The Liar Paradox:
A Consistent and Semantically Closed Solution

By

Ryan Edward Young

A thesis submitted for the degree of Doctor of Philosophy
of The Australian National University

July 2011

This thesis develops a new approach to the formal definition of a truth predicate that allows a consistent, semantically closed definition within classical logic. The approach is built on an analysis of structural properties of languages that make Liar Sentences and the paradoxical argument possible. By focusing on these conditions, standard formal definitions of semantics are shown to impose systematic limitations on the definition of formal truth predicates.

The alternative approach to the formal definition of truth is developed by analysing our intuitive procedure for evaluating the truth value of sentences like "P is true". It is observed that the standard procedure breaks down in the case of the Liar Paradox as a side effect of the patterns of naming or reference necessary to the definition of Truth as a predicate. This means there are two ways in which a sentence like "P is true" can be not true, which requires that the T-Schema be modified for such sentences.

By modifying the T-Schema, and taking seriously the effects of the patterns of naming/ reference on truth values, the new approach to the definition of truth is developed. Formal truth definitions within classical logic are constructed that provide an explicit and adequate truth definition for their own language, every sentence within the languages has a truth value, and there is no Strengthened Liar Paradox. This approach to solving the Liar Paradox can be easily applied to a very wide range of languages, including natural languages.

The work contained in this Thesis is my own and, unless otherwise attributed, is not the result of any joint effort or research.

Ryan Young

ACKNOWLEDGEMENT. Firstly, I would like to thank Peter Roeper for his time, effort, patience and attention to detail. His willingness to pause and take time to consider and think through all of my ideas, even the worst ones, has been greatly appreciated. While our discussions tended to go round in circles, he succeeded in bringing order and rigour to my work which I would have struggled to achieve otherwise.

I am very grateful to Ingolf Max, and the other academics at the Institut für Logik und Wissenschaftstheorie at the University of Leipzig, for their time and support during my stay there. I learnt a lot and my ideas matured significantly in the stimulating environment they have created.

I am also very grateful to Brian Garrett, Daniel Stoljar, Margaret Brown and many others for their ability at various times to put up with me and make sure that I successfully got to the end of this thesis.

Finally, I have to thank Rebecca, Gwyllym and Sebastian for being there, not complaining and distracting me when I needed it. Their inspiration and ability to keep me grounded has contributed immeasurably to the quality of the work.

Contents

Introduction

The Liar Paradox is one of the oldest and has been one of the most intractable paradoxes in the Western philosophical tradition. It dates back at least to the time of Aristotle, as its use is attributed to Eubilides, one of Aristotle's contemporaries and adversaries.[1] However, despite constant philosophical and logical attention since this time, and in stark contrast to other ancient paradoxes such as Zeno's Paradoxes of Motion, no consensus has been reached over how to respond to the Liar Paradox. Moreover, this lack of consensus has become more pronounced in recent times, despite the fact that more specific attention has been paid to the paradox since the middle of the twentieth century than in any previous period. The lack of consensus is not, however, due to any failure by philosophers and logicians to devise solutions since "the problem we face is far from a lack of solutions; rather, we have an overabundance of conflicting ones."[2]

These problems are on the surface surprising, since one of the defining features of the Liar Paradox is the simplicity with which it can be presented. A typical presentation of it only requires a couple of sentences, and it does not depend on any specialised or technical concepts. Moreover, an ordinary presentation is sufficient to allow one to grasp the basic mechanism by which the paradox occurs and have an idea of what is required in order to solve the paradox.

To see this, we will consider the following sentence:

1. This sentence is not true.

This sentence (1) is simple and easy to understand. However, if we make the obvious assumption that the 'This sentence' in the sentence refers to the sentence itself, we run into the following problem:

> If sentence (1) is true, then it follows that what it says must be true. What it says is that it itself is not true. Therefore, if (1) is true, it follows that (1) is not true. However, if we assume the other alternative, that sentence (1) is not true, then what sentence (1) says is in fact true, from which it follows that (1) is true. Therefore, if (1) is not true, then (1) must be true.

[1] Roy Sorenson. *A Brief History of Paradox*. Oxford: Oxford University Press, 2003, Ch 7, pp83-99

[2] Michael Glanzberg. "The Liar in Context". In: *Philosophical Studies* 103 (2001), pp. 217–251, p. 217

In the space of a few lines, we have established that if the sentence (1) is either true or not true, then it must also be the other. Thus it must be either both true and not true; or neither true nor not true. Since neither of these alternatives seems plausible, we have a paradox.

The consequences of this paradox are, if we take the conclusion established by the argument seriously, profound. For if we accept either of these conclusions, we have accepted the truth of a contradiction. In the first case it is of the form $P \wedge \neg P$ and in the second case it is of the form $\neg P \wedge \neg\neg P$. This is unacceptable on a standard understanding of reasoning. Moreover, on traditional accounts of reasoning, if there is a true contradiction, everything is provable. This means, that if the above argument is valid in English (or any fragment of English or any other language), then every sentence within English is provably true. It would therefore be impossible to use English to reason or discuss reasoning since every sentence of English can be shown to be true. If this conclusion is correct, it not only renders pointless vast amounts of writing and discussions, but on a more personal note, means that there is no point completing this thesis, since everything written in it can be proven to be both trivially true and trivially not true.

Stated in this form, the conclusion of the reasoning in the Liar Paradox cannot be correct. Something must have gone wrong in the argument, and our task is to identify the problem. A major project of this thesis is to provide a stable and coherent diagnosis of the problem, however as the vast literature on the subject demonstrates, this is not an easy task.

The Liar Paradox is not simple

As is clear from the exposition above, the Liar Paradox is simple to present, and the mechanism by which the Paradox arises seems easy to understand. We have a situation where we have a grammatically correct sentence that claims of itself that it is not true. This, however, contradicts the standard truth conditions for declarative sentences, that a declarative sentence is true if what it claims is true. This internal contradiction, by means of our intuitive assumptions about truth, produces the type of argument which is typical of the Liar Paradox: on the assumption that the relevant sentence is true, we can directly derive its untruth; and on the assumption of its untruth, we can derive its truth. This simplicity in presenting the Liar Paradox and understanding its basic mechanism, however, does not imply that the Liar is a simple paradox. Instead it tends to hide the complexity of the phenomena that surround it.

