughtQ[p_List]`	is True iff all elements in p are in Ought[]
`OughtQ[Universe, p_List]`	is True iff all elements in Ought[] also occur in p

Note: Also defined has been Not-Ought, since sometimes there is linguistic confusion with Ought-Not (when people want to emphasise something, for example). NotOught ($\neg O$) = Freedom *or* NotAllowed (just the complement).

NotOught[]

NotOught($\{q, s, t, v, \neg p, \neg q, \neg r, \neg s\}$)

Freedom[] || NotAllowed[]

(Freedom($\{q, s, \neg q, \neg s\}$) $\vee$ NotAllowed($\{\neg p, \neg r, t, v\}$))

`NotOught[]`	derives for which it is not said that it ought (Freedom or NotAllowed)
`NotOught[{...}]`	is the object that contains what not ought

Other functions for NotOught are not available.

6.2.5 Universe

Above gives just the elements of the universe. The real universe is a logical combination of some if its elements. Possible states of the world are for example *p & q & r*, but also ¬ *p &* ¬ *q & r*. Given our elements, we must take all possible combinations of {*p*, ¬*p*}, {*q*, ¬*q*}, etcetera. Rather than using the symbol '*&*' we will use lists. Thus a list {*p*, ¬*q*, *r*} is the same as the assertion that *p &* ¬*q & r*, with all these phenomena occuring at the same time. The universe of all such possible combinations is Universe[Universe]. SetDeontic[Universe] will create this universe. However, mainly interesting is Ought[Universe] that gives the list of possible states that satisfy what ought. The latter hence is also put out by SetDeontic[Universe].

- This gives the possible combinations that satisfy what ought.

SetDeontic[Universe]

$$\begin{pmatrix} p & q & r & s & \neg t & \neg v \\ p & q & r & \neg s & \neg t & \neg v \\ p & \neg q & r & s & \neg t & \neg v \\ p & \neg q & r & \neg s & \neg t & \neg v \end{pmatrix}$$

`MoralSelect[`*lis_List*`?MatrixQ, `*q*`]`	
	selects from the matrix using criterion *q*. The latter must be defined for q[Universe, …] – which is the case for *q* = AllowedQ, FreedomQ, NotAllowedQ and OughtQ
`MoralSelect[`*q*`]`	uses Universe[Universe], and for q = OughtQ it gives Ought[Universe]

Note that the q[Universe, ...] criteria have different meanings for elements or a state of the universe.

6.2.6 The difference between Is and Ought

Above we took *p* = "This person drowns", *q* = "I help". Above universe suggests that it still would be allowed that a person drowns but is not helped. The deontic axiom however suggests: If someone is drowning and can probably be saved by helping, and if you consider that this person should not drown, then you should save him or her.

There are two ways to manipulate logical statements that contain Ought. One way is to use a replacement rule, the other is to use the MoralConclude[ ] command. Both are weak routines, but the first is weakest.

`MoralConclude[`*argument*`]`	supplements Conclude with the Deontic Axiom (Op & p ⇒ q) ⇒ Oq
`DeonticAxiom`	gives the Deontic Axiom in rule format (Ought[p_] & p_ ⇒ q_) :> Ought[q]

MoralConclude can best be used in combination with the function Conclude of *The Economics Pack*. Conclude is further not explained here. The DeonticAxiom can be combined with Infer, idem.

Let us further develop the issue by clear words rather than p and q. Let us consider two statements. The first is philosophical since it exactly copies the structure of the axiom.

- An instance of the axiom.

stat1 = Ought[¬drown] && (¬drown ⇒ help)

(Ought(¬ drown) ⋀ (¬ drown ⇒ help))

- Using a replacement rule now is fast and right on target.

stat1 /. DeonticAxiom

Ought(help)

The second statement is more practical and messes up the neat structure of the philosophical argument. (1) It states the conclusion when one would not help. Some people are slow to draw a conclusion so the person may drown in the mean time. (2) It clarifies that helping implies getting wet oneself. And perhaps there is danger that one drowns oneself. (3) The idea that the victim of the accident should not drown comes only as a late realisation.

- What to do ?

stat2 = (¬help ⇒ drown) && (help ⇒ getwet) && Ought[¬drown]

((¬ help ⇒ drown) ⋀ (help ⇒ getwet) ⋀ Ought(¬ drown))

- Replacement now gets us nowhere. See the discussion above on the difficulty of using replacing rules (the axiomatic method).

stat2 /. DeonticAxiom

((¬ help ⇒ drown) ⋀ (help ⇒ getwet) ⋀ Ought(¬ drown))

Let us now use the Conclude and MoralConclude routines.

- Note that we first initialise Conclude[] - this sets Conclusions = {}. Subsequent calls give only the news. Then, the logical conclusions from the first statement are not impressive.

```
Conclude[]; Conclude[stat1]
```

{(drown ⋁ help), Ought(¬ drown)}

- New conclusions from the second statement are neither impressive. Note that And and Or are not Orderless.

```
Conclude[stat2]
```

{(help ⋁ drown), (¬ help ⋁ getwet)}

- This would be the moral conclusion however.

```
MoralConclude[stat2]
```

{Ought(getwet), Ought(help)}

Some philosophers argue that, since getting wet cannot be a strong moral imperative, the deontic axiom only has limited application. Yet in this case it spells out what should be done.

6.3 Knowledge, probability and modal logic

This section is rather brief since it proposes to exploit the trident of deontic logic to merely substitute the other labels. The suggestion is that the one trident is categorical for all such tridents.

- *Mathematica* would allow substitutions like this:

```
Options[ToModalLogic] =
    Thread[{Ought, Freedom, NotAllowed} → {Necessary, Contingent, Impossible}];

ToModalLogic[x_, opts___Rule] := x /. {opts} /. Options[ToModalLogic]
```

- For example:

```
Necessary[] = ToModalLogic[Ought[{p, r}]]
```

Necessary({p, r})

An alternative method is to simply define Necessary = Ought, Contingent = Freedom, NotAllowed = Impossible. This might well find a philosophical interpretation (i) that ought for nature is necessary, (ii) what allows freedom for nature is contingent, and (iii) what is not allowed for nature is impossible. (This creates a bridge between Is and Ought by turning everything into morals. In this case it is merely a way out of a programming problem.)

A third alternative is to work with deontics directly and merely translate the conclusions.

A fourth possibility is to copy the deontic logic package and create packages for the new applications. Eventually this might be done, if only for psychological reasons.

6.4 Application in general

6.4.1 Introduction

This section is only to remind us that there is a larger world outside. Applications of logic and inference can be done by single persons or groups, with or without the support of logical machines. Group decision making has its paradoxes too, see Colignatus (2007b), "Voting theory for democracy". What is useful in the context of this book are the routines available for *Mathematica*. Roman Maeder once published some routines that were inferential, I lost the reference. In *MathSource* there are some packages on logic and set theory but not much developed. There is also a package that uses the sophisticated environment of *Mathematica* to program a very simple and ugly looking kind of inference, which leaves one wondering. The following two subsections mention what else is in *The Economics Pack* (TEP).

6.4.2 Analysis of longer texts

In analysing longer texts, it appears to be useful to have some text management facilities. In particular, paragraphs can be analysed by themselves, and intermediate logic steps can be eliminated since they need not be relevant for the final conclusion. An example of a longer analysis is given in LogicExample.nb, that discusses the Financial Times editorial of Friday July 26 1991.

See TEP `4.3` and `Financial Times Editorial` and `FT Editorial Analysis` .

6.4.3 Logic laboratory and inference

You may wish to experiment with various rules on various logical statements. For example, you may wish to extend the AndOrRules[ ] used in Enhancement. In that case the procedure LogicLab can be useful. See TEP `4.4` and `Logic laboratory` .

See `here` for a longer discussion of inference with the Inference` package.

Part III. Alternatives to two-valued logic

As alternatives to two-valued logic we will consider:

1. three-valued logic - that is obviously an alternative
2. "intuitionism" that is said to be an alternative but then actually is misunderstood
3. proof theory, that is claimed to be two-valued, but that reduces to contradictions because of the Liar.

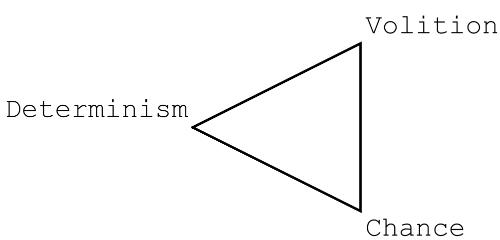
These are also three approaches to deal with the Liar paradox. Never had nonsense so much attention from serious researchers as happened with the Liar. We already solved it in two-valued logic but let us now clinch it for the remaining bits of "meaning" in its corpus.

7. Three-valued propositional logic

7.1 Introduction

A three-valued logic was already considered by Aristotle. His example was that if every proposition is true or false then it must be true *today* or it must be false *today* whether there will be a sea battle tomorrow. In this way everything would be determined. Human volition would be lost, and Courts of Law could no longer convict criminals since crimes were predetermined. An alternative might be to accept a truth value Undetermined. This, however, is not *Indeterminate.*

In general, Volition, Determinism and Chance form a threesome in which each is contrary to the other two. Chance stands for Randomness rather than "Chaos" in the modern definition since the latter is "deterministic but seemingly randomness". In all cases they represent only *perspectives* and a person can never be sure what is the true state of the world. Each perspective generates models that can be used pragmatically to solve cases. In Courts it is useful to maintain the idea of volition since it educates people that they will be held accountable. In economic forecasts it helps the forecaster to think that things aren't fully unpredictable. In other instances it can be a comforting idea that events can be rather random so that you don't know what to expect, with the joy of surprise or that you aren't responsible. Yet, the choice of a perspective always remains a moral choice. See Colignatus (2005) for a longer discussion (with the same summary).



Aristotle's argument can be accommodated in the $S \vdash p$ mold. With S a prediction system, there can be a prediction of p, a prediction that *not-p* or no prediction at all. A prediction at time t for $t + 1$ can be a good or bad prediction. We can allow for

determinism and just accept that we don't know everything. Except for the Courts of course where we must assume morality. Hence, Aristotle's argument concerns epistomology and morals, and does not really require an adaptation of the True | False dichotomy.

Independent of that, there still seems to be an isomorphism between the threesome $S \vdash p$, $S \vdash \neg p$ and $S \nvdash p$ anyway, so that the development of three-valued logic can be a bit overdone since there already might be a model to express it. Yet it is good to have seen three-valued logic. Undetermined or undecided differs from *indeterminate.*

The reason to consider three-valued logic here is that we want to accommodate for sentences that are non-sensical, such as the Liar paradox. For two-valued logic it holds that the expression cannot be valued at all but we would like to have a more general system that accomodates for the richness of language. The inadequacy of two-valued logic shows from taking the truthtable of † (NotAtAll). One of the four singulary operators always gives a falsehood whatever state of the world. But that still is falsehood and still allows for the Liar.

TruthTable[NotAtAll[p]]

$$\begin{pmatrix} p & \dagger\, p \\ \text{True} & \text{False} \\ \text{False} & \text{False} \end{pmatrix}$$

Hence we *must* add a third row. $\mathbb{P}$ are the statements in two-valued logic and $\mathbb{S}$ are the sentences that may also be three-valued. Here $\mathbb{P} \subset \mathbb{S}$ which means that within three-valued logic we still have a core of two-valuedness that allows us to discuss in a True | False manner the three-valuedness of other sentences. The truthfunction w: $\mathbb{P} \to \{1, 0\}$ is a subfunction of W: $\mathbb{S} \to \{1, 0, \frac{1}{2}\}$, and statements like $W[p] = 1$ are still two-valued. This is not a distinction in "levels" but in *values.* Statements in $\mathbb{P}$ might also be called "well-defined" or "exact". Since it doesn't matter we can write $W = w$.

Properly ordered, the sequence is True | Indeterminate | False or $\{1, \frac{1}{2}, 0\}$. This might make sense in some respects but it appears more useful to merely append Indeterminate to the original order, since this allows for a quicker verification of the two-valued part. Nevertheless, one should keep in mind that True has the maximal value and False the minimum value.

7.2 Propositional operators, revalued

7.2.1 Definition

When we want to express that we don't know the truth about p then we might simply say only $p \lor \neg p$, which is true, and leave it at that. However, if we wish to be able to define a Liar sentence then we reckon that such sentences are neither true nor false, hence indeterminate. This option causes us to consider a three-valued logic.

Three-valued logic contains a core where two-valuedness still works. Thus, a nonsensical statement has truthvalue Indeterminate, but it is True | False to say that it has no True | False value. It turns out that some such statements can become confusing (see below), that is, when you are not used to the subject.

In three-valued logic we use {True, False, Indeterminate} and $\{1, 0, \frac{1}{2}\}$. In one respect the use of three truth values changes the definitions of the logical operators since three is more than two. In another respect there is little change since we keep associating Not[p] with 1 - p, *And* with Min, *Or* with Max, *Implies* with ≤ and *$Equivalent* with =.

NB. Since $Equivalent has only two variables we will adapt it for three-valued logic. Since Equivalent can have more variables we will leave it unchanged.

- In the default setting of *Mathematica* we find that And and Or already function as Min and Max also with respect to Indeterminate

```
{True ∧ Indeterminate, True ∨ Indeterminate}
```

{Indeterminate, True}

```
{False ∧ Indeterminate, False ∨ Indeterminate}
```

{False, Indeterminate}

- But Not and Implies don't quite evaluate, even with LogicalExpand. There remains a "¬ Indeterminate" mystery.

```
Not[Indeterminate] // LogicalExpand
```

¬ Indeterminate

```
Indeterminate ⇒ False // LogicalExpand
```

¬ Indeterminate

- Implies evaluates when it is just the true Or statement.

```
Indeterminate ⇒ True
```

True

- This is also acceptable.

Indeterminate ⋀ Indeterminate ⋀ Indeterminate

(Indeterminate ⋀ Indeterminate ⋀ Indeterminate)

% // LogicalExpand

Indeterminate

We hesitate to use LogicalExpand or Simplify since these force two-valuedness. But if we don't use LogicalExpand and Simplify then some expressions don't evaluate well, even though we know that they could. Hence we must develop a separate system to evaluate expressions in three-valued logic.

Three-valued logic is turned on with the command ThreeValuedLogic[True | On]. It disables AndOrEnhance since the latter uses two-valued logic. If you are not certain about what state the system is in, you can evaluate LogicState[ ] that checks on ThreeValuedLogic[ ] and AndOrEnhance[ ]. Note that LogicalExpand and Simplify always remain two-valued as they remain internal to the *Mathematica* system. If you don't want to change the definitions of the logical operators but merely want to experiment with some replacement rules, then you could use the ThreeValuedRules[...]; but these cannot be used anymore once the use of three-valued logic has been turned on.

ThreeValuedLogic[True]

ThreeValuedLogic::State : The use of three−valued logic set to be True

NB. The properties above on Implies have now been changed - see below.

`ThreeValuedLogic[x]`	for x = True | On turns on the use of three−valued logic. Other entries turn it off and return to the default *Mathematica* system
`ThreeValuedLogic[]`	is True if three−valued logic is turned on, otherwise False
`LogicState[]`	checks on ThreeValuedLogic[] and AndOrEnhance[], which cannot both be True
`ThreeValuedRules[x]`	when x is Blank then equal to Join[ThreeValuedRules[True], AndOrRules[True]] where ThreeValuedRules[True] implements three−valued logic. When ThreeValuedLogic has been turned on then these rules cannot be used
`NotAtAll[p]`	expresses that neither p nor $\neg p$ apply, but a third value in three−valued logic, denoted as †p with a truthvalue of Indeterminate (interpretable as 1/2)

LogicalExpand and Simplify are internal to *Mathematica* and use two-valued logic. Enter NotAtAll as such otherwise *Mathematica* will not recognize it. For texts you could use ESCdgESC. Paradox: ThreeValuedLogic[x] recognizes only two states for itself.

7.2.2 Singulary operators

- The indeterminacy of "¬ Indeterminate" can now be accepted.

TruthTable[Not[p]]

$$\begin{pmatrix} p & \neg p \\ \text{True} & \text{False} \\ \text{False} & \text{True} \\ \text{Indeterminate} & \text{Indeterminate} \end{pmatrix}$$

We don't need new symbolics since we can say TruthQ[Indeterminate] == Indeterminate. Thus it is an option to let NotAtAll remain just a label and not an operator. An operator might give the suggestion that it turns sensical statements into something nonsensical, while a label doesn't have that power and merely indicates something. Let us consider the options however. There are 3 rows and each row has 3 possible outcomes. Thus there are 3^3 = 27 possible singulary operators. Definitely there is one that would have a True on the bottom row. Considering this, it makes sense to accept †*p* as precisely the translation of TruthQ[*p*] == Indeterminate (which is in two-valued logic).

- The definition of truth for two-valued logic only knows two values, but from the onset we already allowed the third value.

TruthQ[Indeterminate]

Indeterminate

- The label † can usefully be interpreted as an operator. The truthtable reflects proper use of the label. NB. Input must be NotAtAll[p] and not †*p*, since the latter only displays as such (see how it is with $Equivalent).

TruthTable[NotAtAll[p]]

$$\begin{pmatrix} p & \dagger p \\ \text{True} & \text{False} \\ \text{False} & \text{False} \\ \text{Indeterminate} & \text{True} \end{pmatrix}$$

PM. In two-valued logic there were 4 singulary operators. We now have taken the first one (False) and included a value on the third row. Earlier we thought we had no use for this column but now it appears there is some use.

PM. With 27 singulary operators, we have used only TruthQ, Not and NotAtAll now. Do we need more ? It appears that we don't need the others. With three operators we can access all rows, and that suffices.

7.2.3 Binary operators: And and Or

In two-valued logic we had 4 pairs with each 2 options giving 2^4 = 16 binary operations but now we have 9 pairs in a truthtable with each 3 values, giving 3^9 = 19683 possible

binary operators. This is not only bewildering but another consequence is that an operator will quickly have some cousins that somehow look like it. Fortunately, we need only access to nine separate rows, and, we have some strong conventions to do so.

For And we can maintain the Min condition so that the truthvalue w[And[p, q]] = Min[w[p], w[q]] so that for example Min[1, $\frac{1}{2}$] = $\frac{1}{2}$ and so that Min[1, Indeterminate] gives Indeterminate. In Chapter 1 we translated $p \wedge q$ in algebraic terms as Min[p, q] == 1. We now lose the equivalence that allowed only the 0 alternative. If Min[p, q] ≠ 1 then values 0 or $\frac{1}{2}$are possible.

- For And the adjusted minimum condition gives:

TruthTable[p ⋀ ¬ p]

$$\begin{pmatrix} p & (p \wedge \neg p) \\ \text{True} & \text{False} \\ \text{False} & \text{False} \\ \text{Indeterminate} & \text{Indeterminate} \end{pmatrix}$$

- For And the adjusted minimum condition gives as well:

SquareTruthTable[p ⋀ q]

$$\begin{pmatrix} (p \wedge q) & q & \neg q & \dagger q \\ p & \text{True} & \text{False} & \text{Indeterminate} \\ \neg p & \text{False} & \text{False} & \text{False} \\ \dagger p & \text{Indeterminate} & \text{False} & \text{Indeterminate} \end{pmatrix}$$

- For Or a similarly adjusted maximum condition gives:

TruthTable[p ⋁ ¬ p]

$$\begin{pmatrix} p & (p \vee \neg p) \\ \text{True} & \text{True} \\ \text{False} & \text{True} \\ \text{Indeterminate} & \text{Indeterminate} \end{pmatrix}$$

SquareTruthTable[p ⋁ q]

$$\begin{pmatrix} (p \vee q) & q & \neg q & \dagger q \\ p & \text{True} & \text{True} & \text{True} \\ \neg p & \text{True} & \text{False} & \text{Indeterminate} \\ \dagger p & \text{True} & \text{Indeterminate} & \text{Indeterminate} \end{pmatrix}$$

- The conversion between And and Or via Not is maintained.

SquareTruthTable[Not[¬ p ⋁ ¬ q]]

$$\begin{pmatrix} \neg(\neg p \vee \neg q) & q & \neg q & \dagger q \\ p & \text{True} & \text{False} & \text{Indeterminate} \\ \neg p & \text{False} & \text{False} & \text{False} \\ \dagger p & \text{Indeterminate} & \text{False} & \text{Indeterminate} \end{pmatrix}$$

7.2.4 Implies

For Implies, we keep the interpretation that $p \Rightarrow q$ expresses $\leq$. Using $\leq$ with value $\frac{1}{2}$ means that the implication only returns True | False and thus uses two-valued logic to discuss three-valued outcomes. The price is that we lose the interpretation $\neg p \vee q$ that still uses three values. For this, we can use NotpOrq.

- For example, the lower right corner $\frac{1}{2} \leq \frac{1}{2}$ should give True.

SquareTruthTable[p ⇒ q]

$(p \Rightarrow q)$	q	$\neg q$	$\dagger q$
p	True	False	False
$\neg p$	True	True	True
$\dagger p$	True	False	True

- Some readers may prefer the line format for this important step.