A quick survey of the range of literature on the Paradox neatly illustrates the complexity of the relevant phenomena. For not only do few authors agree on the correct solution to the Paradox, but not many more authors agree on how exactly to identify the problem posed by the Paradox. For example, some locate the problem in traditional assumptions about logic,[3] others locate the

[3] For example, see Graham Priest. "Truth and Contradiction". In: *The Philosophical Quarterly* 50.200 (2000), pp. 305–319; Graham Priest. "The Logic of Paradox". In: *The Journal of Philosophical Logic* 8 (1979), pp. 219–241; Hartry Field. "Semantic Paradoxes and Vagueness Paradoxes". In: *Liars and Heaps: New Essays on Paradox*. Ed. by

problem in our understanding of truth bearers,[4] others locate the problem in the role of context[5] or the limited applicability of ordinary assertions[6], while yet others attribute the difficulty to a faulty understanding of the pragmatics of language[7] or even take the Liar Paradox at least in some sense as evidence of the ultimate indefinability of truth.[8] There are strong intuitions behind each alternative, and the range of different alternative diagnoses illustrates the complexity of the issues that surround the Liar Paradox.

This complexity can be most clearly seen in the range of different examples of the Liar Paradox that have been uncovered by modern research. Historically, the examples of the Liar Paradox that were advanced usually involved somewhat artificial, contrived sentences. These were examples such as "This sentence is false"; or "I am now lying". Modern research, however, has shown that the Liar Paradox arises far more commonly than these rather artificial types of examples. The simplicity of the mechanism that generates the Liar Paradox ensures that it often arises where we would least like it to arise. There are two issues that have arisen in modern research which are particularly relevant.

The first was most forcefully demonstrated by Kripke in his seminal paper "Outline of a Theory of Truth",[9] although the type of example that Kripke uses pre-dates Kripke's work in the literature.[10] Kripke considered the ordinary assertion "Most of Nixon's assertions about Watergate are false."[11] While this is normally an unproblematic assertion, Kripke described a situation in which it becomes paradoxical, namely when Nixon's assertions are evenly balanced between truth and falsity except for the one assertion that the person who uttered the above statement is telling the truth. As can be easily checked, these two assertions create a two sentence version of the Liar Paradox. The lesson that Kripke drew from this was that "many, probably most, of our ordinary assertions about truth and falsity are liable, if the empirical facts are extremely unfavorable, to exhibit paradoxical features."[12] That means that, unlike the impression generated by the types of examples that were traditionally used, there is nothing inherently problematic in the sentences themselves that are used to generate the Liar Paradox.

The second issue in relation to the ubiquitousness of the Liar Paradox that has arisen in modern research can be summarised under the label of the Revenge Problem. Modern researchers routinely

J. C. Beall. Oxford: Clarendon Press, 2003, pp. 262–310

[4] For example, see Keith Simmons. "Reference and Paradox". In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 230–252; Alan Weir. "Token Relativism and the Liar". In: *Analysis* 60.2 (2000), pp. 156–170

[5] See Glanzberg, "The Liar in Context", see n. 2; Michael Glanzberg. "A Contextual-Hierarchical Approach to Truth and the Liar Paradox". In: *Journal of Philosophical Logic* 33 (2004), pp. 27–88

[6] See Jon Barwise and John Etchemendy. *The Liar: An Essay on Truth and Circularity.* New York: Oxford University Press, 1987

[7] See A. P. Martinich. "A Pragmatic Solution to the Liar Paradox". In: *Philosophical Studies* 43 (1983), pp. 63–67

[8] For example, see Charles S. Chihara. "Priest, the Liar, and Gödel". In: *Journal of Philosophical Logic* 13 (1984), pp. 117–124

[9] Saul Kripke. "Outline of a Theory of Truth". In: *Journal of Philosophy* 72.6 (1975), pp. 690–716

[10] For example, see L. Jonathon Cohen. "Can the Logic of Indirect Discourse be Formalised?" In: *The Journal of Symbolic Logic* 22.3 (1957), pp. 225–232

[11] Kripke, see n. 9, p. 691

[12] Kripke, see n. 9, p. 692

use the Liar Paradox itself to critique proposed solutions to the Paradox. The standard strategy works as follows: any proposed solution must draw a line between truth and untruth somewhere; once this line has been identified, one can usually construct a version of the Liar Paradox which operates on this line (whether or not it is expressed in terms of truth). Thus one can almost always provide a counterexample to a proposed solution to the Liar Paradox by reformulating the Paradox in a way that uses the distinctions that the solution proposes. In this sense, the Liar Paradox takes revenge on the proposed solution, and we have the Revenge Problem. The ease with which this can normally be done demonstrates the power of the Liar Paradox and has lead to the suggestion that ultimately the Liar Paradox is not solvable.

Paradoxical vs Non-Paradoxical

While it will be shown that such a pessimistic conclusion is not warranted, it captures the scope of the challenges presented by the Liar Paradox. Any meaningful discussion of, or solution to, the Liar Paradox must draw a clear distinction between the paradoxical cases and the non-paradoxical cases, and any proposed solution must satisfactorily deal with all paradoxical cases without incorrectly affecting any non-paradoxical cases. Both Kripke's argument and the Revenge Problem, however, demonstrate that this is not an easy distinction is to make accurately.

Kripke's argument undermines a lot of work and discussion of the Liar Paradox, since these are often characterised by discussions of Liar Sentences. While there are sentences which unquestionably trigger the Paradox, Kripke showed that "it would be fruitless to look for an *intrinsic* criterion that will enable us to sieve out ... those sentences which lead to paradox."[13] That is, there is nothing in Liar Sentences as *sentences* which can distinguish them from non-paradoxical cases. The clearest example of this are cases like Kripke's where the same sentence can be either paradoxical or non-paradoxical depending on the context. This means that any attempt to draw the distinction between the paradoxical and the non-paradoxical purely at the level of sentences cannot work. This raises the significant question of how we are to draw this distinction.

The Revenge Problem, on the other hand, raises the question of whether it is in fact possible to draw a solid distinction between the paradoxical and the non-paradoxical. As mentioned, the problem turns on the fact that once a proposed solution has distinguished between truth and untruth in any sense, the paradox can be usually formulated using this distinction. However, any proposed solution must draw the line somewhere between truth and untruth, and in doing this must develop some distinction between the paradoxical and the non-paradoxical. When the Revenge Problem bites, the line between truth and untruth is shown to be inadequate, and therefore the distinction that was made between paradoxical and non-paradoxical cannot be correct. If the Revenge Problem cannot be avoided, then it will also be impossible to make a solid distinction

[13] Kripke, see n. 9, p. 692

between the paradoxical and the non-paradoxical.

This process can be neatly illustrated by the simple solution to the Liar Paradox which proposes the introduction of a category of sentences which are neither true nor false as a solution to the Liar Paradox. This is commonly done on consideration of paradoxical sentences of the form "This sentence is false". However, as is well known, any simple version of this solution fails to account for any example which replaces "false" with "not true". That is, the solution proposes the category of 'neither true nor false', which means that the line between truth and untruth shifts to being between 'true' and 'neither true nor false'. As soon as we reformulate the Liar Paradox in terms of this distinction ("This sentence is *not true*"), the Liar Paradox arises again. The analysis of the paradox which supports this proposed solution failed to correctly draw the line between the paradoxical and the non-paradoxical, and therefore the solution failed. While this is a simple example, the frequency with which the Revenge Problem causes exactly the same problems suggests that it may in fact be impossible to definitively draw a line between the paradoxical and the non-paradoxical.

It will be demonstrated in this thesis that this suggestion can remain as a suggestion, as it is in fact possible to make a sharp distinction between the paradoxical and the non-paradoxical. However, a sharp distinction is only possible if we take Kripke's point outlined above very seriously. If we want to draw a distinction between the paradoxical and the non-paradoxical, we need to be clear about what *sort of thing* we are considering as paradoxical or not. For example, it is often assumed, either explicitly or implicitly, that in the case of the Liar Paradox it is the sentences that are paradoxical. If this is the case, then the distinction between the paradoxical and the non-paradoxical has to be a distinction between the paradoxical sentences and the non-paradoxical sentences. Kripke showed, however, that this is not possible as many perfectly satisfactory sentences can be paradoxical in unusual situations. The first challenge in drawing the required distinction is to get clear on what sort of thing we need to consider in drawing the distinction.

What *sort of thing* is paradoxical?

The problem identified by Kripke is that "it would be fruitless to look for an *intrinsic* criterion that will enable us to sieve out ... those sentences which lead to paradox."[14] That is, there is no guarantee that the sentences which are (potentially) paradoxical will share any common set of properties on which we can draw a coherent distinction between the paradoxical and the non-paradoxical and therefore on which we could base a satisfactory solution. Kripke's conclusion is based on the observation that any sentence which includes a truth ascription could be paradoxical in appropriate circumstances. There is therefore nothing different about paradoxical sentence when compared to non-paradoxical sentences, since many sentences can be both paradoxical and

[14] Kripke, see n. 9, p. 692

not paradoxical in different contexts.

Given the role that context and external situations play in Kripkean examples, some role for context has to be included in the sort of thing we want to investigate. The obvious solution to this problem would therefore be to investigate situations or states of affairs, in conjunction with sentences, as the sort of thing on which we could draw a distinction between the paradoxical and the non-paradoxical.

While it is plausible that there may be a clear distinction between paradoxical and non-paradoxical situations, any strategy of this sort runs into problems when we consider the infinitely large range of possible combinations of sentences and situations. If we consider only a sentence such as "That is not true" within English, there are infinitely many possible referents of this sentence (including itself) embedded in infinitely many possible situations. An infinite subset of these referents will in turn refer to some other sentence, or set of sentences, and in each of these there will be again infinitely possible referents of the sentences in this infinite subset. The fact that some of these referents to the sentence "That is not true" will, in the appropriate context, be paradoxical, is purely due to the combinatorics of the situation. However, identifying exactly which of these possible combinations of sentence and context are paradoxical and which are not is, at the very least, not an easy task. When we in turn consider the huge variety of sentences that we can start with, it should be clear that identifying a rule-governed method of producing all possible types of situation/sentence combinations which are paradoxical is a mammoth task, which may or may not be solvable. At the very least it is a major research project in its own right. Without such a rule-governed method, it is impossible to be sure that we are drawing the line between the paradoxical and the non-paradoxical correctly.

Moreover, even if we can draw a line between the paradoxical and the non-paradoxical on this basis, it may not be very useful in terms of grounding a solution to the Liar Paradox. The distinction between the paradoxical and the non-paradoxical will, using sentence/context sets, be largely drawn through an understanding of the combinatorics of the situation. However, we cannot change the combinatorics that lead to the Liar Paradox without a fundamental reworking of the way we use languages such as English. Requiring this as part of a solution to the Liar Paradox is neither likely to be plausible nor to gain widespread acceptance.

Given that there are problems with investigating sentences and sentence/contexts combinations, we could consider the common philosophical reflex to look at different possible truth-bearers. To take a couple of examples, we might consider looking at propositions, or sentence-tokens. Importantly, each of these requires some sort of context sensitivity in their definition. Context is often necessary to determine the proposition expressed by a sentence; and a sentence-token, as a token, necessarily belongs to a particular context. However, while each of these have been advocated as solutions to the Liar Paradox, each of these is faced with essentially similar problems.

It has been argued that taking propositions as primary truth bearers can solve the problem since Liar Sentences either do not express a proposition,[15] or express a false proposition.[16] These solutions naturally face the standard problems that all accounts of propositions face. However, independently of these problems, turning to propositions as part of an attempt to better understand the Liar Paradox produces the same problems as sentences. For whether we consider the sentence itself, or the proposition expressed by "Most of Nixon's assertions about Watergate are false"[17], nothing intrinsic can be found which separates paradoxical from non-paradoxical use of that sentence. The proposition expressed will be the same whether it is embedded within a paradoxical context or not. Considering propositions does not help us draw any line between paradoxical cases and non-paradoxical cases.

The alternative move to take sentence tokens as the basic truth bearer does not help either.[18] It would be entirely consistent on this account that one assertion of "Most of Nixon's assertions about Watergate are false" is true, and while another is false because it is paradoxical. However, this is not enough since there is nothing in the sentence tokens themselves which differentiate paradoxical tokens from non-paradoxical tokens. We still need something about the context, but that opens up the problems outlined above in the case of sentences. Thus considering sentence tokens does not in itself add any more information to an attempt to differentiate between the paradoxical and the non-paradoxical.

Thus focusing our investigation on different possible truth bears does not help us distinguish the paradoxical from the non-paradoxical since there is nothing in the truth bearers themselves which reflect the difference between the paradoxical Kripkean cases from non-paradoxical cases. Adding the relevant context directly into our considerations, however, adds enormous complexity, which brings in questions about the decidability of the problem. Moreover, a distinction on these grounds would be based on a consideration of the combinatorics of language reference, which is not something that can be plausibly altered for natural languages.

Languages

Looking at the alternatives to taking sentences as the focus of the investigation into the Liar Paradox examined here, any alternative truth-bearer to sentences runs into the same problems as Kripke showed existed for sentences, since it is not possible to distinguish between paradoxical and non-paradoxical cases without referring to external context. However, including external context in the basic unit of investigation poses problems of a different sort, the immense complexity that arises on this approach undermines attempts at progress. It is very difficult to be sure that the

[15] See Laurence Goldstein. "A Unified Solution to Some Paradoxes". In: *Proceedings of the Aristotelian Society* 100 (2000), pp. 53–74

[16] See E. Mills. "A Simple Solution to the Liar". In: *Philosophical Studies* 89 (1998), pp. 197–212

[17] Kripke, see n. 9, p. 691

[18] For one example, see Alan Weir. "Token Relativism and the Liar". In: *Analysis* 60.2 (2000), pp. 156–170

line between the paradoxical and the non-paradoxical has been drawn correctly.