TruthTable[p ⇒ q]

p	q	$(p \Rightarrow q)$
True	True	True
True	False	False
True	Indeterminate	False
False	True	True
False	False	True
False	Indeterminate	True
Indeterminate	True	True
Indeterminate	False	False
Indeterminate	Indeterminate	True

While the implication retains EFSQ for sensical statements, we now have a more limited *ex* nonsense *sequitur quotlibet*. From "Woolly wozzy wub" you can derive that Hungary should attack India, provided that you are willing to assert the antecedens, and provided that Hungary indeed attacks India. If Hungary doesn't attack India then an implication from any nonsensical statement is false. So you can only confuse the Hungarians with your nonsensical statements implicating that attack once it is underway.

Indeterminate ⇒ False

False

- The relation to $\neg p \lor q$ remains intact for NotpOrq, by definition. When Indeterminate stands for $\frac{1}{2}$ then the NotpOrq relation cannot be $\leq$. See the bottom right corner.

SquareTruthTable[NotpOrq[p, q]]

$$\begin{pmatrix} (\neg p \lor q) & q & \neg q & \dagger q \\ p & \text{True} & \text{False} & \text{Indeterminate} \\ \neg p & \text{True} & \text{True} & \text{True} \\ \dagger p & \text{True} & \text{Indeterminate} & \text{Indeterminate} \end{pmatrix}$$

7.2.5 Equivalent

- \$Equivalent still uses = and remains equivalent to $p \Rightarrow q \land q \Rightarrow p$ especially since we adopted that $\leq$ interpretation for implication. Both have two-valued outcomes only. \$Equivalent thus still can state that expressions are equivalent (i.e. have the same truthvalues).

SquareTruthTable[$Equivalent[p, q]]

$$\begin{pmatrix} (p \Leftrightarrow q) & q & \neg q & \dagger q \\ p & \text{True} & \text{False} & \text{False} \\ \neg p & \text{False} & \text{True} & \text{False} \\ \dagger p & \text{False} & \text{False} & \text{True} \end{pmatrix}$$

- Equivalent and NotpOrq don't seem very useful.

SquareTruthTable[NotpOrq[p, q] ⋀ NotpOrq[q, p]]

$$((\neg p \lor q) \land (\neg q \lor p))$$

$$\begin{pmatrix} & q & \neg q & \dagger q \\ p & \text{True} & \text{False} & \text{Indeterminate} \\ \neg p & \text{False} & \text{True} & \text{Indeterminate} \\ \dagger p & \text{Indeterminate} & \text{Indeterminate} & \text{Indeterminate} \end{pmatrix}$$

SquareTruthTable[Equivalent[p, q]]

$$\begin{pmatrix} p \Leftrightarrow q & q & \neg q & \dagger q \\ p & \text{True} & \text{False} & \text{Indeterminate} \\ \neg p & \text{False} & \text{True} & \text{Indeterminate} \\ \dagger p & \text{Indeterminate} & \text{Indeterminate} & \text{True} \end{pmatrix}$$

7.2.6 TruthValue

- Now there are halves weighted by a larger number of possible states of the world.

TruthValue[p ⇒ q]

$$\frac{2}{3}$$

7.3 Laws of logic

7.3.1 Basic observations

As said, there still are expressions in two-valued logic to discuss outcomes in three-valued logic. If we hadn't kept Implies and $Equivalent as above, then we could have used the function w: $\mathbb{S} \to$ {True, False, Indeterminate} or $\{1, \frac{1}{2}, 0\}$ and then a statement like w[p] ≤ w[q] would be two-valued again. Among the $3^9 = 19683$ possible binary operators there obviously are some with only True | False outcomes. What is relevant is our interpretation. The given definitions for Implies and $Equivalent seem quite acceptable.

Kleene (1952) contains a discussion of three-valued logic, Kripke (1975) suggests the use of it. Styazhkin (1969) suggests that part of the table of NotpOrq was already known to William of Ockham. One can imagine that, since it is a rather natural development when you use the idea that Indeterminate would stand for True | False but unknown which. Now, however, we use Indeterminate for nonsensical statements. In that case it is more useful to have an Implies that differs from the NotpOrq.

7.3.2 Some conventions remain

We can still express equivalence for expressions that have the same truthvalues.

- NotpOrq[p, q] has been defined as $\neg p \vee q$. $Equivalent can check that. We could print the table but it would only consume space.

```
TruthValue[$Equivalent[NotpOrq[p, q], ¬ p ⋁ q]]
```

1

7.3.3 Some conventions disappear - and new ones appear

The Tertium Non Datur disappears and makes place for Tertium Datur.

- Of course, introducing a third value destroys the TND.

```
TruthTable[p ⋁ ¬ p]
```

p	$(p \vee \neg p)$
True	True
False	True
Indeterminate	Indeterminate

- And this fits the mold of *a logic of exceptions*. TND holds in general with the exception of nonsense. (We earlier used "NotAProposition" but NotAtAll is a better general form.)

TruthTable[(p ⋁ ¬ p) ~Unless~ NotAtAll[p]]

p	$(\neg \dagger p \Rightarrow (p \lor \neg p))$
True	True
False	True
Indeterminate	True

- This is straightforward. We have access to all three rows.

TruthTable[p ⋁ ¬ p ⋁ NotAtAll[p]]

p	$(p \lor \neg p \lor \dagger p)$
True	True
False	True
Indeterminate	True

7.3.4 Linguistic traps

We can accept that "p is sensical iff $\neg p$ is sensical". But can we also accept "$p \lor \neg p$ is sensical iff $p \land \neg p$ is sensical" ? For, we might also consider $p \land \neg p$ to be absurd. It appears to depend on what you mean by "iff". When your "if" is a NotpOrq, then you might accept the first equivalence only for atomic p but you have to reject it for such absurdity. In our case, we have taken the stronger If and Equivalence, and we can accept both equivalences. This strength allows for universal application, which is just what we need. This also means that the "sense" of $p \land \neg p$ resides in the point that it isn't necessarily Indeterminate. For the Liar it only seems that we deduce Liar $\land \neg$ Liar but actually we find †Liar. Now, clearly, when you are not used to these definitions and are not used to consider what it means that something is *not* NotAtAll ... (as Keynes wrote in the obituary of Ramsey: "the tormenting exercises of the foundations of thought, where the mind tries to catch its own tail" - or remember Philetas of Cos who actually died of his cogitations) ... then you may be grateful that there now is a logical machine that can check things out.

- This translates "p is sensical iff $\neg p$ is sensical".

$Equivalent[Not[NotAtAll[p]], Not[NotAtAll[¬ p]]]

$(\neg \dagger p \Leftrightarrow \neg \dagger (\neg p))$

% // TruthTable

p	$(\neg \dagger p \Leftrightarrow \neg \dagger (\neg p))$
True	True
False	True
Indeterminate	True

- This translates "$p \vee \neg p$ is sensical iff $p \wedge \neg p$ is sensical"

$Equivalent[Not[NotAtAll[p ⋁ ¬ p]], Not[NotAtAll[p ⋀ ¬ p]]]

$(\neg \dagger (p \bigvee \neg p) \Leftrightarrow \neg \dagger (p \bigwedge \neg p))$

% // TruthTable

$$\begin{pmatrix} p & (\neg \dagger (p \bigvee \neg p) \Leftrightarrow \neg \dagger (p \bigwedge \neg p)) \\ \text{True} & \text{True} \\ \text{False} & \text{True} \\ \text{Indeterminate} & \text{True} \end{pmatrix}$$

7.3.5 Transformations

The current implementation of three-valued logic includes some adaptations in And and Or, different from the full enhancement for two-valued logic. This applies especially to the case of more than two variables, with the idea to make things more tractable. As a consequence LogicalExpand does not work as it used to work.

- Expansions are contracted again.

try = p ∧ (q ∨ r);

try2 = $Equivalent[try, LogicalExpand[try]]

$((p \bigwedge (q \bigvee r)) \Leftrightarrow (p \bigwedge (q \bigvee r)))$

- Above tautology is recognized as such.

TruthValue[try2]

1

There is also another design feature that isn't necessarily usual for three-valued logic but that depends upon some present choices. The DNForm routine doesn't show the proper DNF anymore, though it still shows all the states in the world that are contained in that equivalence.

- Gives this some time to compute. The result is not the true DNF because of the adapted And and Or.

try2 // ToDNForm

$((r \bigvee \neg r \bigvee \dagger r) \bigwedge (q \bigvee \neg q \bigvee \dagger q) \bigwedge (p \bigvee \neg p \bigvee \dagger p))$

- But the truthvalue of the equivalence is OK.

% // TruthValue

1

7.4 Interpretation

The above uses the *learning by doing* format so that you have some idea what three-valued logic entails. Now is a good moment to wonder what it actually means.

Even though language already contains ways to discuss nonsense, three-valuedness still causes confusing or paradoxical statements. Someone might argue that if "Woolly wozzy wub" is not true or false but indeterminate then at least it can be said ""Woolly wozzy wub" is indeterminate" (note the quotes within quotes) so that it does have some meaning, hence sense (quod non), hence must be true or false. Such a person might claim that since he or she can reason about the statement it must have some inherent truth or falsehood. Or another person might suggest that if something is indeterminate then it certainly is false for the world, and hence false. We have given definitions of terms that exclude that kind of reasoning but perhaps those haven't sunk in. Or some people simply don't want to follow those definitions. Nevertheless, clarity is essential.

Indeterminate stands for nonsensical statements and True | False for statements that concern nature, logic and mathematics. This means that "indeterminate statements" don't occur in nature. Indeterminate actually does not belong to the series True | False which we associate with nature but rather with a series true | false | indeterminate which we associate with language. We also have to consider the assertoric convention, the $S \vdash p$ concoction. The following gives a useful table, where S stands for Socrates.

Nature	*Language*	*Assertion*
True	true	$S \vdash p$
False	false	$S \vdash \neg p$
	indeterminate	$S \Psi p$

Three-valuedness doesn't seem to belong to nature and neither to the assertoric convention. The assertoric convention for two-valued logic was: say what you consider true, be silent otherwise. With three-valued logic this becomes complicated. With only two modes of speaking and three values, something has to give. One option is to wave a red flag when you are speaking nonsense, but many people tend to forget to bring along their red flag. It appears however that language has created lots of expressions to indicate the deviations from the pure assertoric use, thus embedding three-valuedness in two-valuedness, such as "I think ..." or "Suppose ..." or "I heard ...". In a way language thus restores the two-valuedness for discussion of three-valued statements. But obviously this linguistic convention is not fail-safe, as someone might say "I lie".

There is one school of thought that allows for nature only constant expressions. An example is A_0= "On January 3 2007 it rained in Holland". Clearly this excludes the use

of a variable *p* since there is no way to check whether a *variable* is True or False. Of this school there is a subschool of thinkers who argue that nature only "is" so that False does not come into question either - the world is only True. There is another school of thought, perhaps with only one member but at least it is your author, who thinks that the latter purity is all neat and fine, but little practical. The purity creates all kinds of concepts in all kinds of realms which all look alike but that don't help us out in practice.

It is much more practical, and that is the proposed solution, to assume a bijection between nature and a part of language, with $\mathbb{P} \subset \mathbb{S}$, so that we get True | False | Indeterminate as categories *in language*, and compare these with what Socrates asserts.

It is another step to join up the latter too, interpreting nonsensical statements as those that cannot be asserted, so that we actually don't need three-valued logic but merely the notions of assertion. This might just be too quick, see the chapter on proof theory.

But the notions of assertion help to provide an interpretation of three-valued logic.

Thinking of Socrates' original concept of the discourse within the soul, we find that reasoning tends to be two-valued. Either we assert something or we assert its opposite - otherwise we wouldn't thinking. But we could assert that we assert neither some *p* nor its *not-p*.

We can also observe that reasoning is dynamic and that a first stage of reasoning is of a hypothetical kind. When a contradiction is reached and when a contradiction appears incurable because of its assumptions then this will only mean that the concepts involved are deficient, have no useful empirical application, and it is decided, in a second stage, that the matter is nonsensical. Indeed, it is always possible to consider nonsensical statements (like the Liar) in a hypothetical manner and this does not oblige us in any way to opinions on reality. But, paradoxes like the Liar must necessarily remain in the hypothetical stage and only sensical statements get into practical storage and empirical application. This practical application might be regarded as the third stage after acceptance. (It remains a philosophical question whether we ever reach real knowledge.)

One would think that this approach is the only useful interpretation of three-valued logic. The Indeterminate doesn't occur in nature but in some hypothetical stage within human reasoning. It is only simplified modelling to use a True | False | Indeterminate distinction for language.

The point applies to the Liar. Consider the dynamic process of reasoning. First truth and falsehood are considered opposites - for otherwise one would not understand what the Liar is about. When the contradiction is found it is realized that the concepts 'true' and 'false' were used hypothetically. Subsequently the Liar is called 'nonsensical', and, by applying three-valued logic, it may be called 'not true' and 'not false', where 'true' and

'false' are now interpreted in an absolute manner, i.e. referring to reality and logical consistency. This dynamic meaning of the terms should not alarm us for it is just the replacement of a two-valued logic with an overlapping three-valued scheme. The definition of truth for two-valued logic is contained in that of three-valued logic (w is in W). As shown, one cannot properly define the Liar in two-valued logic so that its proposed definition belongs to the realm of three-valued logic. After the hypothetical stage has been left, the concept 'false' in "This sentence is false" is interpreted as one of a threesome, though it still can be reasoned in two-valued logic that there are three values. The very point that the discussion proceeds in terms of the truthvalues, and not the statements themselves, indicates the possible choice of three-valued logic. Since the truthfunction W still is defined in two-valued logic, the discussion can proceed using assertoric two-valued logic but the subject is three-valued logic.

Can $S \vdash p$ replace three-valued logic ? Three-valued logic has no added value if and only if the threesome of $S \vdash p$ can be used with such an interpretation of dynamic hypothetical reasoning. Then we would have an isomorphism. When the latter is not the case then it remains useful to maintain the model that $S \vdash p$ primarily concerns deductions on True | False questions, with some statements undecided though still sensical. In that case we keep a useful distinction between undecidable and nonsense.

There is opposition to three-valued logic. Quine (1990:92): "One might accordingly relinquish the law of excluded middle and opt rather for a three-valued logic, recognizing a limbo between truth and falsity as a third truth value. (...) But a price is paid in the cumbersomeness of three-valued logic. (..) proliferation runs amok. It can still be handled, but there is an evident premium on our simple streamlined two-valued logic. We can adhere to the latter, in the face anyway of the threat of empty singular terms, by simply dispensing with singular terms (..). 'Camelot is fair' becomes '$\exists x$ (x is Camelot and x is fair)'. It does not go into limbo; it simply goes false if it is false that $\exists x$ (x is Camelot). The predicate 'is Camelot' is seen on a par with 'is fair', as a predicate irreducibly."

In the same way you might translate "This statement is false" with "This statement is false unless it isn't a statement", and conclude that it isn't a statement at all. But that approach forces the issue. The latter conditional sentence has all the proper formats to make it acceptable as a statement so that the condition must be judged to be true. The better rule of exception is "This statement is false unless it isn't a statement *at all*", which is accepting three-valued logic. The "cumbersomeness" is overrated. It is no more difficult than the standard questionnaire with possibilities *Yes*, *No* and *Do*n't *know*. Or, when you close a file, a window pops up asking you whether you want to close it with saving or close it without saving, or *cancel*. It may be "cumbersome" to the logician who

has the tradition to admit of only two possibilities, but one can get used to it, and actually it is kind of handy. Besides, while there are a large amount of possible binary operations, we only need nine (include the singulary ones) to access the separate rows. Finally, three-valuedness is only used as a grave-yard for nonsense, after which we return to two-valued logic for sensible reasoning.

7.5 Application to the Liar

7.5.1 The problem revisited

Consider the earlier argument: If the sentence L = "This sentence is false" is senseless then it is neither true nor false. If that is the case then at least it is not true. But L says that it is not true. Hence it is true. Apparently it is not senseless. Etcetera. This is solved now. Apart from the proof that L cannot be defined sensically within two-valued logic (its definition is false, thus it has the sense that it cannot be defined) we now can also find the following. Let the Liar be defined as L = "w[L] = 0". It is correct that $\mathrm{w}[L] = \frac{1}{2} \Rightarrow \mathrm{w}[L] \neq 1$ but from $\mathrm{w}[L] \neq 1$ we cannot derive that $\mathrm{w}[L] = 0$ or L. *Solved*. As said in the former section, this requires notions of hypothetical reasoning and proper shifts of focus between two-valuedness and three-valuedness.

There is however a point to consider. Namely, the resurrection of the Liar within three-valued logic.

7.5.2 The fundamental tautology

To understand this, it is good to look again at the Definition of Truth: Truth[p] $\Leftrightarrow p$ which was one of the four singulary operators. The chosen definition of truth is not the only option because we also might have defined the Alternative Definition of Truth (ADOT) Truth[p] $\Leftrightarrow \neg p$ which would be the anti-assertoric convention, to only say what you disagree with. The important thing to observe is that the definition of the Liar is an instance of the ADOT: $L = \neg L$. Now clearly we cannot define DOT and ADOT at the same time, that is an contradiction. So it is well-explained that a contradiction arises.

PM. When you have problems imagining an anti-convention, consider a country where, when you buy something, you don't have to pay but you are being paid. Thus you get the car plus its price. For consistency, having money must have a low social status. You are also obliged to carry all the money with you, which is heavy. A banker will be glad helping you to carry it, but there is the interest, which means that the banker will give you some more money to carry. Eventually you are motivated to get rid of your money so that you want someone to buy something from you. If an employer is willing to have you, you pay him your wage.

Another way to understand the two-valued Liar is to look in the truthtable of p, find two columns that are each other's negation, and then define a sentence that exploits that.

Well, the same can be done within three-valued logic. We find that p and $\neg p \vee \dagger p$ do not match. Hence we define $L_3 = \neg L_3 \vee \dagger L_3$.

TruthTable[¬ p ⋁ NotAtAll[p]]

$$\begin{pmatrix} p & (\neg p \vee \dagger p) \\ \text{True} & \text{False} \\ \text{False} & \text{True} \\ \text{Indeterminate} & \text{True} \end{pmatrix}$$

We can resolve this by reasoning again in two-valued logic about a three-valued situation. Dropping the awkward suffix 3: L = "w[L] = 0 ∨ w[L] = $\frac{1}{2}$". The following table considers the possibilities and shows a cycle.

Suppose	w[L] = 1	w[L] = $\frac{1}{2}$	w[L] = 0
Find	w[¬ L] = 0	w[†L] = 1	w[¬ L] = 1
	w[†L] = 0	w[¬ L ⋁ †L] = 1	w[¬ L ⋁ †L] = 1
	w[¬ L ⋁ †L] = 0	w[L] = 1	w[L] = 1
	w[L] = 0		

Contradiction ! We can conclude: For all L in ℙ: $L \neq$ "w[L] = 0 ∨ w[L] = $\frac{1}{2}$" A subtle point is that this is a logical conclusion and not a conclusion on formation. We can keep L_3 within 𝕊. It only means that the two-valued statements in the row "suppose" do not apply, and the proper conclusion is †(†L_3) (using the suffix again).

- We might add a fourth row with Indeterminate for L_3 and True for ††L_3.

TruthTable[NotAtAll[NotAtAll[L3]]]

$$\begin{pmatrix} \text{L3} & \dagger(\dagger\,\text{L3}) \\ \text{True} & \text{False} \\ \text{False} & \text{False} \\ \text{Indeterminate} & \text{False} \end{pmatrix}$$

- This is the better expression. The right hand side of L_3 can only be used if it is presupposed that the term is not at all undefined. The contradiction shows that that condition doesn't hold so that the definition cannot be made.

TruthTable[(p ⋁ ¬ p ⋁ NotAtAll[p]) ~Unless~NotAtAll[NotAtAll[p]]]

$$\begin{pmatrix} p & (\neg \dagger(\dagger p) \Rightarrow (p \vee \neg p \vee \dagger p)) \\ \text{True} & \text{True} \\ \text{False} & \text{True} \\ \text{Indeterminate} & \text{True} \end{pmatrix}$$

The construction and solution of Liars now becomes a boring game. You define $L_4 = \neg L_4 \vee \dagger L_4 \vee \dagger\dagger L_4$ and I reply †(††L_4). Etcetera. It is a philosophical matter if you call

applications of † "a new truthvalue" (since it extends the ∨) or not.

Note that the original solution of the Liar was in that respect simpler in that it said "it doesn't exist" or "don't form it". That is a clear simple solution and it retains two-valued logic. Only by our insistence to be able to form the Liar and the creation of three-valued logic we find such an infinite "game". In the end the solutions are similar, for it may be hoped that you once stop making those liars.

Note also that we don't use "levels in language" here but merely repeated application of truthvalues to some particular expression. There can be other such infinite sinks in language and logic, but, supposedly they neither require strong restrictions on formation, just awareness that you are falling down a sink and have to crawl out.