Each of these alternatives focus our investigations on actual paradoxical cases. However, the structure of the paradoxical argument in the Liar Paradox suggests that this is not necessary. The Liar Paradox argument always begins with a hypothetical assumption: “Suppose this particular sentence is true, ...” of “If that particular sentence is not true, ...”. This means that the Liar Paradox only requires the potential existence of a paradoxical case to arise. So long as, for example, we *can* construct an appropriate example of the Liar Paradox in English, then the consequences posed by the Paradox above need to be faced. We cannot, for example, deal with the Liar Paradox by banning the use of paradoxical sentences or statements. The Liar Paradox bites if these sentences are merely constructable.

Given that the Liar Paradox turns on the possibility of certain types of sentences or situations, this suggests that the focus our investigations into the Liar Paradox should be on the conditions which make these sentences and the Liar Paradox possible, rather than actual cases of the Paradox. This would mean that our focus should be on the linguistic structures that make paradoxical sentences possible, and the reasoning which generates the Paradox. That is, if we are to take this suggestion seriously, we should be investigating languages, rather than sentences or propositions or situations.

This idea that the focus of investigations into the Liar Paradox should be focused on languages is supported by the brief account of the Paradox given above. As stated above, the most serious consequence of taking the Liar Paradox seriously is that our ability to use languages to express truth is called into question. This consequence is at the level of languages, and moreover any serious attempt to solve the Liar Paradox therefore either seeks to revolutionise our understanding of languages, or constructs a new formal language to demonstrate why it is not a problem. Since languages are central to solving the Liar Paradox, it is reasonable that they should become the focus of our attempts to understand and explain the paradox. That is, we should focus on understanding the conditions set up by languages which make the Liar Paradox possible.

This approach is further supported by the simple observation that in general there is a much clearer distinction between languages which are affected by the Liar Paradox and those which are not. To take slightly extreme examples, English is obviously affected in some sense, while Classical Sentential Logic is not. The fact that the Liar Paradox does not affect Classical Sentential Logic is not a deep property of that language, it simply does not possess sufficiently rich vocabulary. However this is a property that does not depend in any way on context. Similarly, the fact that English is affected is also context independent. No matter how the world is, the linguistic structures in English allow the construction of liar sentences, and once these are constructable we can run the Liar Paradox argument. This existence of a context independent distinction between the paradoxical and the non-paradoxical allows us to define the Liar Paradox much more easily and effectively.

Each of these reasons strongly suggests that languages should be the primary focus in investigations into the Liar Paradox. They define the conditions which make the Paradox possible, they are the main focus in developing solutions and there seems to be a clear distinction between languages that are affected by the Liar Paradox and those which are not affected.

How are Languages affected by the Liar Paradox?

While we can pick a couple of examples to show that there should be a clear distinction between languages which are affected by the Paradox and those that are not, this distinction cannot be made without a clear idea of what it actually means for a language to be affected by the Liar Paradox. For there are significant differences between the ways that languages can be affected by the Liar Paradox.

The first difference is that some languages may be necessarily affected while others are only contingently affected. Any language in which it is possible to construct examples like "This sentence is not true" case above, will be necessarily affected. No matter how the world is, no matter what facts are true, the language will be affected by the Liar Paradox. However, there will be languages where the problematic sentences are ones like Kripke's examples, where they are paradoxical only if certain facts are true. Kripke's example is only paradoxical if Nixon's assertions about Watergate are perfectly balanced between true and false except for the one that refers to the speaker of the statement. Languages where examples like this are the only paradoxical cases are contingently affected by the Liar Paradox.

The focus of this thesis will be on languages which are necessarily affected for a couple of reasons. Firstly, the languages we are most interested in are necessarily affected. Secondly, constructing a language which is only contingently affected is a non-trivial task, since the linguistic mechanisms which allow the paradox in contingent cases almost always allow necessary cases. Finding clear rules which disallow necessary cases but allow contingent cases is difficult, and will not help us understand or solve the paradox.

Another difference in the way that languages are affected by the Liar Paradox can be illustrated by a couple of sketched cases. Suppose we take the language formed by adding a suitable mechanism for referring to sentences (such as a Gödel Numbering) to a Classical logic and by adding the T-Schema as an axiom schema. So long as there is at least one paradoxical sentence (the Diagonal Lemma is taken to demonstrate that this will be the case) a contradiction will be derivable in the language, and everything will be provable. This means that the language is trivialised - everything is provable and therefore one can no longer assert anything meaningful in it. If we consider English, on the other hand, it is affected by the Liar Paradox in the sense that we can construct sentences which, *under certain assumptions*, trigger the Paradox and hence from which we can derive a contradiction. We will discuss what these assumptions might be later, but the important thing to note is that the Paradox only arises if certain non-trivial assumptions are made.

In the first case, the Liar Paradox arises purely from the definition of the language, and therefore the language is trivialised, and therefore useless. If we wish to fix the situation, it is necessary to change part of the definition of the language and therefore define a new language. Two formal languages with differing definitions and differing sets of theorems are necessarily different languages. In the second case, the paradox does not necessitate that we give up on English and define a new language to use instead. The normal response to the Liar Paradox is to question the assumptions that are necessary for the paradox, as these are not considered to be essential to the definition of English. In the first case the Liar Paradox renders the language useless, and in the second case the Liar Paradox calls into question assumptions about the language.

In order to understand this difference more clearly, it is necessary to examine a key difference between natural languages such as English, and formal languages like Classical Predicate Logic. Formal languages are normally defined so that the key principles of reasoning or logic[19] that apply to that language are a part of the explicit definition of the language. Thus, for example, the difference between the languages of Classical Predicate Logic and that of Intuitionistic Predicate Logic is not in the vocabulary or the grammatical rules on what constitutes a valid sentence (or well formed formula), but in the principles of reasoning (i.e. the axioms and /or rules of inference) defined within those languages. The particular rules of reasoning, whether defined as inference rules or axioms, are an explicit part of the definition of normal formal languages.

Natural languages, on the other hand, do not include an explicit definition of the principles of reasoning that apply to the language. If we come across a string of words such as “The dog that the ball”, we can condemn it as ungrammatical and therefore not an English sentence. On the other had, if we come across an invalid argument, say “Not every dog chases balls, therefore, no cat chases balls”, this is perfectly valid English. However, it is condemned on grounds of reason or logic, not on the grounds that it is not valid English. This suggests that, unlike the case of normal formal languages, principles of reasoning are not intrinsic to the definition of natural languages as one can presumably alter the principles of reasoning without altering the language.

However, it might be commented that natural languages do not have an explicit definitions of anything, so it means little to say that there is no explicit definition of the relevant principles of reasoning in a natural language. Formal languages are explicitly defined and therefore reasoning is explicitly defined; natural languages are not explicitly defined and therefore reasoning is not explicitly defined. While this is strictly true, it does not mean that we cannot draw a relevant distinction between the grammatical rules of English and the principles of reasoning that apply to English, which does not exist for formal languages. It is very rare to have a substantial argument

[19] This is primarily meant to include purely logical reasoning. That is, the reasoning which governs logical connectives and quantifiers.

about whether a particular sentence is grammatically sound, but it is far more common to have a discussion about whether a particular piece of reasoning is actually valid. If a grammatical rule is changed, it is a change to the English language. We accept that a language can change over time due to new grammatical constructions. However, if an accepted principle of reasoning changes, we take it as a change to our understanding of the language. It seems absurd to argue that a philosopher changes English when they present a new philosophy of language. There are however no rules of reasoning that we can change in the case of formal languages without changing the language.

One final challenge to the validity of this distinction is that it is only a surface distinction, and that when one considers the semantics of the relevant languages the distinction disappears. The difference between Classical Logic and Intuitionistic Logic does not only lie in the difference in the principles of reasoning, but also in the different semantic interpretation that the symbols in the different languages have. The different semantics for the different languages, it might be argued, determines the different principles of reasoning. Furthermore, in languages like English, the semantics of logical words like "If" determine the correct principles of reasoning for English, and therefore there is no difference between formal languages and natural languages like English.

While there may be a correct or true semantics for English, the lack of consensus on this matter means that if there is, we do not know what it is. The meaning of logical words such as "If" certainly fixes some rules of inference. It seems highly unlikely that we could accept that someone understands the meaning of "If" if they deny Modus Ponens. However, unlike the case with formal languages, the meaning of "If" does not fully determine the correct rules of reasoning about it. To borrow the terminology of Natural Deduction, while Modus Ponens defines the appropriate Elimination rule of "If", the correct Introduction rule or rules are not always clear. The debates over the nature of conditionals are a good illustration of this. A similar story can be told for many logical words and important concepts in English. The standard meanings of these words partially define the relevant reasoning (in a strict logical sense), but they are not completely defined. This is not the case for formal languages.

Importantly, this difference between standard formal languages and natural languages corresponds to the difference in the way that the Liar Paradox affects languages as identified above. If the principles of reasoning are an essential part of the definition of the language, then if a contradiction arises then the language typically becomes useless. If the principles of reasoning are, at least partly, a matter of interpretation or assumption, then the Liar Paradox called these into question, rather than making the language useless.

Grammar-Only vs Logical Languages

This distinction that has been identified between formal and natural languages is so useful in understanding the Liar Paradox, that we will adopt explicit terminology for the different types of language. We will label languages which explicitly include the logical principles of reasoning for that language, as "Logical Languages" and languages which do not explicitly include principles of reasoning as "Grammar-Only Languages". It should be noted that the sense of Grammar that is being invoked here is that there are rules governing what are and what are not valid sentences. These rules may rely on semantic information such as meanings of words, as they often do in natural languages. Thus a Grammar-Only language is one which has a set of rules about what are valid sentences, but does not explicitly include in its definition complete information about correct logical reasoning about or from sentences. As argued above, natural languages, such as English, count as Grammar-Only languages on this definition.

Although the distinction has been drawn between formal and natural languages, not all formal languages are necessarily logical languages, and not all natural languages are necessarily Grammar-Only Languages. Furthermore, it is often useful to treat a Logical language as having a Grammar-Only part, and a Logical part. Thus, for a formal language, the Grammar-Only part is the set of syntactic rules that determine what are allowable sentences, or well formed formulas. The Logical part are the axioms or rules of inference which determine which sentences are provable and which are not. This means, for example, that Classical and Intuitionistic logics have identical Grammar-Only parts, and only differ in the Logical part.

Importantly, this distinction between Logical and Grammar-Only Languages corresponds to different ways that a language can be affected by the Liar Paradox. A Logical Language is affected in the sense that it may or may not be trivialised by the Liar Paradox. That is, the Liar Paradox may render a Logical Language trivial (and useless) since everything is provable in that language. Since the principles of reasoning for a (fully) Logical Language are completely defined, it makes sense to ask whether or not the Liar Paradox trivialises the language, and almost always there will be a definitive answer. For Grammar-Only Languages, however, it does not make any sense to ask whether such a language is trivialised by the Liar Paradox. Such languages are only defined by a set of grammatical rules, and grammatical rules do not generate sets of provable sentences. Thus it does not make sense to say that the Liar Paradox means that everything is provable in English. Thus Grammar-Only Languages are not affected by the Liar Paradox in the sense that the Liar Paradox can trivialise them.

However, there is a real sense in which Grammar-Only Languages are affected by the Liar Paradox. Languages such as English are affected by the Liar Paradox since sentences can be constructed in such languages that, under the assumption of certain principles of reasoning about the language and terms within the language, lead to the argument typical of the Liar Paradox, and hence a contradiction. Thus Grammar-Only Languages are affected by the Liar Paradox in the sense that sentences (or something equivalent) are constructable in the language which, under the relevant assumptions, trigger the Liar Paradox.

This distinction between Grammar-Only and Logical Languages is useful for understanding the Liar Paradox as it helps clarify the different senses in which one can say that a language can be affected by the Liar Paradox. Languages can affected by the Liar Paradox in the sense that the language is trivialised by the Paradox, or in the sense that sentences (or similar) are constructable within the language which trigger the Paradoxical reasoning under appropriate assumptions. Languages which are trivialised by the Liar Paradox must obviously be also affected in the second sense, and the appropriate assumptions must be reflected in the principles of reasoning defined within the language. Grammar-Only Languages can only be affected in the second way; while Logical Languages can be affected in both ways.

The Problem

This distinction between Grammar-Only and Logical Languages, and the corresponding distinction between the different ways that a language can be affected by the Liar Paradox helps clarify the problem posed by the Liar Paradox, and hence helps identify strategies for solving the problem. As phrased before, the problem posed by the Liar Paradox is that if the paradoxical conclusion is correct, any statement in English can be proven and therefore is true. That is, to use the term adopted in the previous section, English would be trivialised - everything would be provable in it and therefore it would be useless.