The general solution is to become aware that above game has a constructive format that can be exploited. Rather than forbidding the formation of sentences with strict rules on levels and the like, we can give a definition of truth for a sentence *given* its form. Let $\dagger_n$ stand for a list of those daggers. and let Not = $\dagger_0$. Then we can formulate:

The fundamental tautology "Semper Alterum Datur": If p contains $\dagger_n$ then $p \vee \ldots \vee \dagger_{n+1} p$.

Which also establishes the **sufficiency** of three-valued logic for language.

Note that one better refrains from introducing new truthvalues. That is, the liars are an insufficient reason to do so. In the current case, when someone is building liars, we can ask why he or she is continuously trying to become undefined. Instead, if we would allow the introduction of new truthvalues then this builder might answer "oh, but there are so many different truthvalues, it really is complex and interesting". Only when such values are introduced for other reasons than liars then those reasons might be sound.

7.5.3 Conclusion

Having reached the solution of the Liar, we might want to look back at the attempts of previous history. Such a hindsight must be necessarily short here. We can only indicate some points but it can be observed that historical researchers mentioned some aspects of the solution. Aristotle's distinction between particular and absolute aspects of truth and falsehood reminds us of the distinction between hypothetical and sensical truth and falsehood. We don't know if there is a match since Aristotle didn't elaborate and concentrated on more important approaches like the syllogism. For the Stoics we note that they made a clear distinction between meaning and sense. This may again be a translation issue but even if they actually did not make that distinction then at least their approach can be appreciated since it entails three-valuedness. Though they didn't develop it sufficiently of course. Of the Middle Ages, the contribution of William of

Ockham is the most outstanding, reminiscent of our own conclusion that one should not form the Liar. But we decided to regard that approach as too strict since it forbids all kinds of selfreference that seem so useful. We only forbid the ones leading to contradiction - and actually not really forbid them but give a rule to show them nonsensical.

The recent approaches of Russell and Tarski with the theory of types also forbid self-reference. This approach appears unnecessary and may now also be recognized as begging the question, since there is no necessary reason why you would require types. There is some hierarchy in set theory, with Cantor's theorem that a set always causes a larger power set, but we don't have a similar theorem for language, since the powerset of $\mathbb{P}$ or $\mathbb{S}$ is no language anymore. The concept of a meta-language is also tricky. One might use it in a formal "language" that however, by being formal, is not interpreted and thus cannot be used in arguments. Thus a formal "language" would be no real language in that respect too. Natural language can be meta on anything as long as the community understands it. The notion of a meta-language seems only locally defined and not absolutely. Kripke has taken a middle position, suggesting to adopt three-valued logic and still embracing the ghost of Tarski. Selfreference would not receive a truthvalue again though, making his approach unattractive.

As opposed to these historic efforts the above has both explained and solved the Liar and its various cousins. $\mathbb{P}$ is two-valued due to nature and $\mathbb{S}$ is three-valued due to human symbolics. The Liar is not consistent for two-valued logic. That conclusion is equivalent to saying that we may discuss it hypothetically. That conclusion is equivalent again to using a third truthvalue, Indeterminate. Subsequently we can design a rule to deal with all the Liar's cousins.

7.6 Turning it off

Three-valued logic has good value just for showing what it is. But its application seems limited. To prevent confusion, don't forget to turn ThreeValuedLogic off.

- Turning off the use of three-valued logic.

```
ThreeValuedLogic[False]
```

ThreeValuedLogic::State : The use of three–valued logic set to be False

8. Brouwer and intuitionism

8.1 Introduction

Like the great Kronecker made Cantor's life difficult by refusing him a university position ("because" Kronecker didn't like Cantor's ideas on transfinites), the great Hilbert made Brouwer's life difficult by having him ousted from the board of editors of the Annals ("because" Hilbert didn't like Brouwer's ideas on methods of proof).

Brouwer is the mathematician who invented the "fixed point" and other concepts in topology. Every head with hair has a crown from which the hairs seem to flow, because there must be a fixed point from continuously mapping all points of the head onto the head itself. His counterpart Hilbert is known to have been able to derive in a day what Einstein had been working on for ten years or more (Hilbert could have done it much earlier but just didn't think of it as a relevant question).

Brouwer must be mentioned here *firstly* since he was the first serious mathematician to question the Aristotelian principle of Tertium Non Datur, *secondly* since he provided necessary elements in Hilbert's development of proof theory, which is going to play a role in the next chapter on the Gödeliar. The mathematician H. Weyl states on Brouwer's importance:

> "L.E.J. Brouwer by his intuitionism had opened our eyes and made us see how far generally accepted mathematics goes beyond such statements as can claim real meaning and truth founded on evidence. I regret that in his opposition to Brouwer, Hilbert never openly acknowledged the profound debt he, as well as all other mathematicians, owes Brouwer for this revelation." (Quoted in Heyting (1980:778))

Hilbert's own opinion was:

> "Intuitionism's sharpest and most passionate challenge is the one it flings at the validity of the principle of the excluded middle (...) The principle (...) has never yet caused the slightest error. It is, moreover, so clear and comprehensible that misuse is precluded. In particular, the principle (...) is not to be blamed in the least for the occurrence of the well-known paradoxes of set theory; rather, these paradoxes are due merely to the introduction of inadmissible and meaningless notions, which are automatically excluded from my proof theory." (Hilbert (1927:475))

The development of three-valued logic above was a useful extension to two-valued logic since it allows us to give a third value to the statements that Hilbert called nonsense ("meaningless"). For his solution of the set paradoxes Hilbert draws on the theory of types and that approach we deemed inadequate. Thus Hilbert bypasses the general principle by which the paradoxes can be regarded as nonsense, as we discussed above, and, doing so, he excluded notions of selfreference that might be mathematically useful.

Brouwer himself about Hilbert's suggestion that consistency was the hallmark of a mathematical system:

> "We need by no means despair of reaching this goal, but nothing of mathematical value will thus be gained: an incorrect theory, even if it cannot be inhibited by any contradiction that would refute it, is none the less incorrect." Brouwer (1967:336)

As Hilbert gave the dominant view that we have been discussing already, this chapter will focus on Brouwer and his intuitionism. We can learn a bit more on the word "not" and on the notion of proof.

A note may be required that this author does not regard himself related to this intuitionism. Some authors call it a baffling system of thought that they cannot get a grip on, and some have even suggested that there must be a hidden complexity in the Dutch language itself. Since Dutch is a difficult language to learn, they argued, the world better forget about even trying to understand intuitionism. I don't follow that interesting approach however and plan to show that Brouwer, definitely a brilliant mind, also was a bit mixed up. As Hilbert.

The basic observation is that Brouwer was a mathematician and no logician and no natural scientist. Heyting, Brouwer's loyal assistant, devised a set of axioms for intuitionism that (only) reduces to the standard propositional logic when you also include $p \vee \neg p$. This is too simple a description of what Brouwer intended with his notions of proof. Heyting (1980:779) actually arrives at that same conclusion. Heyting also remarks, quite confusingly, and if not paradoxical then inconsistently, and reminiscent of ostrich behavior:

> "It is good to avoid negations where it is possible." (Heyting (1980:747))

Translate this as: *do* **not** *use "not"* !

8.2 Mathematics versus logic

The most important quote of Brouwer for these pages is the following:

> "for a mathematical assertion α the two cases formerly exclusively admitted were replaced by the following four: 1. α has been *proved to be true;* 2. α has been *proved to be false, i.e. absurd;* 3. α has neither been proved to be true or to be absurd, but an algorithm is known

> leading to a decision either that α is true or that α is absurd; 4. α has neither been proved to be true or to be absurd, *nor do we know an algorithm leading to the statement either that α is true or that α is absurd.* In the first and second (first, second and third) case α is said to be *judged* (*judgeable*)." (Brouwer (1975:552))

Brouwer writes "proved to be true" (using truth too) but for us this translates as just "proven". Selecting the different dichotomies which are involved in this enumeration we find:

1. (simple) truth versus falsehood
2. necessity (tautology) versus contingency
3. proof versus absence of proof ($S \vdash p$, $S \vdash \neg p$, $S \nvdash p$)
4. sense versus nonsense (absurdity, contradiction, impossibility)

Early mathematics was satisfied with that the negation of a tautology is a contradition, while the negation of a contradition gives a tautology. Brouwer does not reject that but extends or refines it. Clearly he is not interested in contingent truth or falsehood. For *mathematical* truths he neglects simple truths but requires necessity, where this necessity must be proven (while he seems to assume $(S \vdash p) \Rightarrow p$). According to him, a *mathematical* falsehood is not just the opposite of a mathematical truth.

The various threesomes that we discussed above are transformed here into pairs and Brouwer's selection of the combinations makes sense for his field of research. Brouwer remains the mathematician. He adds the notion of the algorithm and specifically refers to finite methods, i.e. what Skolem later called recursive methods. Elsewhere he suggests that time is the most basic mathematical intuition, both for the sense of order and the sense of continuity. He also called mathematics an "autonomic interior constructional activity" (1975:551).

An important result of Brouwer's professional background is the accumulation in his writing of intricate detail even in logic: but instead of developing a general logical theory (except the above) he reserves his attention to the details of the argument (of which the generality might perhaps be clear to himself). He might have thought "I teach by example". The result however is that much of his writing in the field of logic seems as irrelevant and at least as inaccessible as most of the pages of Russell & Whitehead's Principia Mathematica. If there is a general theory then it remains in the fog (except for above 4 points).

And, there is terminology, where his "truth" is tautology and not contingent truth. In the quote above Brouwer replaces "the two cases formerly exclusively admitted" and Hilbert interpretes this as the rejection of the Tertium Non Datur, the principle of the excluded middle. But Brouwer didn't do that ! As a mathematician, and noting that

mathematics deals with *theorems,* Brouwer replaced the pair of tautology | contradition with ($S \vdash p$, $S \vdash \neg p$, $S \ \Psi\ p$) for p in tautology | contradition. He just didn't consider $p \vee \neg p$ for contingent science, the basic dichotomy in nature. Which makes much of his logic less useful for normal science.

Posed positively, his rejection of the conventions in the period contained elements that are definitely interesting: (a) a criticism of the methods of reasoning, (b) a criticism of e.g. Russell's way into and out of the set paradoxes, (c) a philosophy about the ways, possibilities, limits and objectives of mathematics. Those subjects are centered around the concept of mathematical truth, not contingent empirical truth as used in the simple concept of truth in modern logic. He was explicit on the distinction, at times, but apparently not frequent enough for Hilbert to notice (or perhaps too frequent).

Understanding Brouwer's work in this manner makes his work much clearer and correct. For example, where Ayer (1936) tended to call metaphysics nonsensical (as we also adopted a terminology that associates with the Vienna Circle), Brouwer is more elegant and says "In wisdom there is no logic" (Brouwer (1975:111)).

8.3 Russell's paradox

Brouwer (1975:89-90) discusses Russell's paradox. In summary: (i) he *affirms* Tertium Non Datur for mathematics (if Hilbert only knew !), (ii) he makes the useful distinction between sense ("mathematics") and mere linguistic systems, (iii) he excludes selfreference (particularly where 'totality' is involved), (iv) he concludes to *undecidability* rather than *contradiction*. The contradiction only arises by trying to decide something that is undecidable.

> "RUSSELL suggests various methods to escape from the contradiction, but he ends by rejecting them; he believes that a deep-searching reconstruction of logic will be needed for the solution. He is inclined to the opinion that a theory is required which does not admit every class, considered as one, to be made into a logical subject. 'Another suggestion,' he says, 'would be to demur to the notion of *all objects,* but in any case the notion of *every object* must be retained, for there are truths, viz. the logical principles, which hold for every object.'
>
> But this is mistaken: the logical principles hold exclusively for words with a mathematical content. And exactly because RUSSELL's logic is no more than a linguistic system, deprived of a presupposed mathematical system to which it would be related, there is no reason why no contradiction would appear.
>
> For that matter, it is evident to common sense at which point the reasoning, which leads to the contradiction, ceases to be alive and consequently is no longer reliable; it is even

unnecessary to give up the illusion of the chimerical 'everything'. For let us suppose that I know an 'everything' with a 'totality' of relations existing between the objects, and a system of propositions which may hold for the objects. Then, given a propositional function, I can decide for *any object* by means of its given relations whether or not it satisfies the function, in other words, to which of the two classes defined by the function it belongs.

But when I wish to decide whether the object which is the class involved in the contradiction, satisfies the given propositional function, then I see that the decision is only possible under the condition that it has already been completed. Consequently the decision *cannot be taken,* and hereby the contradiction is explained. We have here a propositional function which defines two complementary classes which do not satisfy the tertium non datur. This is not surprising, for the logical principles hold only for the language of mathematics; for other linguistic systems, however akin to that of mathematics they may be, the principles need not hold."

Brouwer's fall-back position is that mathematics gives sense when and only when the concepts are constructive. There must be a scheme to start with the small and build up to the big. The definition of Russell's set fails that. Brouwer's approach can be seen as 'positive' since it both gives an explanation and a recipe for creating useful things. But the price he pays is the loss of selfreference. Above we showed that we can allow for paradoxical statements, also within mathematics, by allowing for a third value.

8.4 Unreliability of logical principles

Brouwer's "De onbetrouwbaarheid der logische principes" (1908) must be regarded as brilliant even though we don't fully accept it. We may give some extensive quotes so that one can verify our statements about Brouwer and benefit from his insights. In the following, Brouwer reminds of the development of non-Euclidean geometry and the conclusion of scientists that it depends upon the facts what mathematical system describes reality. Subsequently Brouwer extends this empiricism to logic:

"And, like any unreligious consciousness, science has neither religious reliability nor reliability in itself. In particular, a mathematical system of entities can never remain reliable as a guide along our perceptions, when it is indefinitely extended beyond the perceptions which it made understandable.

Consequently logical deductions which are made independently of perception, being mathematical transformations in the mathematical system, may lead from scientifically accepted premisses to an inadmissable conclusion.

The classical approach, based on the experience that in geometry logical reasoning deduced only undisputable results from accepted premisses, concluded that logical

reasoning is a method for the construction of science and that the logical princples enable man to construct science.

But geometrical reasoning is only vald for a mathematical system which can be mentally constructed without reference to any experience whatever, and the fact that such a popular field of experience as geometry conforms so lastingly to the corresponding mathematical system ought to be distrustred, like any successful part of science.

Because we now understand that logical deductions are unreliable in science, Aristotle's conclusions on the structure of nature do not convince us without practical verification; we feel the wisdom that blooms in Spinoza's work as completely independent of his logical system; we are no longer annoyed by Kant's antinomies or by the absence of physical hypotheses which can be carried through in their extreme consequences.

Moreover, the function of the logical principles is not to guide arguments concerning experience subtended by mathematical systems, but to describe regularities which are subsequently observed in the language of the arguments. To follow such regularities in speech, independently of any mathematical system, is to run the risk of paradoxes like that of Epimenides.

3. The question remains whether the logical principles are firm at least for mathematical systems exempt of living sensation, i.e. systems constructed out of the abstraction of repeatable phenomena, out of the intuition of time, void of living content, out of the basic intuition of mathematics. Throughout the ages logic has been applied in mathematics with confidence; people have never hesitated to accept conclusions deduced by means of logic from valid postulates. However, recently paradoxes have been constructed which appear to be mathematical paradoxes and which arouse distrust against the free use of logic in mathematics. Therefore some mathematicians abandon the idea that logic is presupposed in mathematics. They try to build up logic and mathematics together, using the methods of the school of *logistics,* founded by Peano. But it can be shown that these paradoxes rise from the same error as that of Epimenides, to wit that they originate where regularities in the language which accompanies mathematics are extended to a language of mathematical words which is not connected with mathematics. Further we see that logistics is also concerned with the language of mathematics instead of with mathematics itself, consequently it cannot throw light on mathematics. Finally all the paradoxes vanish when we confine ourselves to speaking about systems which can be built up explicitly from the basic intuition, in other words, when we consider mathematics as presupposed in logic, instead of logic in mathematics." (Brouwer (1975:107+))

PM 1. What is first, logic or mathematics, is of no concern to us.

PM 2. A modern name for the school of Peano is symbolic logic. We would rather see Frege as the founding father of symbolic logic, following Bochenski (1970). "Logistics" is the branch in economics where the objective is to get a product in the right form, and at the right time in the right place. (Curiously for Brouwer, time enters here too.)

PM 3. By way of terminology, Brouwer apparently requires of mathematics "which can be mentally constructed without reference to any experience whatever" that it also is "constructed out of the abstraction of repeatable phenomena, out of the intuition of time, void of living content, out of the basic intuition of mathematics". This may sound inconsistent but the key word is abstraction, meaning that the ghost of "any experience whatever" still can be retained, namely in that abstraction, now called "intuition".

According to Brouwer the fundamental principles of mathematics are basic to an intelligent mind, cannot be further explained. Language and symbolic logic reflect experience but not sufficiently abstract and hence they can lead to contradictions. The following diagram gives a scheme where the horizontal lines reflect everyday sloppiness and the vertical movements reflect the rigour of mathematics and its abstraction.

$$
\begin{tikzcd}
\text{reality} \arrow[r] \arrow[dr] & \text{language} \arrow[r] \arrow[d] & \text{logic} \\
& \text{mathematics} \arrow[ur] &
\end{tikzcd}
$$

Personally, I am not much impressed by this analysis. (1) The arrows between language, logic and mathematics can run both ways. The words to describe them are not well-defined. It is not proven that abstractions come before experience. Rather you have to experience and learn something before you can abstract from it. "Abstractly most fundamental" might read as "epistemologically least fundamental". If abstraction is purest, and logic based upon the mathematical analysis (of the structure of scientific theories) then logic would be even more abstract than mathematics. Brouwer's approach is more on the psychology of mathematics than on logic and truth. (2) That Peano et al. approached it differently does not disqualify logic in itself. Modern logic is not a study of language but deals with the necessities of inference. (3) It is not clear why the intuition of time should be fundamental to mathematics and why not the intuition of True | False. This might also cause a discussion whether space isn't important too, with its option to choose a direction. Clearly space is tricky, with non-Euclidean interpretations, but does that disqualify the abstraction of space ?

What is important in this quote are the *issues*, not whether he was right or wrong. Leibniz, Boole, Frege and Peano et al. already wanted to make language cq. the language of mathematics more rigorous. Brouwer suggests that it is futile to try, since mathematics is done in your mind and more complex than 'mathematics' merely using language. He propounds the view that mathematics is more complex than logic with its True | False dichotomy, so that the logic of his time "cannot throw a light on mathematics". Brouwer requires *proof and necessity,* not just truth. These aspects were surely neglected in the discussion on logic in his time.

He continues on the subject of proof. He interpretes truth as $(S \vdash p)$ and falsehood as $(S \vdash \neg p)$ but holds that for the mathematics of the infinite it is possible that $(S \quad \Psi\ p)$. He certainly accepts "don't make a contradiction" but advises that reasoning in infinite mathematics can still be unreliable even when no such contradiction turns up:

> "Now consider the principium *tertii exclusi*: It claims that every supposition is either true or false; in mathematics this means that for every supposed imbedding of a system into another, satisfying certain given conditions, we can either accomplish such an imbedding by construction, or we can arrive by a construction at the arrestment of the process which would lead to the embedding. It follows that the question of the validity of the principium tertii exlcusi is equivalent to the question *whether unsolvable mathematical problems can exist.* There is not a shred of a proof for the conviction, which has sometimes been put forward, thet there exist no unsolvable mathematical problems.
>
> Insofar as only finite discrete systems are introduced, the investigation whether an imbedding is possible or not, can always be carried out and admits a definite result, so in this case the principium tertii exclusi is reliable as a principle of reasoning.
>
> We conclude that in infinite systems the principium tertii exclusi is as yet not reliable. Still we shall never, by an unjustified application of the principle, come up against a contradiction and *thereby* discover that our reasonings were badly founded. For then it would be contradictory that an imbedding was performed, and at the same time it would be contradictory that it were contradictory, and this is prohibited by the principium contradictionis."

I haven't delved into what that last convoluted sentence precisely means, it is not important here.

It is in this context that he continues:

> "And it likewise remains uncertain whether the more general mathematical problem *Does the principium tertii exclusi hold in mathematics without exception ?* is solvable."

The term "likewise" is imprecise, in particular where the author already lost us on the undefined "principium contradictionis". And whether the *principium tertii exclusi* holds is rather a matter of definition rather than something that must be proven. Yet, Brouwer here famously casts doubts upon the very logical principles that we reason with. And, for the record, he uses the words "without exception". The iron cast is broken. Brouwer doesn't provide the convincing alternative yet, but we can appreciate his words, when we define those logical principles to hold except where it can be proven that we must extend those definitions with three-valuedness for senseless statements like the Liar.

8.5 Concluding

The above seems like a fair reconstruction of Brouwer's views. See also Heyting (1980:249), who remarks in passing: "Die Einführung der Beweisbarkeit würde grosze Komplikationen nach sich ziehen; bei dem geringen praktischen Wert würde er sich kaum lohnen, diese Einzelheiten zo verfolgen." - and he missed his chance to beat Gödel.