However, it was argued in the previous section that the Liar Paradox does not trivialise a Grammar-Only Language such as English, since English does not include explicit rules of inference as part of its definition. Rather, the Liar Paradox affects Grammar-Only Languages in the sense that *under certain assumptions about logical reasoning and truth*, the Paradoxical reasoning arises. The Liar Paradox only trivialises Logical Languages.

We can however take English plus the intuitively correct principles of reasoning and truth accepted in the argument presented above to be one linguistic system, say *English**. In the terminology adopted here, *English** is a Logical Language. The paradoxical argument given above is therefore a proof that *English** is trivialised by the Liar Paradox. We cannot accept the assumptions inherent in the argument without modification, since that leads to an inconsistent logico-linguistic system in which everything is provable.[20] That is, our intuitive understanding of English, Truth and the way that languages represent Truth is inconsistent.

Put in this way, the challenge of identifying the incorrect assumptions in the paradoxical argument becomes the challenge of finding a set of assumptions about truth and reasoning which

[20] Assuming a sufficiently classical logic.

satisfy two conditions. Firstly, they capture the correct concepts and correspond to our intuitions. Secondly, that the Logical Language corresponding to the linguistic system formed by adding these to English (say *English*$^+$) is not trivialised (i.e. affected in the more substantial sense) by the Liar Paradox.

The appropriateness of this way of framing the problem can be seen in the numerous attempts in the literature to define a formal solution to the Liar Paradox. The main components of a formal solution are a formal language and a truth definition for that language, and the aim is to develop such a solution which is not affected by the Liar Paradox. The aim is that a correct formal solution would demonstrate the logical structure for *English*$^+$, the correct linguistic system which includes English and the correct assumptions about reasoning and truth. That is, these formal solutions are designed to embody the correct assumptions, and to demonstrate the mechanism by which the Paradox is avoided is we adopt the correct understanding.

Unfortunately, as commented before, there exist too many solutions rather than no good ones. There exist many formal solutions to the Liar Paradox in which the formal language is not trivialised. That means philosophically that there are many candidates for *English*$^+$ which are not affected by the Liar Paradox in the sense that *English*$^+$ is not trivialised. This should mean that the challenge that we face is choosing between these competing candidates, and developing some criteria by which we can choose between them. However, while all of these formal solutions provide a Logical Language which is not affected by the Liar Paradox, it does not mean that they can all function as candidates for *English*$^+$.

In a philosophical follow-up to his formal definition of truth, Alfred Tarski offered a powerful argument that, in the terminology adopted here, any Logical Language which satisfies certain plausible assumptions is necessarily inconsistent.[21] The key assumptions in his argument were articulated by Tarski as follows:

> (I)-We have implicitly assumed that the language in which the antinomy is constructed contains, in addition to its expressions, also the names of these expressions, as well as semantic terms such as the term "*true*" referring to sentences of this language; me have also assumed that all sentences which determine the adequate usage of this term can be asserted in the language. A language with these properties will be called "*semantically closed.*"
>
> (II) We have assumed that in this language the ordinary laws of logic hold.[22]

There is a third principle that is crucial to his argument, which Tarski considered that he had established as correct, what is now known as the T-Schema. For Tarski this is expressed as: *"X is*

[21] For the clearest account see Alfred Tarski. "The Semantic Conception of Truth: and the Foundations of Semantics". In: *Philosophy and Phenomenological Research* 4.3 (1944), pp. 341–376

[22] Tarski, "The Semantic Conception of Truth", see n. 21, p. 348

true if, and only if, p."[23] Tarski uses X as a name for the sentence p. What is important to note in this context is that the first is an assumption about the Grammar-Only part of a Logical Language, the second is an assumption about correct reasoning and the third is an explicit assumption about Truth. To translate his point into our terminology, Tarski argued that any Logical Language that includes a plausible truth definition (i.e. one that satisfies the T-Schema), that satisfies the ordinary laws of logic and whose Grammar-Only part is semantically closed, is necessarily trivialised by the Liar Paradox.

English$^+$, whatever the correct logical and alethic assumptions are, is a Logical Language whose Grammar-Only part (English) is semantically closed. For this reason, Tarski thought that English was necessarily inconsistent and needed to be replaced by a scientific, logical language for correct discussion of truth.[24] This conclusion of Tarki's has justifiably been rejected by modern philosophers and logicians, but it reveals an important aspect to the challenge of providing a formal solution which is a plausible candidate for *English*$^+$. The Grammar-Only part of *English*$^+$ is semantically closed, and therefore the Grammar-Only part of any formal language which provides a formal solution must be semantically closed. Otherwise, the formal solution cannot correctly capture the logical structure of *English*$^+$. That is, if we want a formal solution which correctly deals with the Liar Paradox as it poses a philosophical challenge for natural languages as we use them, that solution must be semantically closed.

However the vast majority of (and arguably all) formal solutions to the Liar Paradox are not semantically closed. This means that these solutions cannot capture the correct assumptions about reasoning and truth which allow us to reason satisfactorily in English in the ways we habitually do. They can only be accepted as a valid solution if we change the way we use English, if they can be accepted at all. The aim of this thesis is to provide a new formal solution which allows a semantically closed Grammar-Only part to a Logical Language and therefore is a plausible candidate for *English*$^+$.

Summary of rest of the Thesis

The development of a new formal solution must be grounded in an accurate analysis of the Paradox. It was argued above that investigating languages offers a clear method for differentiating between the paradoxical and the non-paradoxical, and this differentiation will be the focus of investigations in the first chapter. Given that two different ways that languages can be affected by the Liar Paradox, it follows that there are two different distinctions that need to be drawn and we can divide this investigation into two projects. The first distinction is the line between Grammar-Only languages that are affected and those that are not. That is, we need to identify which properties a

[23] Tarski, "The Semantic Conception of Truth", see n. 21, p. 344

[24] Alfred Tarski. "The Concept of Truth in Formalized Languages". In: *Logic, Semantics and Metamathematics.* Oxford University Press, 1956, pp. 152–278, p. 165

Grammar-Only Language must possess for Liar Sentences to be constructable within that language. The first project therefore is to develop an account of the grammatical conditions necessary for Liar Sentences to be constructable in a (Grammar-Only) Language.

The second distinction is between those Logical Languages which are trivialised by the Liar Paradox and those which are not. We need to uncover which principles a Logical Language must satisfy for the Liar Paradox to trivialise that language. As noted before, every Logical Language includes a Grammar-Only part, some rules about which sentences are grammatically valid. In formal languages, these are normally purely syntactic rules. The Grammar-Only part of a Logical Language must therefore possess the properties necessary for Liar Sentences to be constructable within that language, which will be identified in the first project. This means that the second project is to provide an account of the principles of reasoning necessary for the Paradoxical reasoning to produce a contradiction and hence trivialise the relevant (Logical) language. These two projects will be tackled in the first chapter, with the aim of providing a comprehensive account of which languages are affected by the Liar Paradox (in both senses) and therefore a deeper understanding of what the Liar Paradox is.