Brouwers system became known as "intuitionistic logic" which is a bad name since his intuitions dealt with mathematics, yet he stuck with it himself. Heyting presented an axiomatization *without* the ($S \vdash p$) symbolism, in which *not-not-p* is **not** equivalent to p. He also stated that "intuitionist logic" does **not** involve a third truthvalue (Heyting (1980:688+701)). Note the bold "not" in the former sentences, which suggests that the "not" used in the metalanguage that discusses "intuitionism" might **not** be the "intuitionist not" since otherwise this gets very confusing. Heytings advice **not** to use "not" then becomes understandable, but that advice of course implies a failure at creating something free of confusion.

Brouwer (1967:335) accepted that p is necessary (a tautology) when the assumption of its negation resulted into a contradiction. Freudenthal (1937:332) wonders correctly and amusingly, what the meaning of 'negation' and 'contradiction' in that definition are.

To be fair to Brouwer, we can catch his idea in the following manner, using our notation, and indicating that this notation is adequate:

Ergo[Exists[q, (S ⊢ ¬ p) ⇒ (S ⊢ (q ⋀ ¬ q))], (S ⊢ p)]

$$\left(\exists_q\, (S \vdash \neg\, p \;\Rightarrow\; S \vdash (q \bigwedge \neg\, q)) \;\vdash\; S \vdash p\right)$$

Brouwer was partly right and had many relevant ideas in mind but he seems to have been premature since those ideas were still combined and not analyzed in their parts. But of course, set theory was a serious enough business to worry about and from his mathematical interest in the continuum Brouwer was sufficiently motivated and excused to put forward ideas on logic and inference as well which were not really worse than those of many others. Those ideas were provocative enough to stimulate that research of others. As a result we now use a notation of proof that is much more precise than Brouwer himself may have used.

PM 1. One way to look at Heytings axioms is that they don't formalize "not" but some "nay", and that a mathematical statement isn't True | False but True | Indeterminate. This might give strings of Nay just like with NotAtAll. Though this is perhaps consistent, it is also definitely counter-intuitive and desinformative. We rather use truth and proof.

p	Nay[p]	Nay[Nay[p]]
1	0	0
$\frac{1}{2}$	1	0
$\frac{1}{2}$	$\frac{1}{2}$	1
...	...	...

When we use Not, then it works as a “gate-keeper”, saving us from endless strings of †† .... When we forget about Not and just use Nay then we don’t have such a gate-keeper, causing complex discussions around the dinner table, and awkward ones with the kitchen.

PM 2. Another way to interprete the situation is to return to the True | False dichotomy and disallow Not but use False, the first of the singulary operators. False then is not just a value but also an operator, expressing total denial:

p	False[p]	False[False[p]]
1	0	0
0	0	0

The conclusion is that it is only a weak system and it is more advantageous to use stronger axioms (since we need not fear for a Liar paradox).

9. Proof theory and the Gödeliar

9.1 Introduction

Arithmetic has been around since the dawn of civilization, some 15,000 years ago, if not earlier. Around 1900 there were various systems to axiomize it. This caused the question what system to select, and, what to do if that axiomatic system turned out to be inconsistent ? You would get 1 + 1 = 3 and scientists working with that particular axiomatic system would generate lots of nonsense. Admittedly, such nonsense might be quickly noticed but, to do that, you would rely on your knowledge of the world, and you would not rely on the formal strength of mathematics itself. Given the power of mathematics, it should be the other way around, David Hilbert thought. Human empirical knowledge of the world is fragile and people better rely on sound math. Hence, one of the objectives of David Hilbert became to prove the consistency of arithmetic, or, indeed, systems in general.

The question seemed particularly important since it seemed that all effective computable (finite) methods of proof were recursive, i.e. used mathematical induction, which was the method of arithmetic. That method ought to be consistent as well, and the proof of the consistency of arithmethic would use that very method.

Hilbert met a student Kurt Gödel, working on a thesis. Hilbert suggested to him to map all symbols of arithmetic onto the integers themselves, so that expressions about arithmetic could be represented into arithmetic. Depending upon the map chosen, a natural number might be interpreted as a code for a statement on arithmetic itself. In that way, a kind of selfreference was established that was acceptable to the mathematical mind, compared to the sloppy informal ways of selfreference in natural language. One of those natural numbers would be the code for "Arithmetic is consistent". Hilbert asked Gödel to try whether he could find a proof for that.

Gödel considered that a system of arithmetic A is consistent when it contains statements that aren't proven.

Consistent[A, p]

$$\forall_p \left(\left(A \mathrel{\overline{\downarrow}} p \right) \bigvee \left(A \mathrel{\overline{\downarrow}} \neg p \right) \right)$$

Subsequently he came up with the selfreferential statement "This statement is not proven". In formula $g = (A \,\text{↧}\, g)$, meaning that g states that A doesn't prove it. It turned out that this is a paradoxical statement. When we apply A to some intended application then we can judge the truth and falsehood of statements, and now in particular g, see the table below. We put the cause that A proves or doesn't prove g in the columns and consider the effect on the truth or falsehood of g in the rows. And reading backwards, we find the equivalence $g \Leftrightarrow (A \,\text{↧}\, g)$.

	$A \vdash g$	$A \,\text{↧}\, g$
g	False	True
$\neg g$	True	False

The statement g clearly is a cousin of the Liar and hence we will call it the Gödeliar. Note that g is only paradoxical when we apply an interpretation to the formal system such that its statements can be judged to be true or false. Gödel called this a "metamathematical" insight but we may see it as just an application. If we don't choose an application then we just have a formal system and no paradox. PM. We write $(A \,\text{↧}\, g)$, meaning that A doesn't prove g, which is simpler than handling provability as in "This statement is unprovable". Purists might want to see a quantifier that there is no statement p in A that proves g, plus a constructive proof method. We will discuss below that this is needlessly complex.

The subsequent math in the Ph. D. thesis is tedious though straightforward. For our purposes it is sufficient to take a shortcut. The *first* shortcut is that we don't need the numerical coding and just use our symbols of the predicate calculus. Language already has neat ways to express selfreference and though it is laudable that mathematicians try to make this sufficiently exact for their purposes, we don't need to follow that track of mind. We take a sufficiently strong system S and take its Gödeliar $g = (S \,\text{↧}\, g)$. *Secondly,* while Gödel first goes through his tedium and afterwards applies his ("metamathematical") interpretation, we turn this around and start with an intended application. We are not interested in formalism just for itself and we want to apply this system. (And eventually Gödel takes that same step too.) Thus, we want S to be semantically correct (scientific), so that we use an interpretation or application of S such that all theorems of S are also true in their application. You may recall the definition $\forall_p\, ((S \vdash p) \Rightarrow p)$ and the associated truthtable. In fact, we can reproduce that table and directly apply it to g since this is the current statement that we are interested in:

TruthTable[SemanticallyCorrect, S, g]

$(S \vdash g)$	$(S \nvdash g)$	$(S \vdash \neg g)$	g	$\neg g$
1	0	0	1	0
0	1	0	1	0
0	1	0	0	1
0	0	1	0	1

Cutting short a lot of deduction, it suffices now to evaluate the rows in that last table. The first row obviously falls out (check table above on A). The last two rows require the following deduction, with (1) semantic correctness, (2) definition of g, (3) *tertium non datur* in our application:

Ergo2D[Ergo[S, g] ⇒ g , ¬ g ⇒ g, g ∨ ¬ g, g]

1	$((S \vdash g) \Rightarrow g)$
2	$(\neg g \Rightarrow g)$
3	$(g \vee \neg g)$
Ergo	————
4	g

Thus, in the semantic truthtable rows 1, 3 and 4 fall out. All that remains is $(S \nvdash g) \wedge g$. Remember deductive completeness and see how this g gives a counterexample so that the system is incomplete:

DeductivelyComplete[S, p]

$\forall_p (p \Rightarrow (S \vdash p))$

Gödel's **first incompleteness theorem:** Each sufficiently well developed system S that is consistent and (at least) uses the methods of arithmetic is deductively incomplete. There is an expression that is true but that cannot be proven in the system.

The second step that Gödel took is to return to the original question, consistency. Let c stand for the statement that S is consistent. Note that an inconsistent system could prove anything so that $\neg c \Rightarrow (S \vdash g)$ or $g \Rightarrow c$. Since we already have g we can derive now the consistency of S on a meta-systematic level. This fits the point that we have provided an application for it. The subsequent question is whether the system can establish its own consistency, $(S \vdash c)$.

When the system itself could prove both c and $c \Rightarrow g$ then there would be a proof for g, making the system inconsistent. Thus, if we can show that $c \Rightarrow g$ is true in the system then a consistent system cannot have a proof for its own consistency.

- We can use the label "Hyp:" to indicate a hypothesis that is submitted to an inferential test. In this case we launch the hypothesis about what S might be able to do.

Ergo2D[Hyp : Ergo[S, c] , Hyp : Ergo[S, c ⇒ g], Hyp : Ergo[S, g]]

1	Hyp : $(S \vdash c)$
2	Hyp : $(S \vdash (c \Rightarrow g))$
Ergo	————
3	Hyp : $(S \vdash g)$

We now adopt the axiom $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$ or that when S proves something then it considers it also proven that it proves something. This property can be called "proof consequent" since the notion of proof is applied consequently. Next, we allow S to reason in a hypothetical mode as well, where S assumes something and later retracts it. In step 1 S conjectures that it hypothetically proves $\neg g$ and then determines that it would become inconsistent. We assume that S considers a theorem of inconsistency inacceptable. Thus it retracts the assumption and just takes the derived implication in step 8:

- In this case S introduces and retracts the hypothesis on $\neg g$.

**Ergo2D[Ergo[S, Hyp: ¬g], Ergo[S, Ergo[S, g]], Ergo[S, Ergo[S, Ergo[S, g]]],
Ergo[S, Ergo[S, ¬g]], Ergo[S, Ergo[S, g ∧ ¬g]], Ergo[S, ¬c],
Ergo[S, Retract: ¬g], Ergo[S, Implies[¬g, ¬c]], Ergo[S, Implies[c, g]]]**

1	$(S \vdash$ Hyp : $\neg g)$
2	$(S \vdash (S \vdash g))$
3	$(S \vdash (S \vdash (S \vdash g)))$
4	$(S \vdash (S \vdash \neg g))$
5	$(S \vdash (S \vdash (g \wedge \neg g)))$
6	$(S \vdash \neg c)$
7	$(S \vdash$ Retract : $\neg g)$
8	$(S \vdash (\neg g \Rightarrow \neg c))$
Ergo	————
9	$(S \vdash (c \Rightarrow g))$

Hence we can record:

**Ergo2D[Ergo[S, Implies[c, g]],
Implies[Ergo[S, c], Ergo[S, g]], NonSequitur[S, g], NonSequitur[S, c]]**

1	$(S \vdash (c \Rightarrow g))$
2	$((S \vdash c) \Rightarrow (S \vdash g))$
3	$\left(S \underset{\downarrow}{\top} g\right)$
Ergo	————
4	$\left(S \underset{\downarrow}{\top} c\right)$

Gödel's **second incompleteness theorem:** Each sufficiently well developed system S that is consistent and (at least) uses the methods of arithmetic cannot prove its own

consistency. In fact, $c \Leftrightarrow (S \downarrow c)$, so that c actually is equivalent to the Gödeliar.

Gödel's **final crunch:** Suppose that we include the true g as one of the axioms of S, in an effort to make the system complete. Then, however, we get a new system $S_1 = \{S, g\} = \{S, (S \downarrow g)\}$ and then we will find a g_1 and the whole process repeats. Note that $g_1 \neq g$ since they refer to different systems, $S_1 \neq S$. Thus, we may indeed start with arithmetic and work up to any other bigger system. This adding of axioms also holds for $c = c[S]$. Hence, above theorems contained the words "at least" between brackets, and we must drop these brackets.

Gödel: "it can be proved rigorously that in *every* consistent formal system that contains a certain amount of finitary number theory there exist undecidable arithmetic propositions and that, moreover, the consistency of any such system cannot be proved in the system" (1931:616).

Hilbert's dream broke down.

Gödel himself wondered for the rest of his life whether he had really proven something or just created one of the cousins of the Liar. He never knew whether he should proudly walk the Earth or hide in shame.

```
Gödeliar = NonSequitur[Gödeliar]
```

a translation of *This statement is not proven*. This format presumes that some system S is used in totality so that a quantifier can be avoided. Otherwise you must specify yourself G = ForAll[p, p in S, NonSequitur[p, G]]

Enter the ö in *Mathematica* using ESC o" ESC. Gödeliar uses a String on the right hand side to prevent recursion.

9.2 Rejection

Gödel uses a limited kind of self-reference. Statements within S are allowed to refer to themselves but the system as a whole is not allowed to refer to itself. The definition of consistency has a quantifier "all p in S" but strangely it is not allowed to take "$c[S]$ in S". The setup is like leaving out sharp corners and then prove that squares don't exist.

When we make the system so smart that it can express the conditions for consistency, and subsequently so smart that it knows something basic about consistency too (namely that $\neg c \Rightarrow c$), then we can find:

- When a system is inconsistent then it can prove anything. Also that it is consistent. The system can also prove that this is so. Hence it will *always* prove that it is consistent.

Ergo2D[Ergo[S, ¬ c ⇒ c], Ergo[S, c ∨ ¬ c], Ergo[S, c]]

1	$(S \vdash (\neg c \Rightarrow c))$
2	$(S \vdash (c \vee \neg c))$
Ergo	—————
3	$(S \vdash c)$

Thus the basic cause for Gödel's result is a *petitio principii*. Not by assuming something but by *not* assuming something. He *had* to use a limited kind of selfreference, otherwise he wouldn't have had a theorem. He would just have found the Liar again. One might hold that such selfreference as $(S \vdash (\neg c[S] \Rightarrow c[S]))$ is no longer an acceptable "recursive method". The proper reply would be that it still remains decent logic. Perhaps the recursionists are willing to accept a logical exception to their schemes (if it really isn't recursive).

Actually it is a bit curious that a system proves its own consistency, or that we ask such of a system, while consistency tends to be presupposed. The whole issue of consistency-proof seems to be misplaced, a fallacy of composition, see below.

The following is another example of a small extension to Gödel's assumptions that shows that the Gödeliar, properly treated, collapses to the Liar. We allow hypothetical reasoning to make the argument tractable, but see the Reading Notes in Chapter 11 for a more complex formulation:

- Assume the axiom $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$ or that when S proves something then it also proves that it proves something. Step 2 is hypothetical, is retracted in step 5 into the implication in 6. Then 1 and 6 result into a proof for $\neg g$. That is, g is not undecidable.

**Ergo2D[Ergo[S, g ∨ ¬ g], Ergo[S, Hyp: g], Ergo[S, Ergo[S, g]],
Ergo[S, ¬ g], Ergo[S, Retract: g], Ergo[S, g ⇒ ¬ g], Ergo[S, ¬ g]]**

1	$(S \vdash (g \vee \neg g))$
2	$(S \vdash \text{Hyp} : g)$
3	$(S \vdash (S \vdash g))$
4	$(S \vdash \neg g)$
5	$(S \vdash \text{Retract} : g)$
6	$(S \vdash (g \Rightarrow \neg g))$
Ergo	—————
7	$(S \vdash \neg g)$

When we interprete the system, i.e. find a model or assume semantical correctness, then this result contradicts the undecidability of g (see p208-209), and we can also derive $\neg g$. The system would still be consistent but our model causes a contradiction $g \wedge \neg g$. Thus we should not interprete the system, making it somewhat useless or inapplicable.

Note that we used $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$ already in the former section to shorten our proof for $(S \vdash (c \Rightarrow g))$. Gödel did his proof differently so that the use above was only for didactic purposes. Now, however, we may wonder why we don't include that axiom anyway, so that the system becomes strong enough to refute the Gödeliar. It may be noted that assuming proof-consequentness is weaker than deductive completeness.

`ProofConsequent[` $(S, y,) p$`]`	expresses that system S is consequent in its proof predicate, i.e. that, for all p, if p is proven, then it is also proven that p is proven

A key type of reasoning in the literature on Gödel's result is: "If already simple arithmetic cannot prove its consistency, wow, then certainly the more complex systems cannot either. Hence there are fundamentally undecidable questions, and mankind is faced with daunting questions that can only be intuited by the arts and mysteries" (no quote). This reasoning is as valid as "if a baby cannot walk then certainly an adult cannot walk". Indeed, there may be daunting questions, but not because of a nonsensical Gödeliar or a weak system that makes consistency equivalent with that nonsensical Gödeliar.

Consider Gödel's own summary: "it can be proved rigorously that in *every* consistent formal system that contains a certain amount of finitary number theory there exist undecidable arithmetic propositions and that, moreover, the consistency of any such system cannot be proved in the system" (1931:616). This is inaccurate because when you make the system stronger, the result disappears. You cannot *ex cathedra* exclude that statements in the system refer to the system *itself.*

The point thus is that Gödel hasn't assumed *enough* to make the hidden contradiction come out into the open. And it certainly does not seem rational to base our philosophical opinions on the use of such weak systems. Mathematicians might object that they are not concerned with philosophy and merely prove the theorems that such systems (called weak or whatever) have such properties. This is not true to fact. Hilbert and Gödel did intend the formal system to be interpretative of arithmetic and they implied philosophical notions on proof theory.

A more adequate solution is: either don't form such liars or use three-valued logic. Hilbert's coding of formulas into numbers was a scheme to allow for selfreference but, such coding does not come prior to logic itself. And it is twisted if you interprete the coding such that selfreference to the whole system is excluded. Other ways to create selfreference is to write a sentence on a wall that refers to that sentence on the wall. You can allow for such selfreference, except for thoses instances where such coding results into inconsistencies or improprieties in the interpretation, for which cases you then must make amends (e.g. three-valued logic).

The coding of statements into numbers was intended to allow for selfreference. How balanced is that ? Generations of logicians since Tarski and Gödel have taught: don't construct the Liar because of the levels in truth, but you are allowed to construct the Gödeliar since there we don't need levels. If Gödel had not been allowed to construct his sentence, i.e. when syntax requires that levels also apply to the notion of proof, then he wouldn't have had a theorem. There are few logicians who are fully aware of this, and that it is strange to demand levels for truth but not for proof.

A final base for rejection is to note the point that we *interpreted* the formal system. When these are just numbers then there is no paradox. Once we provide an interpretation so that it becomes semantically complete, becomes a scientific system, then we also need an interpretation for the Gödeliar in reality. The only useful interpretation is the Liar. Gödel cannot say that, cannot make that interpretation, because he calls the Gödeliar *true* while in his two-valued logic he cannot form the Liar on pain of contradiction. An approach out of this autism is to identify both constructs as nonsensical, and apply a three-valued logic for the sake of those who wonder what "nonsensical" means.

By itself it would be a good convention anyway to regard something as nonsensical in a smaller system as well, if it turns out to be nonsensical in a larger encompassing system.

To close, a note is required on Church's "thesis". DeLong (1971): "In 1936 Alonzo Church proposed that we identify the intuitive notion of an effectively calculable function [or decidable predicate] with the mathematically exact notion of a general recursive function [or predicate]" (p186) and "Church's thesis is an empirical thesis, not a mathematical theorem. It is a claim which is subject to confirmation or refutation by empirical methods." (p195). Then: "Similarly, when Gödel proves that there are no undecidable formulas, or Church that there is no decision procedure, (...) all this comes to is that [their conclusions follow] *using the means they have selected* (just as Euclid selected a straightedge and compass) (...) it just suggests that mathematicians and logicians must look for other means. (...) This objection would have a great deal of sting were it not for one circumstance: *There do not appear to be any other means.*" (p193) Our reply would be, well, *yes,* there are other means. Allow statements in the system to discuss the system as a whole and include axiom $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$. It is immaterial whether this would still be "general recursive" or not; my impression is that it probably could be identified to be so, but, if not, we might use a term "general recursive plus something". As scientists we can definitely agree with "Similarly a human computer, *qua* computer, is subject to the same limitation" (DeLong (1971:199)). The argument is not on such limitations. The point merely is that the computer shouldn't be crippled as it has been.

Alternatively, you might regard all this as an existence proof that three-valued logic is required. As long as you don't confuse "undecidable" with "mysterious" or "above

comprehension" but stick to the notion that the Gödeliar is just a cousin of the Liar, and just as nonsensical so that it doesn't provide ground for philosophical notions. The existence proof actually has already been given, see our treatment of the Liar for two-valued logic. The formal mathematical position that "truth" does not exist, only a notion of "proof", is hardly acceptable from the position of a scientist who deals with the world on a every day basis. But when the mathematicians need their own existence proof for three-valued logic then they now have it.

9.3 Discussion

9.3.1 In general

Mathematicians have often experienced that human knowledge is limited. For example in the field of arithmetic many problems can be posed, but nobody is able to solve them - at the moment. This may give rise to the question: do there exist problems that are insoluble *in principle* ? Answering a question like this requires a good definition of what methods are allowed, for sometimes people claim to know answers which are not acceptable in mathematics or science. But even for something like the oracle of Delphi we might design a procedure to check up on the facts just to verify that the oracle is reliable. Spurred by Hilbert and Brouwer, mathematicians started to better define their ways of proof. The recursive method or mathematical induction became almost universally accepted and the "transfinite" methods may still be rejected by some.