The second chapter will examine various formal solutions that have been advanced. Tarski's work will be a particular focus, since Tarski more than anyone else is responsible for defining the parameters of the modern debates about the Liar Paradox. Moreover, the various formal solutions which have been advanced since Tarski can be understood as attempts to construct non-trivial Logical Languages which are as close to being semantically closed as possible. It will be argued that many existing formal solutions suffer from a systematic limitation. They cannot, within the logical language, assign certain sentences about Liar Sentences the correct truth status, according to the internal properties of the solutions themselves.

The aim of the third chapter is to track down the cause of this systematic limitation. The fact that there is a common limitation within existing solutions suggests either that it an unsolvable problem, or that there is a systematic flaw in the strategies that are currently being used. It is important here to note the difference between what is here defined as a logical language and a formal language. A logical language is a language which includes an explicit definition of its principles of reasoning. A formal language is a language in which a mathematical or symbolic logic is defined. Given that the formal solutions that are advanced are constructed within modern formal languages, it is vital to question whether there are assumptions within modern formal logic which are incompatible with a satisfactory solution to the Liar Paradox. It will be argued that two key semantic assumptions about formal languages, namely that truth values are completely defined with respect to a semantic model and that sentence types are the primary truth bearers, are incompatible with a satisfactory philosophical solution to the Liar Paradox.

The second half of the thesis will turn to developing a more satisfactory solution to the Liar

Paradox on the basis of the analysis in the third chapter. The fourth chapter will develop a blueprint for a more satisfactory formal truth definition by focusing on our understanding of how truth is defined in a language, and particularly the way that we evaluate the truth value of sentences which include the truth predicate. With this focus, the interaction between predicates like truth and the reference structures necessarily for the definition of a truth predicate become very significant, and it is shown that care is needed in the definition of a truth predicate. This blueprint will be developed entirely independently of any particular formal logic, and will therefore be applicable to any potential formal logic. The blueprint is shown to be successful in solving the Liar Paradox in a very appealing way in the fifth chapter, where it is implemented in a Classical Sentential Logic. This definition is provably consistent, which is a remarkable result.

The sixth chapter builds on and improves the blueprint from the fourth and fifth chapters by exploring the questions about the appropriate truth bearers that arose in the analysis in the third chapter. It is argued treating name/sentence pairs as truth bearers in formal languages allows us to improve the definition in the fifth chapter to a consistent, semantically closed formal truth definition. Once again, this is defined in a Classical Sentential Logic, and arguably provides a philosophically satisfactory formal truth definition.

The seventh chapter concludes the thesis by summarising the key features of the approach adopted in chapters four to six, and discusses its applicability to understanding truth in natural languages. It is argued that the key problem the Liar Paradox produces, that we cannot be sure our use of natural languages is consistent or coherent, is solved as this approach provides an understanding of truth which is consistent, applicable in natural languages and semantically closed.

Chapter 1

Understanding the Liar Paradox

The key point that was made in the Introduction is that the most important type of thing to investigate when considering the Liar Paradox is languages. There is a clear distinction between paradoxical and non-paradoxical languages, and languages set up the conditions which make the paradox possible. Moreover, the key philosophical problem posed by the Liar Paradox is a problem about whether our use of natural languages is consistent and therefore whether we can actually use them in the way that we assume we do. This change of focus from sentences or cases to languages means it is necessary to rethink our understanding of the Liar Paradox.

As argued in the Introduction, drawing a distinction between the paradoxical and the non-paradoxical is a key task in understanding a paradox. Given that languages set up the conditions which make the Liar Paradox possible, we will look at the conditions under which languages are affected by the paradox. Languages which are affected by the Liar Paradox will satisfy these conditions. Therefore identifying these conditions will provide a set of criteria which distinguish languages which are affected by the Liar Paradox from those which are not. This investigation will proceed from the two different types of language identified in the Introduction, as Grammar-Only and Logical Languages are affected differently by the Liar Paradox.

1.1 Why Languages?

The Introduction introduced the idea that languages are central and essential to any understanding of the Liar Paradox, but it is important to make this idea and the reasons for it clear. The concept of language at play is a very broad one, which encompasses, at a minimum, natural languages, formal languages and potential formalisations of intuitive reasoning. As a paradoxical argument, the Liar Paradox obviously depends on principles about language and certain principles of reasoning to get off the ground. Also crucial to the our investigation is what we will refer to as the logico-linguistic structure of a language, that is the collection of logical and linguistic (in a broad sense) principles which hold when we consider the language in question. However, this logico-linguistic structure depends on the language in question, and in any case Logical Languages incorporate much of it into their explicit definition. Languages therefore will be taken to be our primary focus of investigation.

The importance of languages in investigating the Paradox can be seen firstly in the fact that the Liar Paradox calls into question our ordinary use of natural languages. The grammatical rules of natural languages allow us to build and use sentences in particular ways which are inconsistent on our intuitive understanding of the logico-linguistic structure of natural languages. Moreover, in the literature, the standard approach to presenting a seriously worked through solution to the Liar Paradox is to provide a formal truth definition in a formal language. The key point to note about these solutions is that invariably it is the language which is modified more heavily to deal with the Paradox, rather than the truth definition. That is, the problem posed by the Liar Paradox, and our approach to its solution turn on languages, which is good evidence that language should be central to any investigation.

More importantly however is the fact that it is a language which establishes the conditions which make the Liar Paradox possible. When the grammar of a language satisfies certain conditions, it will follow that a liar sentence will be constructable; and when the logico-linguistic structure of that language satisfies further conditions, then that sentence will give rise to a contradiction. Without the relevant conditions being satisfied at the language level, the Paradox will not arise.

This is particularly important since one of the important features of the Liar Paradox is that it arises so long as a relevant example is constructable within a language. The structure of the paradoxical argument – it requires only the *assumption* that a sentence is true (or not true) – means that the paradoxical reasoning will arise even if a problematic sentence is never actually used or constructed. The important fact is therefore whether problem cases are constructable, not whether they exist. Thus the key factor in determining whether the Liar Paradox arises is whether the language allows the relevant cases to be constructable. The properties of the language are central to the Liar Paradox, and we will therefore be focusing the investigation on languages.