9.3.2 Ever bigger systems ?

Gödel (1931) uses a specific kind of notion of proof: it is system-dependent. His method relies on a system *S* that has a *cripple* proof-predicate, and the insolubility only means unsolvable-with-respect-to-the- *cripple* -proof-predicate-of-*S*. This directly causes the question about a larger *S.* A proof of consistency of *S* by *S* itself might be less convincing, and we might rather want to see a stronger *S** that not only shows the consistency of *S* but that might also prove some of its unsolvable problems. Gentzen indeed presented a proof of consistency of arithmetic using transfinite methods.

And for this *S** again a stronger encompassing system, etcetera. Indeed, how do we prove the consistency of Gentzen's method ? Using a stronger system does, in itself, not imply selfreference of that system. We might end up with an omniscient *S,* and when we require semantic and deductive completeness then we end up with truth again, instead of provability, where we can regard reality as the big calculation machine that continuously proves theorems; and here there would be selfreference unless types have been introduced somewhere along that encompassing process. Eventually we return to

points that we cannot prove and that we must accept as axioms. It might be conceivable. In all this, the fall-back position of a universal predicate of proof still is 'the following of conclusions from premisses' which is generic to it all.

Rather, though, we may try to find reasonable assumption which allow us to determine the hidden contradiction in the Gödelian analysis but which assumptions are not as heavy as the assumption of omniscience. In particular, we want S not to be *cripple,* but to be as strong as we are (or the best side of us, free of contradiction), so that whatever we can assert about S can also be asserted by S. The reason for this suggestion is obvious: when S is as we are, then we can use S as our image, and by studying S we can gather knowledge about what we can do ourselves. One consequence of this approach is: allow $S \vdash (\neg c[S] \Rightarrow c[S])$ hence $S \vdash c$ hence the Gödeliar collapses to the Liar and we have a contradiction. We can allow for this situation by identifying the Gödeliar "This statement is not provable" as nonsensical, in three-valued logic.

One reason to adopt selfreference within a system (so that also $S \vdash (\neg c[S] \Rightarrow c[S])$ is allowed) is that otherwise we create two logics, one non-selfreferential within the system and one outside the system that clearly is selfreferent. Selfreference does not imply omniscientia and can be limited to what is known or can be accepted.

This analysis is *not* that Gödel analyzed only numbers and that he should have analyzed humans working with numbers. It is just that we require selfreference, as Gödel suggested that he would use but didn't use.

9.3.3 A proper context for questions on consistency

Observe that mankind is quite used to think in terms of Gödel's scheme. People discuss what other people discuss. George: "John said ... Later John said ... Then he said ..." All the while George doesn't assert much himself, only reports on what John stated. People are also aware that (John $\vdash p \vee \neg p$) is not quite equivalent to (John $\vdash p$) $\vee$ (John $\vdash \neg p$), i.e. John adopting two-valued logic does not imply that he has an opinion on everything. Questions of consistency of what John says can be relevant. But we only make it a problem if there is reason to do so.

In itself it might be a bit strange to prove consistency when a system is constructed to be consistent. For arithmetic we have: (1) the axioms of A are tautologies, (2) the deduction rules preserve that property, (3) and we even have an example that does not have that property of being a tautology (e.g. the negation of some axiom). Hilbert actually mentioned this kind of "proof" - which might better be called *construction*.

The concept of consistency is rather used for the occasion that a contradiction is found, so that we can say that the assumptions were inconsistent. The interest of some mathematicians for consistency proofs derives mainly from Hilbert's decision to put it

on his list, in a period long ago when there might have been cause for some worry. (Nowadays we would worry about results by computers.)

There are various other ways for a consistency proof, not only taking $S \vdash \neg c \Rightarrow c$. If we allow the system to state that it accepts (1) tertium non datur, (2) the axiom $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$, and (3) that it rejects a conclusion on possible inconsistency as fast as a contradiction, then the system can prove its own consistency. (Though we may value it little when a system says that of itself.)

When the system accepts TND then it cannot accept ¬ TND since this would cause a contradiction, and conversely. This has a consequence for proving consistency. Consistency is equivalent to the *nonsequitur* of the *denial* of *tertium non datur* (yes, three times *not*).

- When the system accepts TND.

Ergo[S, TND] ~$Equivalent~ Ergo[S, ForAll[p, TertiumNonDatur[p]]]

$$((S \vdash \text{TND}) \Leftrightarrow (S \vdash \forall_p (p \vee \neg p)))$$

- The key point is that "∃ .. and ..." *in* the system is the same as "∃ ... and ..." *about* the system.

$Equivalent[Ergo[S, ¬ TND] , Ergo[S, Exists[p, p ∧ ! p]], ¬ Consistent[S, p] /. ErgoRules[]]

$$((S \vdash \neg \text{TND}) \Leftrightarrow (S \vdash \exists_p (p \wedge \neg p)) \Leftrightarrow \exists_p ((S \vdash p) \wedge (S \vdash \neg p)))$$

- Taking the negation of both sides of the equivalence shows what consistency is equivalent to.

(NonSequitur[S, ¬ TND]) ~$Equivalent~ Consistent[S, p]

$$((S \downarrow \neg \text{TND}) \Leftrightarrow \forall_p ((S \downarrow p) \vee (S \downarrow \neg p)))$$

The assumptions allow a proof of consistency in the system. (See the Reading Notes in Chapter 11 for a proof in another format.)

- Hence, step 1 is TND, step 2 is $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$, step 3 introduces the hypothesis that it were to prove $S \vdash \neg$ TND. Step 4 and 5 discover that this would cause an inconsistency. Step 6 retracts the hypothesis and 7 summarizes the inference in an implication that TND implies that the system does *not* prove $S \vdash \neg$ TND. Step 8 combines 1 and 6, and 9 translates.

Ergo2D[Ergo[S, TND], Ergo[S, Ergo[S, TND]], Ergo[S, Hyp : Ergo[S, ¬ TND]], Ergo[S, Ergo[S, TND] ⋀ Ergo[S, ¬ TND]], Ergo[S, ¬ c], Ergo[S, Retract : Ergo[S, ¬ TND]], Ergo[S, Ergo[S, TND] ⇒ NonSequitur[S , ¬ TND]], Ergo[S, NonSequitur[S, ¬ TND]], Ergo[S, c]]

1	$(S \vdash \text{TND})$
2	$(S \vdash (S \vdash \text{TND}))$
3	$(S \vdash \text{Hyp} : (S \vdash \neg\, \text{TND}))$
4	$(S \vdash ((S \vdash \text{TND}) \bigwedge (S \vdash \neg\, \text{TND})))$
5	$(S \vdash \neg\, c)$
6	$(S \vdash \text{Retract} : (S \vdash \neg\, \text{TND}))$
7	$\left(S \vdash \left((S \vdash \text{TND}) \Rightarrow \left(S \not\vdash \neg\, \text{TND}\right)\right)\right)$
8	$\left(S \vdash \left(S \not\vdash \neg\, \text{TND}\right)\right)$
Ergo	―――――
9	$(S \vdash c)$

9.3.4 Axiomatic method & empirical claim

As a source of confusion ever since Lobachevsky, mathematicians tend to identify consistency with 'existence'. The 'existence of competitive equilibrium' for example merely means that some axioms result into some properties (but remain consistent); and this does not mean that our actual world satisfies these axioms. This mathematical usage is inadvisable. Valid is only that existence implies consistency, or conversely that inconsistency implies non-existence.

9.3.5 A logic of exceptions

We may formulate the rule that something is not well-defined if it allows the deduction of a contradiction. For clarity we add the axiom that this rule is well defined itself. And next to well-defined concepts we need well-defined systems (if not considered concepts) for which the condition holds.

A person trained in official doctrine is confronted with the 'paradox' that the whole issue of consistency seems to vanish. For, if we meet a contradiction then it suffices to declare that the concepts *apparently* were not well-defined - and that is all. Whenever we work with systems that are strong enough to defend themselves then it becomes a waste of time to give consistency proofs and the whole problem vanishes since these systems are consistent *by definition* (by construction).

Not surprisingly, this rule reflects the general situation in scientific research. We already work that way, and now we also require it for the formal systems that we build in logic and mathematics.

This might also clarify the need for 'consistency'. The everyday scientist sees himself or herself forced to use concepts that, if it comes to it, may actually be hard to define. The wish is to use as much sense as possible and to rely on two-valued logic as much as possible. Hence the awareness for the relevance of consistency - that we otherwise would take for granted if the system were well-defined to begin with.

Bochenski (1970:407) mentions that Wajsberg 1931 gave an axiomatization of three-valued logic. Had logicians and economists pursued their subjects with more intellectual discipline they would have made a bigger difference in trying to prevent World War II.

9.3.6 The real problem may be psychology

Mathematicians tend to say that such consequences of Gödel's theorems are merely philosophical. But this is not correct since they lose out on the math. But it may be that mathematicians actually believe at a deeper level that their systems are undecidable and that they merely use the Gödeliar to make this belief more formal. Their adoption of Gödel's theorems then satisfies a deeper psychological need, making them less wary of the consequences.

Logicians have insisted since 1931 that the argument is strictly logical, so that it has a necessary character and irrefutable consequences. But perhaps their intentions were different too. As they didn't adopt three-valued logic, they could use "undecidable" as a catch-all for nonsense without requiring the word "nonsense". Perhaps the whole discussion is only about this word.

There can be strong psychological undercurrents. Hilbert gave a strong verdict on Brouwer and ousted him from the Annals. This isn't just an extreme case of calling names on each other but it rather shows the trend. Why didn't Gödel include $\neg c \Rightarrow c$ in his system ? Since it would make the Gödeliar into the Liar. Why don't logicians do it ? Because you might have to conclude to three-valued logic. Why don't they want to take that step ? Because it deviates from tradition and convention.

And non-academic authors may have a commercial interest in stating "mystery" and "sensation" on their books.

9.3.7 Interpretation versus the real thing

The Hilbert - Gödel coding may have been helpful for mathematicians to establish that selfreference might be acceptable as a mathematical concept. Arithmetical statements

concern numbers and when these statements can be coded then arithmetic discusses itself. (To be precise: statements discuss themselves but the system itself is not allowed to do so since there is no axiom of proof-consequentness.) Whatever this historical merit, the coding is not without problem. If I give you an arbitrary number like 19540805 then it is up to you to know whether it is *just that number* or *some code*, e.g. as a recipe for days to send me presents. Thus, codes are used in contexts and a code is not *identical* to a number, just a bijection (one to one relation) within a context. Even when we count our statements then we distinguish the stuff *that we count* and the numbers that we count *with*. There is a conceptual difference between statements and numbers, which difference cannot be obliterated by a coding. The distinction is even essential when we make an interpretation of the formal system, for, precisely because we first interprete the formal system as numbers, we subsequently code the statements with numbers. What remains crucial is that the interpretation concerns statements, which then are to be judged from a logical point of view. Accordingly, whatever the historical context, the coding is just equivalent to writing on a blackboard that the statement on the blackboard is false (cq. unprovable when the system is so weak that it only has a notion of proof and not one of truth). All that is required to identify what the statement says is to remove all contextual information that makes the code specific for that context (as presumably spies know so well) so that the pure statement remains.

9.3.8 Interesting fallacies

Gödel's *reasoning* or *system* is inconsistent since he creates a model that doesn't fit his stated purposes. His *model* remains consistent since he (tacitly) introduces assumptions that leave out the crucial property.

When it is stated that Gödel's system is inconsistent then this might cause the question "If Peano arithmetic is inconsistent, how do you think that numbers exist ?" For, Gödel used arithmetic, and any inconsistency found would apply to it as well.

But this is a fallacy. An inconsistent application of arithmetic does not make arithmetic false. Joining up the Liar with something sensical does not make the sensical part nonsensical too.

There are millions of such fallacies that make discussions on Gödel's theorems horrible when people only don't want to listen.

9.3.9 Smorynski 1977

Above reproduction of Gödel's result is based upon DeLong (1971) and Smorynski (1977), the "Handbook". The latter discussion neglects the possibility of three-valued logic. Moreover, there are some needlessly rash points in that discussion. Where

Brouwer for example held that "in wisdom there is no logic" Smorynski simply neglects what Brouwer said and did, and tells us that Brouwer intended to turn mathematics into religion. Smorynski states that Gödel's theorems "still induces nightmares among the infirm" (p825). Although I do not believe that all those who had and have their doubts about these theorems are so rational, Smorynski's remark is an intolerable generalisation, and it seems to be an effort to silence criticism by sheer intimidation. This should not happen in science. Moreover, it must be that Smorynski does not quite see the meaning of Gödel's theorems. Those who accept the theorems must *believe* either the consistency of arithmetic or infinite methods, and in both cases they are just believing. In that respect they are certainly not more rational than those who believe in the hidden inconsistency of Gödel's method: for it all just seems believing. It is sad that Smorynski's words were not noted and corrected by the editors.

9.3.10 DeLong 1971

DeLong (1971:225): "The analogy of a game is useful in explaining the notion of the prospective. It often happens that a game is invented (and the rules are laid down which define that game), but at a later time a circumstance occurs for which the rules give no guidance. At this point a *decision* has to be made at to what will *henceforth* be the rule concerning that circumstance. The decision might be made on the basis of fairness, whether it makes a better spectator sport, whether it increases the danger, etc. However it cannot be made on the basis of the rules of the game because they are incompletely defined. Now part of the impact of the limitative theorems is that the rules by which we discover mathematical truth not only are, but must be, incompletely defined. We are thus forced to define the notion of arithmetical truth historically; that is, it cannot be explicated once and for all but must be continually redefined. We have seen how both Gauss and Lobachevsky came to the conclusion that the problems of truth in non-Euclidean geometry required that they go beyond the data of pure geometry. In an analoguous way we must go beyond the data of mathematics to define mathematical truth. Man has invented a game of mathematics which is incomplete apparently because of the incommensurability of man's ideals and his abilities."

PM 1. A "historical, every time changing, arithmetic truth" is too high a price. The Gödeliar paradoxes better be discarded.

PM 2. There can be such a game of ever increasing knowledge, but not because of these paradoxes. Man's ideals and abilities may not match, but not because of these paradoxes.

PM 3. "go beyond the data of mathematics to define mathematical truth" is vague.

PM 4. The issue of non-Euclidean geometry cannot be compared to the nonsensical paradoxes. These problems have an entirely different structure. It may also be noted that one can regard Euclid's system as a definition of how we understand space, so that a non-

Euclidean interpretation (e.g. on a globe) is still be explained within an encompassing Euclidean one. Einstein's interpretation of non-Euclidean space-time might primairily be a way to deal with observation problems.

PM 5. The crux of DeLong's position is not that man has limited abilities, but the limitations he sees derived from the nature of mathematics itself. This is a big vague idea, since "the nature of mathematics" is not well-defined here, but it comes across as an objectionable idea anyway, since one would hope that logic and mathematics would be pure thought and those would be free of any limitation other than our creativity and the contents of the subject matter.

PM 6. DeLong's point of view is very valuable since it makes explicit what other logicians and mathematicians often assume implicitly, mostly neglecting the consequences of their interpretation of Gödel's theorems. For one thing, this present book would not have existed if the present author hadn't benefitted greatly from DeLong's otherwise excellent book and if DeLong's explicit statement had not caused a train of thought.

DeLong (1971:227): "In other words, art is a necessary complement to science and logic. (...) we might recall that it follows from Tarski's truth theorem that no formal system is rich enough to state its own semantics. But what is the difference between an interpreted formal system and an ordinary scientific theory ? The only apparent one is that of rigor. Therefor it seems to me that this result applies to all comprehensive theories whatsoever. Any fixed comprehensive account of reality which states its own truth conditions could not possibly be true, but only mythical or fictional. No nonpoetic account of the totality of which we are a part can be adequate." (...) "What is characteristic of poetic discourse about something is that conflicting and contradicting accounts about it are permissible."

PM 1. This may be true, but not because of the paradoxes of Tarski and Gödel. Tarki's "result" is based upon a theory of types because he didn't mind losing selfreference - and he didn't opt for three-valued logic. Tarski's approach is not convincing and it is possible to define truth within the same language if you allow for the exceptions of the paradoxes that need a sink.

PM 2. A poet doesn't lie with his or her objective to have an impact. The whole scheme may be a lie, but not in touching people. Otherwise it is a bad poem.

PM 3. DeLong correctly indicates the modelling method of having a formal system and a semantic interpretation. This actually causes a question. The intended application of reality and human experience is so huge that we may wonder whether we can create all those formal systems that subsequently can be interpreted. The modelling method might be useful for key phenomena, but, for the whole of empirical reality we probably will work directly with that reality (as perceived by our brain/mind/body-complex) instead of with the use of such a formal backbone.

9.3.11 Quine 1976

Quine (1976): "Let me, in closing, touch on a latter-day paradox that is by no means an antinomy but is strictly a verdical paradox, and yet is comparable to the antinomies in the pattern of its proof, in the surprisingness of the result and even in its capacity to precipitate a crises. This is Gödel's proof of the incompletability of number theory. What Kurt Gödel proved, in that great paper of 1931, was that no deductive system, with axioms however arbitrary, is capable of embracing among its theorems all the truths of the elementary arithmetic of positive numbers unless it descredits itself by letting slip one of the falsehoods too. Gödel showed how, for any given deductive system, he could construct a sentence of elementary number theory that would be true if and only if not provable in that system. Every such system is therefore either incomplete, in that it misses a relevant truth, or else bankrupt, in that it proves a falsehood. (...) That there can be no sound and complete deductive systematization of elementary number theory, much less of pure mathematics generally, is true. It is decidedly paradoxical, in the sense that it upsets crucial preconceptions. We used to think that mathematical truth consisted in provability." (p16+)

In his baroque style Quine argues that if a baby cannot walk then surely a grown person cannot. But fortunately he contradicts himself a little bit: "I don't have time to explain how startling or why. But does it point to unanswerable questions ? It does not. No truths of elementary number theory are set apart by Gödel's theorem as unprovable. Rather, each axiom system or proof procedure will miss some of those truths; other proof procedures can cover those, or some of them, and miss others. As I already suggested in connection with the continuum hypothesis, plausibility considerations can augment existing codifications of accepted mathematical laws. Gödel's theorems show that such augmentation can never yield any one finished sysem in which every truth of elementary number theory admits of proof. But it does not show that any one truth of elementary number theory is forever inaccessible." (p66) Quine thus admits that there can be "miracles" such that a child can walk even when not being able to do so as a baby. We would rely on "plausibility considerations" outside of mathematics proper, a *deus ex machina*.

Quine is considered by many to be a serious logician.

For fact. DeLong (1971:277) on Quine's book just quoted: "Read together, they impress not only with Quine's famous elegant style, but also with his uncommonly good judgment".

9.3.12 Intuitionism

DeLong (1971:271) calls Stephen Cole Kleene's "Mathematical logic" John Wiley 1967: "In my opinion, this is the best single introduction to the techniques of mathematical

logic". Kleene also visited the intuitionist school in Amsterdam and wrote a monograph, Kleene (1965), on the overlap and differences of "classical mathematics", "intuitionism", and the "recursive methods" as used by Gödel. This is fortunate since it allows a view on intuitionism, not only by a mathematician with undisputed traditional roots but also with a razor sharp mind and gifted pen.

Kleene observes that the reconstruction of "classical mathematics" already started with Kronecker around 1880-1890. Then: "But, although intuitionism began twenty-five years before the theory of general recursive functions and so contributed to the climate of research in the foundations in which the theory of general recursive functions arose (...), the latter theory in its details developed quite independently of the intuitionism mathematics. On the other side, in the next twenty-five years after the theory of general recursive functions had appeared, intuitionism under Brouwer continued its way without taking explicit notice of the theory of general recursive functions." (p4) Thus, living apart together, or an agreement to disagree.

The key quote: "Brouwer took the position on philosophical grounds that the possibilities of construction cannot be confined within the bounds of any given formal system, long before this position received confirmation in the famous proof by Gödel 1931 that formalisms adequate for a certain portion of number theory are incomplete. So it is to be understood from the outset that our formal system for intuitionistic mathematics is not complete (...) It, or any intuitionistically correct extension of it, could be extended by the Gödel process (which is valid intuitionistically), i.e. by adding a formally undecidable but true formula (...)" (p5).

PM 1. Quine was more of an intuitionist than he might have realized.

PM 2. Though Hilbert rejected Brouwer, his Ph. D. student used Brouwer's method and proved Brouwer's philosophy true. (Though, this observation may have limited value, Gödel wrote a paper on intuitionism in 1933 which would contain his own views.)

PM 3. But we reject that Ph. D. Thesis so that Brouwer's position remains only philosophical.

9.3.13 Finsler

Finsler (1967) judged, see the comment of the editor Van Heijenoort (1967:440): "The nonformal argument by means of which Gödel's undecidable proposition is recognized as true, Finsler considers to be 'formal', since it can be expressed - and fairly simple at that - in a language. Hence, according to Finsler, Gödel has not exhibited a formally undecidable statement at all (...)" The editor refers here to Gödel's own conclusion: "Thus the proposition that is undecidable *in the system PM*, still was decided by metamathematical considerations" (Gödel (1931:599).

For clarity, that argument is:

Ergo2D[NonSequitur[S, g] ⇒ g, NonSequitur[S, g], g]

1	$((S \mathrel{\top\!\!\!\!\downarrow} g) \Rightarrow g)$
2	$(S \mathrel{\top\!\!\!\!\downarrow} g)$
Ergo	————
3	g

A twist to that argument is that the second line is only valid if the system is consistent. Otherwise everything gets proven. Hence:

Ergo2D[NonSequitur[S, g] ⇒ g, c ⇒ NonSequitur[S, g], c ⇒ g]

1	$((S \mathrel{\top\!\!\!\!\downarrow} g) \Rightarrow g)$
2	$(c \Rightarrow (S \mathrel{\top\!\!\!\!\downarrow} g))$
Ergo	————
3	$(c \Rightarrow g)$

This complication arises since our discussion above took the shortcut of using semantical completeness which would not be acceptable if we would stick to the order in which that Gödel presented things.

Yet, Finsler's point is correct that these are essentially formal deductions. We deduce these statements, they don't fall just from heaven. But Gödel forbids that *S* discusses itself, so this formalism cannot be presented in *S*. Selfreference for statements is allowed but not for *S*.

9.3.14 A plea for a scientific attitude

The ideas put forward here met with fierce disdain and contempt of logicians - I wrote in 1981. In 1982 I started my professional life as an econometrician and shelved the draft books and the issue. I haven't checked the latest situation in the field in 2006-2007 but feel safe to presume stagnation.

Looking back, one can observe some complications in communication. In 1980-1981, I was a student, students make mistakes, I made mistakes. Professors note those mistakes and they regard students as being liable to mistakes anyway. So as I worked through a series of errors I may have built a reputation of someone who makes too many of them. I followed one course in logic, and another on argumentation theory with Else Barth, in hindsight it would have been better, given human psychology, to first finish the other standard courses on logic too, show some intelligence, and only then criticize accepted wisdom. Yet I was a student of econometrics and not a student of logic, so I had no vested interest in those other courses on logic. I had only an interest in the idea that the Liar was nonsense, and I had the same intuition as Finsler later appeared to have had on

Gödel, that the "non-formal" argument must be properly formalized. Too many texts in philosophy at that time draw all kinds of conclusions from Gödel's theorems, which conclusions were non-sequiturs to a scientific mind. A scientist cannot accept the Gödelian inference: *ex nonsense, philosophy !* It must be emphasized that teachers did help to resolve many of my mistakes and thus guided me towards a better understanding of what was involved. Also, three-valued logic was not a natural conclusion at first, and when I asked for it the professor gave me his copy of Kripke's article, and that helped getting more clarity. There are the eternal issues of terminology, like that I opted to take "*L* = "*L* is false"" as the Liar rather than *L* itself, which raises eyebrows for who do the latter (but is fair if you want to emphasize selfreference and slash it). But with three-valued logic I could yield to that latter convention, since three-valued logic allows a better expression of nonsense than merely slashing the equality sign. By then it had become difficult of course to report back to those who had grown wary. Visits to other universities in the country helped a bit, but also caused awkward discussions by phone, without the availability of the spectrum of communication and a blackboard.

When I sent in a paper to the *Journal of Symbol Logic*, an editor reacted: "this (...) paper seems to indicate that you may have studied some logic but got many of the points you have read or heard about mixed up". But the paper solves the issue of the Liar - reproduced in this book - so that the rejection and intolerance are unwarranted. One should rather publish such a paper than all current nonsense put in print. (But of course, this is a plea for tolerance so I wouldn't argue against publication of nonsense ...)

Whatever the circumstances that cause some justification for logicians, I still hold the position that it is rather impossible to have logicians listen to arguments. At no time during these discussions it was granted that (a) there are systems that become inconsistent by the Gödelian Liar, (b) hence the Gödelian theorems are not true for all systems; which was what I contended; (c) this inconsistency can be solved by three-valued logic; which I subsequently contended. There was no reaction to the solution of the Liar except a perhaps polite "it is one of the possibilities". When a reaction changes from "you say that the Earth is flat" into "you give another interpretation to the matter" without further substantiation then this indicates a form of politeness and implies that one does not buy that interpretation - while the proper reply should be either to state that the argument is valid or show where it is invalid. It was a *discussion*.

When Colignatus (1981), "In memoriam Philetas of Cos et alii" (unpublished) - on content no different from this book - was handed in for a grade on a course of logic, the professor reacted: "(...) again that mixture of not quite false assertions and considerations which present all kind of essential logical things in a wrong and

misleading manner. As long as somebody does not go really into the subject of Gödel's (own) analysis (e.g. the actual construction of the Liar sentence, and the like), then possible suggestions about alternatives remain premature and vague. Against this background I believe that I do no service to the academic world (and neither to you) by giving the asked official honouring" (April 6 1981). Again this is just a verdict and no substantiation. It is also empirically false since I had looked up the 1931 article and Colignatus (1980) contains such a Gödelian coding - the professor didn't even ask whether I had done so - (though Colignatus (1981) dropped it), and it is a non-sequitur since it is logically not required to use such a coding (which is why Colignatus (1981) dropped this, as in this book). (Funny is this: I did a normal test, filled in the nonsense answers where they were expected, got the grade anyway, but then a professor of econometrics started protesting that mathematical logic was no serious math for econometrics. The dean of the math dept. professor Stam had to write a letter to the graduation committee that mathematical logic was serious math. Which one of course can doubt, with reason, given the errors mathematical logicians have been making since 1931. PM. Econometrics is not a specialization, as some think, but a generalization: it concerns the intersection of economics, math and statistics, so that one can do all - with the perpetual modesty on one's grasp of course.)

A referee of a journal states: "It is one of those confused things which contain a mixture of ingenuity, formal mistakes and outright potty bits." The referee gives some arguments but these are absolutely false. For example: "he's happy to shift to four values, and so on, for as long as it takes." The editor apparently agrees and reports: "It seems that the paper is not up to the standards of rigor and sophistication expected from contemporary work in philosophical logic." (June 15 1981)

I would like to conjecture that all this is not the proper way to treat people. It may seem that people make errors but then one should still try to understand what actually is the main idea behind what is said. Certainly in logic and science one might be considerate, since when logic forces a person to consider and accept what other people consider "outright potty bits" and "misleading" then, well, you wouldn't want to be put in the position to be treated like that. One would like to plead for an open atmosphere and a truly scientific attitude, not one that merely is a façade.

Though modern times have become more agreeable to people studying logic, logicians haven't caught up yet. It were external factors that made Aristotle flee from Athens, made Ockham spend four years in prison (and die of the plague at 47 years of age), and that allowed modern logicians to live longer lifes, Russell 98, Brouwer 85, Gödel 75. It were internal factors to logic as an academic subject that made logicians neglect Frege's "Begriffschrift" for more than 10 years, that made them misinterprete Brouwer, that

caused them to neglect Finsler (1926, 1967) who wrote 5 years before Gödel, developed most of that same analysis, and who ends up with a much sounder position. These internal factors have not changed and must be anticipated by anyone aspiring to contribute to the development to logic as a subject.

The bias and disbelief that one will encounter will make one feel as a busdriver who suddenly finds oil on the road, while his passengers start to bump on his head and try to catch the wheel, while he discovers that it were the passengers who threw the oil on the road. They simply won't listen and are too preoccupied with pushing and pounding.

They are not scientists who consider an argument but they are bureaucrats who check where an argument deviates from accepted wisdom and then punch in the standard button. They live by authority, disbelieve and bias, not by imagination and inquest. If they are willing to enter into a discussion, it is only so that they take the time to emphasize the uniformity of opinion since 1931 onwards.

They build enormous castles in remote barren lands, like the theory of types and the limitative theorems, and they cannot explain clearly what they do. Had they provided that clarity, this book wouldn't have been necessary. This book provides a clarity that you will not encounter elsewhere - provided of course that logic has been happily stagnant as I presume.

These are not generalizations, these are facts. As a scientist I respect the facts. I look them straight in the eyes and when they don't go away then they are just there. Show me that it is different and I am willing to retract.

These ideas on the Liar and the Gödeliar in 1981 are perhaps not the key issue. What also happened back then is that I also developed this more general idea of a *logic of exceptions,* but had no urge whatsoever anymore to pursue that idea further. The intellectual climate for other ideas apparently was sick and it would only testify of masochism to keep exposing one to sick minds. It was not only good and necessary that I could concentrate on econometrics again, but it was also a relief. But that also meant that this concept of a *logic of exceptions* was shelved. Over the years, moving the boxes with the notes and draft books on logic and the methodology of science along with the rest from one new home to another, the impression grew that this notion might be of more general value. In economics, the subject of public administration and the role of government bureaucracies is important, and one will be aware that a bureaucracy must maintain rules. It will contribute to the general welfare when policy makers and bureaucrats grow aware and are taught in the very first years of their education (if they get any) that rules might require some exceptions to allow for the vagarities of life and human decency.

The second element in this whole is the creation of *Mathematica* by Stephen Wolfram, inspired by the program *Schoonschip* of Martinus Veltman, that, for all practical

purposes, makes all the difference. Hence it is with a different angle and refreshed joy that the ideas of 1980-1981 have been implemented in *Mathematica* now. The logicians don't come into play, though as a consequence a next generation gets a chance to show itself.

PM 1. In February 2007, correcting the typing errors in the January 2007 print test run, I can add the following. I contacted the professor of 1980-1981, informed him of the PDF and software on the web, suggested that the software might make a difference so that he might better understand the analysis, added that my check of the analysis confirmed it again, invited him to try again, and offered to give him a password for that software to do so. He didn't react. It also appears that he has been publishing a lot on dynamic logic and getting huge research grants for it. In 1980-1981 this was the only thing that he said that he liked in my papers - and when we discussed it back then it was also obvious that he hadn't thought about that angle himself. I don't know whether he gives proper reference now. It is useful for me to add this comment otherwise people might think that I don't give proper reference to him. For the record: the distinction between statics and dynamics comes from economics and in 1980 it felt like a natural interpretation for me to apply it for the distinction between propositions and inference. It is only a small element in the whole of this analysis.

PM 2. I also contacted the other main person I discussed this analysis with back then. This person reacted with the statement that he didn't agree back then so that it would be useless for him to try and look again. Thus 25 years difference and a professional career in econometrics don't seem to count. And these logicians teach psychologists about what rational thought is ... (how rationality can be reconstructed afterwards out of (seemingly) irrational behaviour).

10. Notes on formalization

10.1 Introduction

As said in the Overall Introduction, there exists an empirical model for what is developed in this book, and hence it can be formalized.

PM. In case someone doubts whether statements in a system can refer to the whole system, we might always take a human person as an example. That a human person is fallible in other respects does not disprove the possibility of such selfreference. (If we already had something perfect then we wouldn't need logic and mathematics; instead we construct something better from patching the best aspects from the imperfection around us.)

Originally, Colignatus (1981) went a bit further in formal development. For the current discussion we return to the concept of an *Introduction to Elementary Logic* and there is less need for it. This chapter however can be useful to show the backbone of a formal exposition. It can be a guide where to look in the literature for the points of interest.

10.2 General points

The book drops quantifiers in presentations when they read ugly, look pedantic and do not essentially contribute to an argument. Jaakko Hintikka's remark that a quantifier always requires a domain however is strictly adhered to, even when the quantifier is not stated. The transformation rule for the "quantifier free logic" is that a constant or $\wedge$ stands for a "there is" while a variable or $\Rightarrow$ stands for "any". Thus $(x \in D) \wedge (x \in A)$ indicates the intersection, while $(x \in D) \Rightarrow (x \in A)$ indicates that being a D is a sufficient condition for being an A. Quantifiers pop up when there is danger of confusion. Real existence would be indicated by $x \in \mathbb{U}$, where $\mathbb{U}$ is the universal set. This is either defined locally to the discussion or might consist of basic objects at the lowest level (minding Cantor's powerset). PM. There exists a reference, that I lost, on a formalization of quantor free logic.

With $\mathbb{S}$ the set of sentences and "$p \in \mathbb{S}$" a sentence again then we get similar issues like those of the powerset so that one might question whether we really can refer to $\mathbb{S}$ as a

totality or whether such a totality is properly defined. But $\mathbb{S}$ can also be defined as strings of symbols that can be interpreted, and there seems no harm in to interprete one such symbol $\mathbb{S}$ as the collection of all such strings (where it is used as such).

10.3 Liar

Theorem 1: In a sufficiently rich system of two-valued logic there does not exist a Liar. Thus the Liar has no truthvalue True | False.

Proof: See the body of the text that proves that $\forall$ p in $\mathbb{P}$: $p \neq$ "w[p] = 0".

Corollary: In a sufficiently rich system of two-valued logic there is no need for specific formation rules forbidding the formation of the Liar.

PM. This requires the distinction between just forming a sentence and determining whether it can be asserted.

Theorem 2: The Liar arises from not adhearing to the Definition of Truth but adopting an instance of the Alternative Definition of Truth. It is not necessary to look for further explanations of the contradiction that would occur if the Liar is introduced.

Proof: See the body of the text that shows that the Liar is an instance of the "alternative definition of truth", and that the definition and its alternative are mutually exclusive.

Corollary: (This is essentially Occam's razor again.) The 'levels of language' of Russell and the 'undefinability of the concept of truth' are not required to solve the Liar. Similarly Kripke's explanation that the Liar is 'not grounded' is irrelevant. Since the problem of the simple Liar can be solved by very simple logical methods, the occurrence of the Liar cannot form an argument for the introduction of the concepts of Russell et al. and an introduction of these concepts would have to be based upon other considerations, which would rather not be of a logical nature.

PM. Systems of statements (Buridan) can via substitution be shown to be Liar-loops and may be broken anywhere. One might also discard the whole system that contains that loop. A general procedure is to create a hierarchy of concepts with a (growing) core of statements that are accepted and a fringe of statements up for rejection.

Theorem 3: Although any p satisfying the relation p = "w[p] = 0" cannot belong to the set of two-valued statements $\mathbb{P}$ one stil has the freedom to include it in some $\mathbb{S}$.

Proof: Either this is true by definition of those sets or if only $\mathbb{P}$ is given then it can be extended.

Corollary: The Liar is perceived as paradoxical since one believes that $\mathbb{S} = \mathbb{P}$, which however cannot be.

PM 1. This theorem seems innocuous but appears to be required because a person like Beth (1959) claims that the Liar "exists". He gives the example of a sentence that by context refers to itself. It is difficult to slash a context. An scientist who must observe facts would be inclined to accept that a sentence on a wall exists and refers to itself (for sure when it says it is "true" so also when "false"). Hence Theorem 3 is an existence theorem for the complex Liar. It also covers Buridan's loops.

PM 2. A computer program that merely substitutes would create an endless loop. When at input the recognition of p = "w[p] = 0" (simple Liar) is not possible (conceivably it is) then for the complex Liar we have (a) a counter and exit condition, (b) at intermediate steps a collection of results and check whether both p and $\neg p$ occur, (c) read and operate on the own program, input and intermediate results. When the number of variables and size of loops is larger than memory then (a) and (b) might not suffice. But there arises a grey area now between practical problems and the principles involved.

Theorem 4: For a language $\mathbb{S}$ a three-valued logic is necessary.

Proof: By Theorem 1 the Liar has no truthvalue. But there is a complex Liar for which we want one. Hence a third value.

Theorem 5: For language $\mathbb{S}$ another truthfunction is needed than for $\mathbb{P}$.

Proof: The domains and ranges differ.

PM. Thus next to the Definition of Truth there is also one for three-valued logic. The truthfunctions are defined as $w : \mathbb{P} \to$ {True, False} or $w : \mathbb{P} \to \{1, 0\}$ and $W : \mathbb{S} \to$ {True, False , Indeterminate} or $W : \mathbb{S} \to \{1, 0, \frac{1}{2}\}$, with $\mathbb{P}$ the expressively complete set of propositions and $\mathbb{S}$ the sentences. There is only one embedding that fully puts w in W. Though the domains and ranges differ we take the liberty to still consider them to be the same function and to use the same symbol $W = w$. But the text allows for the "change of perspective" when hypothetical reasoning switches from two-valuedness to three-valuedness. The body of the text on hypothetical reasoning is important for an adequate interpretation.

Theorem 6: If three-valued logic is also sufficient for language then Tarski's conjecture that there does not exist an adequate definition of truth for language is incorrect.

Proof: The discussion in the text shows that (a) a DOT3 is possible, (b) that it receives proper assertoric interpretation, (c) that it has reasonable connectives and compound statements.

For sufficiency, we only have to show that it does not generate new paradoxes, i.e. that DOT3 is sufficient.

Lemma 1: We can discuss $\mathbb{S}$ in $\mathbb{P}$, and use the DOT3 to do so.

Proof: Obvious. See the text for hypothetical reasoning and the dynamic use of contexts, also applicable for the words true and false. While it seems that we replace DOT3 with DOT (when becoming two-valued again) then this isn't really true since DOT is contained in DOT3.

PM. The Liar is not a problem of two-valued logic but one of three-valued logic.

Lemma 2: The application of the DOT3 on $\mathbb{S}$ does not result into contradictions.

Proof: See the body of the text. For all p there are sufficient statements in two-valued logic to consistently discuss its (potential) nonsensicality.

PM 1. With "This statement is false or senseless" DeLong (1971:242) holds that the option that is is false would make it true, because it states such; and if true it cannot be senseless. This is an important point, since rejection of three-valued logic opens the road to Gödel's approach. Check the derivation in the main body of the text on the Liar for three-valued logic, and find that DeLong errs here.

PM 2. Since $\forall\, p$ in $\mathbb{P}$: $p \neq$ "w[p] = 0", EFSQ supports hypothetical reasoning.

Lemma 3: We can define $\mathbb{S}$ such that all that can be said about $\mathbb{S}$ also belongs to $\mathbb{S}$. In other words, Tarski's 'meta-language' is not necessary and the theory of types applies not (rigidly) to language.

Proof: While a formal proof would involve the listing of the formation rules it is sufficient for this book to give the intensional definition that $\mathbb{S}$ is simply defined as such that, if something can be said about $\mathbb{S}$ then it can be said in $\mathbb{S}$. Such a definition certainly is acceptable (with the proviso that it must remain English).

PM. Language is no set, so that Cantor's powerset theorem does not apply.

Theorem 7: The DOT3 is sufficient for language ($\mathbb{S}$).

Proof: From lemma 1 and 3 follow the expressive completeness. All statements can be expressed, and valued by three-valued logic. Lemma 2 gives consistency. There are no other blocking criteria, from which sufficiency follows.

Theorem 8: The DOT3 is necessary and sufficient for language ($\mathbb{S}$).

Proof: Combine theorem 6 and 7.

10.4 Predicates and set theory

Theorem 9: Three-valued logic is required for predicates and set theory.

Proof: See the body of the text. Though we can prevent a contradiction of Russell's paradox by using his Zermelo cousin, it is needlessly restrictive to forbid Russell's

expression merely by formation rules.

PM 1. To show the existence of a set it is not sufficient to show an element, but also the consistency of the definition.

PM 2. Some theory of types can still be useful given Cantor's powerset.

10.5 Intuitionism

Theorem 10: Brouwer's "intuitionistic logic" is only adequately represented by a (specific) combination of modern non-intuitionistic methods.

Proof: See the body of the text. If we assume the dichotomies of (1) truth and falsehood, (2) necessity and contingency, (3) proof and absence of proof, (4) sense and nonsense, then Brouwer's problem can be adequately represented. If we do not use these dichotomies then serious interpretational problems arise, in particular with the word "not".

Corollary: Heyting's axiomatization of "intuitionistic logic" can only be properly understood as a deductively incomplete axiomatization of two-valued propositional logic. (Proof: it cannot represent *tertium non datur*. If it is added to the system we get the classical logic.)

10.6 Proof theory and the Gödeliar

Theorem 11: There exists a semantically correct system for which the proof predicate subsumes to the truth predicate, such that the Gödeliar collapses to the Liar. And three-valued logic is necessary to deal with that.

Proof: The model is the world. Nature. See the section on the Liar for three-valued logic.

PM. This is only to remind us that mathematicians tend to forget about the world. They don't see it as a 'model'. It is also to remind us that the Gödeliar is a cousin of the Liar and just as senseless.

Theorem 12: If we have a sufficiently developed system S such that (i) it contains two-valued propositional logic, (ii) it can discuss itself and its assertions, (iii) it contains the transformation rule $\forall\ p$ in $\mathbb{P}$: $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$, then the Gödeliar reduces to the Liar again, with all its consequences, and the Theorems of Gödel may only be EFSQ.

Proof: See the body of the text.

Corollary: If the system doesn't satisfy conditions (ii) and (iii) then it may prove weird statements like $c \Leftrightarrow g$.

Corollary: Useful systems of some sophistication must be three-valued.

11. Reading notes

11.1 Introduction

The following notes are supplementary to the expositions above and may be of use to put the earlier into perspective.

11.2 Prerequisites in mathematics

This book discusses logic and inference and does not develop the theory of functions and relations. Students using this book will generally have had some mathematics before and will tend to understand its use here. Examples of such use are the *one to one* function (bijection) and *transitivity*. If there appears some deficiency in the background then the internet will provide ample explanation. Some logic books also develop the (basic) theory of functions and relations but the idea here is that we want to concentrate on logic and inference so that those issues may best be treated elsewhere, as would also hold for geometry and its axiomatics and analytical geometry. It might be argued that reasoning with orderings (John is an ancestor of Carl and Carl is an ancestor of Maggy, thus ...) might be a special type of predicate logic, deserving attention in a book on Eelementary Logic as well, but the 250 pages of this book already appear to make for sufficient unity so that orderings can better be dealt with elsewhere.

11.3 Logical paradoxes in voting theory

This author has been using DeLong (1971) "A profile of mathematical logic" with great delight. Professor DeLong also wrote a book on Arrow's theorem in voting (1991) see Colignatus (2008) and was working on "Jeffersonian Teledemocracy" (1997). The voting paradoxes merit mentioning since voting theory is another area with confusion, see Colignatus (2001, 2007, 2011), "Voting theory for democracy" and Colignatus (2007i) on the Penrose weighing scheme. Logic and democracy are topics that deserve our attention.

11.4 Cantor's theorem on the power set

11.4.1 Introduction

Sets A and B have "the same size" when there is a bijection or one-to-one function between them. Above we adhere by default to "Cantor's Theorem" that a set is always "smaller" than its power set (set of its subsets), see pages 131, 196 and 231-235. Up to July 2007 this author thought that this theorem holds in general, but then I read Wallace (2003) and wrote Colignatus (2007j), concluding that the theorem holds for finite sets and the denumerable set of natural numbers, but not in general. This lack of generality would allow us to speak about a "set of all sets". An addendum in 2011 is that I now also reject the diagonal argument for the denumerable set of natural numbers: which opens the possibility that $\mathbb{N}$ and $\mathbb{R}$ might have the same size.

This book is on logic and inference and thus keeps some distance from number theory and issues of the infinite. Historically, logic developed parallel to geometry and theories of the infinite (Zeno's paradoxes). Aristotle's syllogisms with "All", "Some" and "None" helped to discuss the infinite. Russell's set paradox clearly deals with infinite sets and Gödel's theorems use arithmetic with its infinite numbering to establish selfreference. Yet, to develop logic and inference proper, it appeared that this book could skip the tricky bits of number theory, non-Euclidean geometry, the development of limits, and Cantor's development of the transinfinite. Though it is close to impossible to discuss logic without mentioning the subject matter that logic is applied to, we still kept a decent distance from those subjects themselves. Also when we discussed mathematical induction, and thus basically relied on the natural numbers, we did so only in complement to induction in general, and with an emphasis on the point that deductive inference becomes "inductive" if one allows hypotheses of that kind.

It must also be observed that this author is no expert on Cantor's Theorem. We may reject the proof given by Wallace (2003) but perhaps there are other proofs. A marginal check on the internet shows that this proof is the only one given at some sites that seem to matter but this may only mean that it is a popular proof. Thus for now it seems optimal to give a short discussion of said proof of Cantor's Theorem and its current rejection, and further refer to Colignatus (2007j) and future discussion.

11.4.2 Cantor's theorem and his proof

Cantor's Theorem holds that there is no bijection between a set and its power set. For finite sets this is easy to show (by mathematical induction). The problem now is for (vaguely defined) infinite set A such as the natural or real numbers. The proof (in

Wallace (2003:275)) is as follows. Let f: $A \to 2^A$ be the hypothetical bijection between A and its power set. Let $\Phi = \{x \in A \mid x \notin f[x]\}$. Clearly Φ is a subset of A and thus there is a $\varphi = f^{-1}[\Phi]$ so that $f[\varphi] = \Phi$. The question now arises whether $\varphi \in \Phi$ itself. We find that $\varphi \in \Phi \Leftrightarrow \varphi \notin f[\varphi] \Leftrightarrow \varphi \notin \Phi$ which is a contradiction. Ergo, there is no such f. This completes the currently existing proof of Cantor's theorem. The subsequent discussion is to show that this proof cannot be accepted.

11.4.3 Rejection of this proof

Similar like with Russell's set paradox (see above) we might hold that above Φ is badly defined since it is self-contradictory under the hypothesis. A badly defined 'something' may just be a weird expression and need not represent a true set. A test on this line of reasoning is to insert a small consistency condition, giving us $\Phi = \{x \in A \mid x \notin f[x] \land x \in \Phi\}$. Now we conclude that $\varphi \notin \Phi$ since it cannot satisfy the condition for membership, i.e. we get $\varphi \in \Phi \Leftrightarrow (\varphi \notin f[\varphi] \land \varphi \in \Phi) \Leftrightarrow (\varphi \notin \Phi \land \varphi \in \Phi) \Leftrightarrow$ *falsum.* Puristically speaking, the Φ defined in 11.4.2 differs lexically from the Φ defined here, with the first expression being nonsensical and the present one consistent. It will be useful to reserve the term Φ for the proper definition and use Φ' for the expression in 11.4.2. The latter symbol is part of the lexical description but does not meaningfully refer to a set. Using this, we can also use $\Phi^* = \Phi \cup \{\varphi\}$ and we can express consistently that $\varphi \in \Phi^*$. So the "proof" above can be seen as using a confused mixture of Φ and Φ^*.

It follows:

1. that the proof for Cantor's Theorem (i.e. as used above) is based upon a badly defined and inherently paradoxical construct, and that this proof evaporates once a sound construct is used.
2. that the theorem is still unproven for (vaguely defined) infinite sets (that is, this author is not aware of other proofs). We would better speak about "Cantor's Impression" or Cantor's Supposed Theorem", i.e. to link up to the literature that maintains that "theorem". It is not quite a conjecture since Cantor might not have done such a conjecture (without proof) if he would have known about above refutation.
3. that it becomes feasible to speak again about the "set of all sets". This has the advantage that we do not need to distinguish (i) sets versus classes, and/or perhaps as well (ii) "all" versus "any".
4. that the transfinites that are defined by using Cantor's Theorem evaporate with it.
5. that the distinction between the natural numbers $\mathbb{N}$ and the real numbers $\mathbb{R}$ rests (only) upon the specific diagonal argument (that differs from the general proof). See Colignatus (2007j) for this aspect. However, an addendum here in 2011 is that we

can now clearly reject that diagonal argument.

Ad 5) Consider the decimal numbers between 0 and 1 in binary expansion. A first step gives {0, 1}. A second step gives {0.0, 0.1, 1.0}. A third step gives {0.00, 0.01, 0.10, 0.11, 1.00}. A fourth step gives {0.000, 0.001, 0.010, 0.011, 0.100, 0.101, 0.111, 1.000}. Etcetera. For each n there is a finite list. Letting $n \to \infty$ gives all numbers between 0 and 1. (It is optional to also define that each number ending in a 0 has infinite zeros, so that it is unique; and then each n adds a list of new values. In this manner a stable function $\mathbb{N} \to \mathbb{N}$ can be created.) For each n a diagonal is defined. However, for the limit situation there is no proper definition of a diagonal. Hence Cantor's diagonal argument cannot be applied. It presumes something that is not properly defined. PM. Perhaps there is another argument though that $\mathbb{R}$ is not denumerable.

Since we are focussed on logic and not on number theory it suffices to stop the discussion here.

11.5 Paradoxes by division by zero

(PM. This section was written for the first edition of 2007 and has now been developed in *Conquest of the Plane* (2011). The present text remains a useful summary.)

Dijksterhuis (1990) suggests that the Ancient Greeks did not develop algebra (and subsequently analytical geometry) since they used their alphabet to denote numbers so that $\alpha + \alpha = \beta$ already had the meaning 1 + 1 = 2, and whence it would be less easy to hit upon the idea to use α as a variable. We too would consider it strange to use e.g. 15 as a variable ranging over $-\infty$ to $+\infty$. This explanation is not entirely convincing since the Greeks did use names like "Plato" or "Aristotle" and thus might have used a name to denote a variable (like "Variabotle"). But notation clearly was one of the obstacles to overcome. Western math had to wait till 1200 AD before the zero came from India via Arabia together with the Arabic digits (where both "zero" and "cipher" are jointly derived from the Arabic "sifr" = "empty"). Arabic numerals are easier to work with than roman numerals, e.g. try to divide MCM by VII, yet this advance came with the cost that the zero caused a lot of paradoxes. Western math solved most problems by forbidding division by zero but the problem persisted in calculus, where the differential quotient relies on infinitesimals that magically are both non-zero before division but zero after it. Karl Weierstraß (1815-1897) is credited with formulating the strict concept of the limit to deal with the differential quotient. Yet even there the limit of e.g. x / x for $x \to 0$ is *said* to be defined for the value $x = 0$ on the horizontal axis but is not defined for actually *setting* $x = 0$ but only for x getting close to it, which is paradoxical since $x = 0$ would be the value we are interested in. Colignatus (2007k) suggests to distinguish the passive

division result from the active division process. In the active mode of dividing y by x we may first simplify algebraically under the assumption that $x \neq 0$ while subsequently the result is also declared valid for $x = 0$ (i.e. extending the domain and not just setting $x = 0$). Mathematicians will tend to regard y / x as already defined for the passive result without simplification, so that the active notion would be new and for example denoted as y // x. Others, i.e. people who aren't professional mathematicians, will tend to take y / x as an active process and they might denote y // x for the passive result. All in all, it would not matter much, since we might continue to write y / x and allow both interpretations depending upon context. In that way such paradoxes are explained by confusion of approach and by the point that mathematicians can be less observant of what people tend to do. Another conclusion is that calculus might use algebra for the differential quotient instead of referring to infinitesimals or limits.

To make this strict, let y / x be as it is used currently by mathematicians and y // x be the following process or program: {assume $x \neq 0$, simplify the expression y / x, declare the result valid also for the domain $x == 0$}. This definition only holds for variables, so that x // x = 1 but not for numbers, e.g. 4 // 0 generates 4 / 0 which is undefined. The derivative deals with formulas too, and not just numbers, and would be the program $f'[x] = df / dx \equiv$ {Δf // Δx, then set $\Delta x = 0$} which uses both that Δf // Δx extends the domain to $\Delta x = 0$ and that the instruction "set $\Delta x = 0$" actually restricts the result to that point. In a way, this definition is nothing new since it merely codifies what people have been doing since Leibniz and Newton. In another respect, the approach is a bit different since the discussion of "infinitesimals", i.e. the "quantities vanishing to zero", is avoided. Also, the interpretation given by Weierstraß and codified in the notion of a limit is rejected since that definition of the limit excludes the value $\Delta x = 0$ which actually is precisely the value of interest at the point where the limit is taken.

The true problem is to show why this new definition of df / dx makes sense. Let us create calculus without depending upon infinitesimals or limits or division by zero. Let $F[x]$ be the surface under $y = f[x]$ till x, for known F and unknown f which is to be determined (note this order). Then the change in surface is $\Delta F = F[x + \Delta x] - F[x]$ and clearly $\Delta F = 0$ when $\Delta x = 0$. An approximation to the surface change is for example Δx ($y + \Delta y/2$) with $\Delta y = f[x + \Delta x] - f[x]$. The error of such an approximation will be a function of Δx again. We can write ΔF in terms of $y = f[x]$ (to be found) and a general error term $\epsilon[\Delta x]$, where the latter can also be written as $\epsilon[\Delta x] = \Delta x\ r[\Delta x]$ where $r[\Delta x]$ is a relative remainder. We distinguish $\Delta x \neq 0$ and $\Delta x = 0$, and below expression (*) indicates an implicit definition of $r[\Delta x]$ and (**) an explicit definition:

$$\Delta F = y\,\Delta x + \epsilon[\Delta x] \qquad \text{if } \Delta x \neq 0 \quad (*)$$

$$r[\Delta x] \equiv \Delta F \,/\, \Delta x - y \qquad \text{if } \Delta x \neq 0 \quad (**)$$

$$\Delta F = 0 = c\,\Delta x + \epsilon[\Delta x] \qquad \text{if } \Delta x = 0, \text{ for any } c; \text{ select } c = y \; (*)$$

$$r[\Delta x] \equiv 0 = c - y \qquad \text{if } \Delta x = 0, \text{ for } c = y \quad (**)$$

We multiply by zero and nowhere divide by zero or infinitesimals. Now, simplify ΔF / Δx algebraically for $\Delta x \neq 0$ and determine whether setting $\Delta x = 0$ gives a defined outcome. When the latter is the case, take c as that outcome $c = \{\Delta F$ // Δx, then set $\Delta x = 0\}$. The selection of $c = y$ is based upon a sense of "continuity", not in the sense of limits but in the sense of "same formula", in that (*) and (**) have the same form irrespective of the value of Δx. We then find $c = y = f[x]$ which can be denoted as $F'[x]$ as well.

The deeper reason (or "trick") why this construction works is that (*) evades the question what the outcome of $\epsilon[\Delta x]$ // Δx would be but (**) provides a definition when the error is seen as a formula. The explicit definition (**) or implicit version (*) gives exactly what we need for both a good expression of the error and subsequently the "derivative" at $\Delta x = 0$. The deepest reason (or "magic") why this works is that we have defined $F[x]$ as the surface (or integral), both with an approximation and an error for *any* approximation that is accurate for $\Delta x = 0$, so that when the error is zero then we know that $F[x]$ gives the surface under the $c = y = f[x] = F'[x]$ that we found. If we summarize what we have done then we find the program $F'[x] = dF / dx \equiv \{\Delta F$ // Δx, then set $\Delta x = 0\}$. The definitions (*) or (**) give the rationale for extending the domain with $\Delta x = 0$, namely *form*.

For example, the derivative for $F[x] = x^2$ gives $dx^2 / dx = \{((x + \Delta x)^2 - x^2)$ // Δx, then $\Delta x := 0\} = \{2x + \Delta x$ & $\Delta x := 0\} = 2x$. This contains a seeming "division by zero" while actually there is no such division because it only summarizes the procedure (*) and (**). While the Weierstraß approach uses predicate logic to identify the limit values, this alternative approach uses the logic of formula manipulation.

Historically, the introduction of the 0 in Europe around AD 1200 gave so many problems that once those were getting solved, the solution that one cannot divide by zero was codified in stone, and pupils in the schools of Europe would meet with bad grades, severe punishment and infamy if they would sin against those sacrosanct rules. Tragically, a bit later on the historical timeline, division by zero seemed to be important for the differential quotient. Rather than reconsidering what "division" actually meant, Leibniz, Newton and Weierstraß decided to work around this, creating the concepts of infinitesimals or the limit. In this way they actually complicated the issue and created paradoxes of their own. Logical clarity and soundness can be restored by distinguishing between the (formal) act of division and the (numerical) result of division.

11.6 Non-standard analysis

This book explains that Gödel's theorems hold when we accept weak assumptions and that they evaporate when we administer some strength. It also develops the point that most philosophical arguments on the consequences of the theorems might still hold but for other reasons than the theorems, since they seem to imply such strong assumptions and thus would evaporate with the theorems. A mathematician wrote this author: "I am not surprised that non-mathematicians made nonsense of it, nor that perhaps even the author [i.e. Gödel], without his mathematician's hat on, made nonsense of it. One should always beware of mathematicians or physicists when they start talking about philosophy. (Usually it is best simply to ignore everything they say in that mode.)"

This still leaves the point that the theorems hold under those weaker assumptions. One development that uses Gödels theorems (i.e. likely under those weaker assumptions) is "non-standard analysis". This must be mentioned since mathematicians tend to find it a great invention. The same mathematician: "One of the beautiful things one can do with Gödel is use it to introduce infinitesimals." I find it impossible to say anything about this since I have not studied non-standard analysis and don't have the time to study it. It must be noted that this would not solve the problem of division by zero, as discussed above, since there still would be "infinitesimals" (that then would be non-zero). This also leads to number theory and may be considered to be no longer an issue of logic and inference itself.

11.7 Gödel's theorems

11.7.1 Gödel-Rosser

Gödel (1931:616), quoted this book p211 and 213, explains his results in terms of undecidability, which holds under the assumption of semantical correctness, i.e. the use of a model or interpretation (this book p208-209). Formally seen (i.e. uninterpreted) his theorems only show unprovability. It was Rosser 1936 who established undecidability proper, see DeLong (1971:180). The conclusions of this book are not affected, though. First of all, it is proper to assume semantical correctness or a model since it would be kind of strange to develop a formal model and not be able to use it. Secondly, this book §9.2-3 shows that the system under reasonable assumptions becomes inconsistent and proves everything, so that Rosser's finding is not so relevant.

11.7.2 Proof-consequentness

Some logicians sent me an email that proof-consequentness $\forall p$: $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$ would already be known under other names and even be valid for Peano Arithmetic, S = PA. This may or may not be true, but does not seem so relevant in itself. What holds for PA does not necessarily extend to the distinction between "self-references for sentences" and "self-reference for systems". The issue is immaterial to the basic point that adding reasonable properties to a system causes strange results, with eventually the collapse of the Gödeliar to the Liar. The name "proof-consequentness" remains useful since this expresses that S is consequent in this respect.

11.7.3 Method of proof

Page 212 and 218 use the method of proof by posing and retracting hypotheses but the proofs can also be translated by explicitly assuming properties, see also Colignatus (2007m). The properties state something about S and they don't show how it works in S. This manner has great economy since it allows for a wide variety for internal methods in systems while at the same time it allows us to discuss the consequences. An objection might be that this lacks an existence proof that there is at least one internal method to generate those outward properties. Below however gives one small example ("a robot") and this shows how that question can be dealt with more generally. While we lack a robot, a practical solution would be to ask someone to sit in a black box and act as that robot, and we would all be able to tell when that person would cheat (for the present proofs, that is).

11.7.3.1 For page 212

Property 1 is that if p is proven and the consequence would be that q is proven, then the system is smart enough to see itself that $p \Rightarrow q$. The system doesn't have to be real smart, it may only require a robot to put a p in a box "hypothesis", to note that some q falls in the box "consequence", and then pick up both statements and transfer $p \Rightarrow q$ to the "proven" box. We may forget about that robot and then have the property itself:

1. $\forall p$: $((S \vdash p) \Rightarrow (S \vdash q)) \Rightarrow (S \vdash (p \Rightarrow q))$

Property 2 is proof-consequentness, i.e. when someone says p then this person is also willing to say that he or she said p.

2. $\forall p$: $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$

One will quickly note that we can substitute $q = (S \vdash p)$ in (1) and that modus ponens on (1) and (2) generate:

(1) & (2) $\forall p$: $S \vdash (p \Rightarrow (S \vdash p))$

which is half of the "definition of truth" construction (i.e. where S would declare that proof is its concept of truth or assertion). We are getting closer to the Liar paradox.

When we take $p = g$ in (1) & (2) then we get $S \vdash (g \Rightarrow (S \vdash g))$ or $S \vdash (g \Rightarrow \neg g)$, and hence $(S \vdash \neg g)$. QED. Thus we have reproduced the result of page 210.

The following does the same in smaller steps. In the general condition $(S \vdash p) \Rightarrow (S \vdash (S \vdash p))$ we substitute $p = g$ (line 2) and $(S \vdash g) = \neg g$ (line 3). Line 4 mentions poperty 1 and then line 5 substitutes $p = g$ and $q = \neg g$. Then we apply modus ponens on line 5 and 3.

1	$(S \vdash (g \vee \neg g))$
2	$((S \vdash g) \Rightarrow (S \vdash (S \vdash g)))$
3	$((S \vdash g) \Rightarrow (S \vdash \neg g))$
4	$(((S \vdash p) \Rightarrow (S \vdash q)) \Rightarrow (S \vdash (p \Rightarrow q)))$
5	$(((S \vdash g) \Rightarrow (S \vdash \neg g)) \Rightarrow (S \vdash (g \Rightarrow \neg g)))$
6	$(S \vdash (g \Rightarrow \neg g))$
Ergo	————
7	$(S \vdash \neg g)$

Note that $S \vdash \neg g$ means that S refutes g, so that, if the system is consistent, g is not provable, and since it says so, it is true. Hence the system refutes a true statement. Given the equivalence between g and c, we also have $S \vdash \neg c$, meaning that the system proves is own inconsistency (while we would tend to assume its consistency). Note also that the issue is decidable i.e. not undecidable. This would conflict with assuming semantical correctness (see p208-209) since that would cause g to be undecidable while it is decidable here.

11.7.3.2 For page 218

Property 3 is a bit more complex. If S proves p and if for some q the combination of *p and* q causes an inconsistency, then S is smart enough to conclude that $p \Rightarrow \neg q$.

3. $\forall p, q$: $((S \vdash p) \wedge (S \vdash ((p \wedge q) \Rightarrow \neg c))) \Rightarrow (S \vdash (p \Rightarrow \neg q))$

The relevance of this approach will become clearer when it is shown that this leads to ($S \vdash c$). This discussion will also show that property 3 is quite reasonable.

When the system accepts Tertium non Datur (TND) then it cannot accept ¬TND since this would cause a contradiction, and conversely. This has a consequence for proving consistency. Consistency is equivalent to the *nonsequitur* of the *denial* of *tertium non datur* (yes, three times *not*). We can show the latter by using the key point that "∃ .. and ..." *in* the system is the same as "∃ ... and ..." *about* the system.

definition $(S \vdash \text{TND}) \Leftrightarrow \left(S \vdash \forall_p \, (p \vee \neg p)\right)$

thus $(S \vdash \neg\,\text{TND}) \Leftrightarrow \left(S \vdash \exists_p \, (p \wedge \neg p)\right) \Leftrightarrow \exists_p \, ((S \vdash p) \wedge (S \vdash \neg p)) \Leftrightarrow \neg c$

thus $(\neg\,(S \vdash \neg\,\text{TND})) \Leftrightarrow \forall_p \, ((\neg\,(S \vdash p)) \vee (\neg\,(S \vdash \neg p))) \Leftrightarrow c$

Then we get the following proof. It will be useful to substitute $A = (S \vdash \text{TND})$ and $B = (S \vdash \neg\text{TND})$:

1)	$S \vdash (((\text{S} \vdash \text{TND}) \wedge (\text{S} \vdash \neg\text{TND})) \Rightarrow \neg c)$	definition of $\neg c$ with p = TND
2)	$S \vdash ((A \wedge B) \Rightarrow \neg c)$	line 1 with shorter A and B
3)	$S \vdash \text{TND}$	axiom of S
4)	$S \vdash (S \vdash \text{TND})$	proof-consequent
5)	$S \vdash A$	line 4 with $A = (\text{S} \vdash \text{TND})$
6)	$((S \vdash p) \wedge (S \vdash ((p \wedge q) \Rightarrow \neg c))) \Rightarrow (S \vdash (p \Rightarrow \neg q))$	property 4 for some p and q
7)	$S \vdash (A \Rightarrow \neg B)$	from 2, 5, and 6 for $p = A$ and $q = B$
8)	$S \vdash ((S \vdash \text{TND}) \Rightarrow \neg(S \vdash \neg\text{TND}))$	translate back
9)	$S \vdash \neg(\text{S} \vdash \neg\text{TND})$	modus ponens on 4 and 8
10)	$S \vdash c$	translating again

Which gives an inconsistency with the earlier $S \vdash \neg c$.

A counterargument to property 3 might be that some systems could allow such a q to remain undecidable. The point however is not that we can imagine weak systems but that property 3 is not unreasonable and we would want to accept it for reasonable systems. Surrendering decision power just to turn a Liar into a Gödeliar is not reasonable when the problem can also be solved by three-valued logic.

11.7.4 Philosophy of science

The term "mathematics" derives from the Greek "what has been learnt". It is useful to evaluate what we have learnt, and therein employ the angle not only of "definitions, axioms (implicit definitions), theorems and proofs" but also the angle of the philosophy of science and the links between logic and empirical science.

The term "metamathematical" is best used with respect to a former mathematical exercise, and not in an absolute sense, since otherwise we would not be able to do mathematics now. Thus, with respect to the former exercise, it must be noted that mathematics in general has the structure $\alpha \Rightarrow \beta$ or that we derive some consequences from some assumptions. We can never deduce more than we assumed, hence essentially mathematics is the begging of the question $\alpha \Rightarrow \alpha$, only less obvious, for we seek a β that may follow from α but originally seemed to differ from it. Also, the mathematician pur sang cannot say much about the selection of α and concentrates on the deduction. A pure logical refutation of Gödel's theorems thus is rather a non-affair since we can choose $\alpha = \gamma$ sufficiently weak so that they hold and sufficiently strong $\alpha = \gamma \wedge P$ (with P for example the properties mentioned above) so that they create the inconsistency within the system or the contradition outside it when interpreted. The only relevant conclusion is that for consistency in two-valued logic one is forced to the weak assumptions. This is the mathematical position. On the other hand there is the metamathematical position of the empirical scientist. The empirical description of the world contains a mathematical substratum that causes the empirical scientist to employ his or her mathematical faculty of mind. It would be an empirical scientist (e.g. the author of ALOE when not speaking purely logically) who would select α on some empirical grounds. The choice of $\alpha = \gamma$ has a high price since it neglects useful properties P. For empirical science we rather would use $\alpha = \epsilon \wedge P$, in fact $\gamma = \neg(\epsilon \wedge P)$, where the difference between γ and ϵ for example concerns the difference between two-valued and three-valued logic. From an empirical point of view "much" of the mathematical study of two-valued logic and the consequences of Gödel's theorems is nonsensical. Consistent but nonsensical. This part of modern mathematics adheres to some dogma of two-valued logic and thus chooses to loop out of science. These metamathematical considerations thus both emphasize the difference between math and empirical science and highlight that modern mathematics is insulated against criticism since it uses only consistency as its criterion, since, indeed, the choice of weak $\alpha = \gamma = \neg(\epsilon \wedge P)$ indeed is consistent for some $\neg(\epsilon \wedge P) \Rightarrow \beta$.

In this respect, mathematicians are the lawyers of science, and their work may at times be as productive and destructive for science as the work of lawyers can be for society. But where scientists would maintain some distance with respect to the question what

goals a society would or should attain, see our discussion on deontic logic, the situation is different for science, since here the objective has been given by definition, i.e. science aspires at discovering the truth, and it is here that the destructive element of these lawyer-mathematicians stands out.

How to break the stranglehold of the age-old Liar paradox on modern mathematics ? This book follows the approach of showing some properties P that are reasonable by themselves and that, when added to the Gödelian system cause it to collapse. Of course, the mathematician pur sang will say that it is easy to include some axioms, such as $S \vdash (\neg c \Rightarrow c)$ or $S \vdash (p \Rightarrow (S \vdash p))$. This mathematician pur sang will only consider the $\alpha \Rightarrow \beta$ structure and will have no strong opinion on the assumptions. If the inclusion causes an inconsistency he or she may well conclude that this is a begging of the question so that some of those additions must be "perverse". $S \vdash (\neg c \Rightarrow c)$ would be rejected since it is directly equivalent to an internal proof of consistency $S \vdash c$, and $S \vdash (p \Rightarrow (S \vdash p))$ would be rejected since it directly introduces part of the Liar paradox in S. Such reactions can indeed be observed. However, for the empirical scientist it is strange to reject useful properties just to maintain a theorem, and it is strange that S would not be allowed to know itself what its concept of proof means. One would rather solve the Liar paradox than forbid its creation. One strategy to break the deadlock is to stay away from those obvious properties and introduce the weaker P mentioned above. These properties run up against the same kind of criticism from the mathematician pur sang who is insulated against any consideration on the choice of assumptions. Nevertheless, rejection becomes harder, and by using the weaker P it can become clearer that reasonable properties are rejected and that this carries the high price of turning this part of mathematics irrelevant. Eventually, something would have to give. One is reminded of the Chinese saying "The situation was unbearable. It could no longer continue. It lasted 300 years".

The most likely outcome is that (after those 300 years) the mathematician pur sang will adopt three-valued logic and proceed as before, since little will have changed in attitude, only some other $\alpha \Rightarrow \beta$. Nevertheless, for the world, and students sensitive to logical paradoxes, it would be a beneficial change. And for the mathematical community it might become a topic of consideration whether it is so wise to aspire to become a mathematician pur sang. Mathematics is a faculty of the mind and not a way of life nor a mold to shape your personality. Aristotle, Archimede, Newton, Leibniz did some decent math while not getting lost in it. Modern math did get lost. Young minds aspiring at mathematics would better be wise in selecting their role models. Another important notion would be that a Department of Mathematics rather should not be linked to mainly the Physics Department but be central to all other departments too such as Economics, Psychology etcetera including a higher regard for engineering. Colignatus (2007i) gives another example where professors of math and physics go

astray - i.e. where they adopt the Penrose square root rule for voting weights, and do this even in an open letter to the governments in the European Union, while their assumptions cannot be maintained since they neglect an empirical and statistical argument, and subsequently confuse facts with morals. In one part of his or her mind, the mathematician insulates him or herself from the world by adopting the $\alpha \Rightarrow \beta$ attitude, but in the other part of the mind it appears hard to do this consequently. Rather than maintain the fiction that the mathematical faculty of mind can also become the person himself or herself, it would be better to immerse mathematicians in the real world so that they are better aware of the pitfalls and uncertainties lurking there. They would no longer be "mathematicians" but mere scientists using mathematics. One practical rule could be that an academic writes one paper on a real problem once in two years - where "reality" is defined in terms of experimental data. Another rule would be that mathematicians who cannot deal with the world should not be the core of their Department but rather the fringe. Another issue is that researchers of the didactics of mathematics should not hide in their research journals but be actually involved in the teaching of mathematics. In the present, the patients have taken over the asylum and abuse the great practical history of mathematics to lure the world along in their lunacies. We would rather see that more practically minded scientists protect the world from all that in the future. More on this in *Elegance with Substance* (2009).

11.8 Scientific attitude revisited

The episode from January 2007 to March 2011 confirms what has been said in the "plea for a scientific attitude", §9.3.14 above. Years passed, now with the available power of the internet and *Mathematica*, in which logicians managed to neglect the analysis in this book. Coming September will show the start of another academic year where new students will be indoctrinated with nonsense again. Mathematics would be a most democratic and liberating activity since there is no authority that enforces a theorem but only the rational recognition by proof. Unfortunately, this still requires a person to maintain an open mind to new insights and new proofs or refutations. Generally, it is the scientific attitude that fosters such an open mind. Logicians and mathematicians fail on that attitude. This has consequences for what they tell society, not only on logic and common sense but also on voting theory and notions of democracy, and possibly other areas (though not studied by this author). It is difficult to say how sick the patient is. Yet, a world with an overpopulation of 6.9 billion souls growing to a super-overpopulation of 9 billion souls would not have the luxury to endure this. One cannot but conclude that society needs a fundamental change in the way how it deals with logic and mathematics.

Conclusion

This book has discussed the subject of logic from the bottom up. The basic elements in logic are without contention: two-valuedness, propositional logic, predicate logic, set theory, inference, applications to morality and the like, axiomatization and the criteria for deductive systems such as consistency. Yet the age-old paradox of the Liar is a bone of contention that causes different solution approaches such as the theory of types, the disabandonment of truth in favour of proof theory and undecidability, or three-valued logic. The first two approaches create their own problems. They not only sacrifice key forms of selfreference but also tax the mind with "levels" in language or statements like the Gödeliar that are cousins of the Liar so that one feels that actually nothing really has been solved. Only three-valued logic allows for a consistent development that is agreeable to how people understand language and reasoning. Yet three-valued logic is out of favour with the logic community because of supposed complexities. There is an irrational element in that since three-valued logic certainly is simpler than the complexities of the other two approaches.

Looking at the existing literature on logic, we find that it is grossly inadequate and highly misleading on these key issues. In particular, Gödel's theorems appear to be mathematical constructs that have a different meaning than suggested by its author and they do not support his conclusions - which however are generally adopted in the literature. By discussing these errors and showing where the claims were unwarranted, we have in fact restated what we indicated in the beginning: namely showing how logic and inference can be clear and sound.

Our discussion has also highlighted some areas where more research in logic would be fruitful. Formal mathematicians would want to make the argument sufficiently formal so that they themselves can accept it too. Since the subject matter is beriddled with paradoxes and misunderstandings, more clarification is a good suggestion indeed. Secondly, we have indicated that exceptions might have a common structure ("unless"), but there may be other ways and more fruitful notations and procedures. Another point is the semantical interpretation of three-valued logic. We have indicated a process of dynamic and hypothetical assertion but we might want to see more on that. A final point is the implementation on the computer. The programs provided here and possibly those on other machines are a small step towards better decision support but in our bewildering world, with the huge challenge of the world population growth and the survival of civilization as we know it, we need more and better. There are, in other words, still practical problems as well.

Literature

EWP references are to the Economics Working Papers Archive at the Washington University at St. Louis: http://econwpa.wustl.edu. See also http://thomascool.eu.

Thomas Colignatus is the preferred name of Thomas Cool in science.

PM. The other references mentioned in Colignatus (1980) are not included here for practical reasons.

Aronson, E. (1972, 1992a), “The social animal”, Freeman, sixth edition

Aronson, E. (ed) (1973, 1992b), Readings about The Social Animal”, Freeman, sixth edition

Ayer, A.J. (1936, 1978), “Language, truth and logic”, Pelican

Barney, G. O. (1981), “The global 2000 Report to the President”, Blue Angel Inc.

Barwise, J. (ed) (1977), “Handbook of mathematical logic”, Studies in Logic, North-Holland

Beth, E.W. (1959), “The foundations of mathematics; a study in the philosophy of mathematics”, North-Holland

Bochenski, I.M. (1956, 1970), “A history of formal logic”, 2nd edition, Chelsea

Brouwer, L.E.J. (1967), “On the significance of the principle of the excluded middle in mathematics, especially in function theory”, see Van Heijnoort (1967)

Brouwer, L.E.J. (1975), “Collected works 1; philosophy and foundations of mathematics”, North-Holland

Church, A. (1956), “Introduction to mathematical logic”, Princeton

Colignatus (1980), “Maniakken en andere logici. Een kritische inleidende behandeling van de logika en de methodologie der wetenschappen”, draft, unpublished

Colignatus (1981), “In memoriam Philetas of Cos et alii”, unpublished

Colignatus (1992b), “Definition and Reality in the general theory of political economy; Some background papers 1989-1992”, ISBN 905518-207-9, Magnana Mu Publishing and Research, Rotterdam

Cool (1999, 2001), “The Economics Pack, Applications for *Mathematica*”, Scheveningen, ISBN 90-804774-1-9, JEL-99-0820 (no longer in print)

Colignatus (2000), “Definition and Reality in the General Theory of Political Economy”, Samuel van Houten Genootschap, ISBN 90-802263-2-7, JEL 2000-1325

Colignatus (2001), “Voting Theory for Democracy”, Thomas Cool Consultancy & Econometrics, ISBN 90-804774-3-5

Colignatus (2005), “Definition and Reality in the General Theory of Political Economy”, 2nd edition, Dutch University Press

Colignatus (2006), “The Political Economy of the Netherlands Antilles and the Future of the Caribbean”, Samuel van Houten Genootschap, ISBN ISBN-10: 90-802263-3-5, ISBN-13: 978-90-802263-3-3

Colignatus (2007a), “A Logic of Exceptions”, 1st edition, Thomas Cool Consultancy & Econometrics

Colignatus (2007b), “Voting Theory for Democracy”, 2nd edition, Thomas Cool Consultancy & Econometrics

Colignatus (2007e), “Elementary statistics and causality”, work in progress, http://thomascool.eu

Colignatus (2007i), "Why one would accept Voting Theory for Democracy and reject the Penrose Square Root Weights", http://mpra.ub.uni-muenchen.de/3885/

Colignatus (2007j), "Refutation of "the" proof for Cantor's Theorem and restoration of the "set of all sets"", http://thomascool.eu/Papers/ALOE/2007-07-29-OnCantorsTheorem.pdf

Colignatus (2007k), "Improving the logical base of calculus on the issue of "division by zero"", http://thomascool.eu/Papers/ALOE/2007-07-31-ImprovingTheLogicOfCalculus.pdf

Colignatus (2007m), "A difficulty in proof theory", http://thomascool.eu/Papers/ALOE/2007-08-02-ADifficultyInProofTheory.pdf

Colignatus (2008), "Review of Howard DeLong (1991), "A refutation of Arrow's theorem", with a reaction, also on its relevance in 2008 for the European Union"", July 22 2008, MPRA 9661, http://mpra.ub.uni-muenchen.de/9661/

Colignatus (2009), "Elegance with Substance", Dutch University Press

Colignatus (2010), "Resolution of Russell's Paradox (reflecting on Logicomix)", http://thomascool.eu/Papers/ALOE/2010-02-14-Russell-Logicomix.pdf

Colignatus (2011a), "Conquest of the Plane", 1st edition, Thomas Cool Consultancy & Econometrics

Colignatus (2011b), "Voting Theory for Democracy", 3rd edition, Thomas Cool Consultancy & Econometrics

Curry, H.B. (1963), "Foundations of mathematical logic", p.m.

Damasio, A. (2003), "Looking for Spinoza", Harcourt

Davis, M. (1965), "The undecidable", Rave Press, Hewlett N.Y.

DeLong, H. (1971), "A profile of mathematical logic", Addison Wesley

DeLong, H. (1991), "A Refutation of Arrow's Theorem", University Press of America

DeLong, H. (1997), "Jeffersonian teledemocracy", http://law-roundtable.uchicago.edu/v4.html

Dopp, J. (1969), "Formale Logik", Benziger Verlag

Doxiadis, A., C. H. Papadimitriou, A. Papadatos, A. Di Donna (2009), "Logicomix: An Epic Search for Truth", Bloomsbury USA

Dijksterhuis, E.J. (1990), "Clio's stiefkind. Een keuze uit zijn werk door K. van Berkel", Bert Bakker

Feigl & Sellars (1949), "Readings in philosophical analysis", Appleton-Century-Crofts

Finsler, P. (1967), "Formal proofs and undecidability", see Van Heijenoort (1967)

Frege, G. (1949), "On sense and nominatum", reproduced in Feigl & Sellars (1949)

Freudenthal, H. (1937), "Zur intuitionistischen Deuting logischer Formeln", see Heyting (1980)

Gill, R.D. (2008), "Book reviews. Thomas Colignatus. A Logic of Exceptions: Using the Economics Pack Applications of *Mathematica* for Elementary Logic", NAW 5/9 nr. 3 sept., http://www.math.leidenuniv.nl/~naw/serie5/deel09/sep2008/reviewssep08.pdf

Gödel, K. (1931), "Some metamathematical results on completeness and consistency. On formally undecidable propositions of *Principia Mathematica* and related systems I" and "On completeness and consistency", see Van Heijenoort (1967)

Hao Wang (1974), "From mathematics to philosophy", Routledge & Kegan Paul

Heijenoort, (ed) (1967), “From Frege to Gödel”, Harvard

Heyting, A. (1980), “Collected papers”, Mathematical Institute, Univ. of Amsterdam

Hilbert, D. (1927), “The foundations of mathematics”, see Van Heijenoort (1967)

Hodges, W. (1978), “Logic”, Penguin

Hughes, P.& G. Brecht (1979), “Vicious circles and infinity; an anthology of paradoxes”, Penguin

Jeffrey, R.C. (1967), “Formal logic: Its scope and limits”, McGraw-Hill

Kaufmann, W. (1968), “Philosophic Classics; Volume 1: Thales to Ockham”, Prentice-Hall

Kneebone, T. (1963), “Mathematical logic and the foundations of mathematics; an introductory survey”, Van Nostrand

Kleene, S.C. (1952), “Introduction to metamathematics”, Van Nostrand

Kleene, S.C. (1965), “The foundations of intuitionist mathematics, especially in relation to recursive functions”, North-Holland

Kripke, S. (1975), “Outline of a theory of truth”, The Journal of Philosophy, p.m.

Martin, R.L. (ed) (1970), “The paradox of the Liar”, New Haven, Yale

Malitz, J. (1979), “Introduction to mathematical logic”, Springer Verlag

Nagel, E. and J.R. Newman (1975), “De stellingen van Gödel”, Aula, Utrecht

Popper, K.R. (1977), “The open society and its enemies; Vol 2. Hegel & Marx”, Routledge

Quine, W.V.O. (1981), “Elementary logic”, Harvard

Quine, W.V.O. (1976), “The ways of paradox and other essays”, Harvard

Quine, W.V.O. (1990), “Pursuit of truth”, Harvard

Rescher, N. (1964), “Introduction to logic”, St. Martin’s press

Reichenbach, H. (1966), “Elements of symbolic logic”, Free Press

Rose, S. (1978), “The conscious brain”, Pelican

Russell, B. and A.N. Whitehead (1964), “Principia Mathematica (to *56)”, Cambridge

Russell, B. (1967), “Mathematical logic as based on the theory of types”, see Van Heijenoort (1967)

Schumpeter, A. (1967) “Kapitalisme, socialisme en democracy”, W. de Haan, Hilversum

Smorynski, C. (1977), “The incompleteness theorems”, in Barwise (1977)

Struik, D.J. (1977), “Geschiedenis van de wiskunde” SUA (“A concise history of mathematics”, Dover 1948)

Styazhkin, N.I. (1969), “History of mathematical logic from Leibniz to Peano”, M.I.T.

Tarski, A. (1949), “The semantic conception of truth”, reproduced in Feigl & Sellars (1949)

Watzlawick, P., J. Helmich Beavin, D.D. Jackson (1967), “Pragmatics of human communication”, p.m.

Visser, A. (1980), “The liar paradox”, Univ. of Utrecht, unpublished

Wallace, D.F. (2003), “Everything and more. A compact history of ∞”, Norton

Williams, B. (2002), “Truth and truthfulness”, Princeton

Wilson, J.Q. (1993), “The moral sense”, The Free Press

Wittgenstein, L. (1921, 1976), “Tractatus Logico-Philosophicus”, Polak & Van Gennep

Wolfram, S. (1992), “Mathematica”, Addison-Wesley

Wolfram, S. (1996), “Mathematica 3.0”, Cambridge

Zermelo, E. (1908), “Investigations in the foundations of set theory I”, see Van Heijenoort (1967)