1.1.1 The Focus on Languages

The goal of focusing on languages in understanding the Liar Paradox is to identify the conditions a language must satisfy for it to be affected by the Liar Paradox. This will help identify precisely what it is that leads to the Paradox arising, and therefore clarify the possible approaches to resolving the paradox. There were, however, two different types of languages identified in the Introduction which were affected by the Liar Paradox in different ways. These two types of languages will therefore be investigated separately and two separate sets of criteria will be devised.

The first will identify when a Grammar-Only Language is affected by the Liar Paradox, in the sense that a contradiction arises under certain assumptions about reasoning and truth. The second will identify when a Logical Language is affected, in the sense that it is trivialised. These criteria, if successful, will accurately distinguish between all the languages which are affected by the Liar Paradox, and all languages that are not; and it is a distinction that cannot be affected by the Revenge Problem. It will therefore provide a solid conceptual basis on which a solution to the Liar Paradox can be built.

An in depth analysis of the role of languages in the Liar Paradox may nevertheless seem to be unnecessary, since it does not seem difficult to identify what the Liar Paradox is. There are certain sentences which are true if, and only if, they are not true; and therefore give rise to a contradiction. These sentences are typically formed by setting up a situation where the sentence ultimately says of itself that it is not true, possibly via other sentences. While this is correct, analysing the Liar Paradox in this way does not provide any insight into why these sentences arise, or the conditions under which they cause a paradox. Without a clear understanding of these, we cannot have truly understood the Paradox and do not have a strong basis for formulating a resolution.

This however does not imply that paradoxical sentences and situations are not important, and are not to be investigated. There is no Liar Paradox if there are no paradoxical sentences or situations. However, while they will be investigated, the focus of the investigation must be different. Rather then trying to understand what it is which makes these cases paradoxical, the aim will be to identify the necessary linguistic and logical conditions which make these cases possible. Paradoxical cases arise only if certain linguistic and logical structures are available, and these structures are only available in languages that are affected by the Liar Paradox. The focus of investigation into paradoxical sentences and situations will therefore be on the minimum set of linguistic and logical conditions which could make the particular cases possible. The aim is to use different cases to build a set of necessary conditions which all languages which are affected by the Liar Paradox must satisfy.

1.1.2 Separating the Liar Paradox from other Paradoxes

Before this can be done, there remains one factor which needs to be clarified. It may seem like a minor point, but we need to be able to identify what it is that make the Liar Paradox the *Liar* Paradox, as opposed to any other Paradox. Trivially, any genuine paradox will lead to an inconsistency and a trivialised logical language. If we are to precisely differentiate languages which are affected by the Liar Paradox from languages which are not affected, it is important to have a clear idea of what the Liar Paradox is, as different logical languages will be affected by different paradoxes. Without a clear distinction between the Liar Paradox and other paradoxes, it is not possible to ensure that any developed distinctions relate to the Liar Paradox, and not to paradoxes in a more general sense.

It should be noted that we would hope that any solution that is developed on this basis could solve other paradoxes, particularly closely related paradoxes like Curry's Paradox. However, the aim is to develop a solution to the *Liar* Paradox, and for this it is important to precisely understand the *Liar* Paradox. If we discover, on this basis, that there are other paradoxes that are caused by the same assumptions or mechanisms, then this discovery should be made from a precise understanding of each of the paradoxes separately, not by associating the paradoxes as similar from the beginning of the investigation.

The most obvious feature that distinguishes the Liar Paradox from some other paradoxes is that, as the *Liar*, it has something to do with truth. Moreover, simple Liar Sentences appear to be distinguished by the fact that the content of the sentence is not consistent with the sentence being either true or not true. In the most common case, what the sentence says ("This sentence is not true") directly contradicts the conditions under which the sentence, as a declarative sentence, could be true (or not). That is, for the sentence as a declarative sentence to be true, what it says must hold. However, the semantic content of the sentence (what it says about what is the case) means that this cannot be the case. The semantic content of the sentence is inconsistent with the standard conditions under which the sentence would be true.

The concept of the *semantic content* of a sentence being adopted here is to be understood as what the sentence says is the case, or similarly, as what a competent user of the language would understand about the way things are by understanding the sentence. Under different philosophies of language, it could be equivalent to concepts such as the meaning of the sentence, or the truth conditions. Here we are trying to analyse the situation without recourse to any specific philosophy of language, and so will not decide between any of the alternatives. The key fact is that the semantic content of a sentence is the information we use to test whether the sentence is true or not. Intuitively, if the semantic content (what the sentence says) holds, then the sentence is true and vice versa.

It is important to note that the contradiction which arises in the Liar argument is a 'direct' contradiction. That is, the semantic content contradicts the relevant conditions themselves. It is not the case that the combination of what the sentences says and some truth conditions together imply a contradiction, but that they contradict each other. This fact is reflected in the argument structure that is distinctive of the Liar Paradox: If we assume that a particular sentence is true, then it follows that that sentence is not true; similarly, if we assume that the sentence is not true; then it follows that it is true. The entailment in the case of the Liar Paradox by which each conclusion follows is very strict. We do not need any auxiliary hypotheses or information beyond what the relevant sentences say (the semantic content) and the conditions on the relevant sentences (as declarative sentences) being true. Moreover, the contradictory conclusion is not reached via the derivation of any other inconsistency, but follows directly from these relevant facts. This direct contradiction also means that the arguments are symmetrical when one assumes truth or one assumes untruth.

This fact that no auxiliary hypotheses or information beyond the semantic content of the sentences and the conditions on the relevant sentences being true are needed in deriving the contradiction holds for multiple sentence versions of the Liar Paradox. In the case of a multiple sentence version of the Liar Paradox, if we assume that any one of the relevant sentences is true (or not true), then simply from what the various sentences say and the relevant conditions on the sentences as declarative sentences being true, it follows that the initial sentence is not true (or true). Importantly, every sentence in the derivation will either be one of the sentences or its negation, a statement which expresses the semantic content of a sentence, or a statement of the conditions on a declarative sentence being true. No other type of sentence is necessary in the derivation of the contradiction, and the first contradiction that can be derived contradicts the initial sentence.

This strict characterisation of the Liar Paradox allows us to differentiate the Liar Paradox from both Curry's and Yablo's Paradoxes. In the case of Yablo's Paradox, the argument structure is not symmetrical for the assumption of truth versus the assumption of untruth. If we assume that a relevant sentence in a Yablo series is not true, then it is not possible to prove that that sentence must be true without first proving a separate contradiction. It follows that Yablo's Paradox is not a case of the Liar Paradox, which is assumed by the fact that we use different names for the two different types of paradox.

In the case of Curry's Paradox, the classic exposition turns on the fact that from a particular type of sentence, it is possible to prove any sentence in the language. The fact that we could prove that the sentence itself is not true is not essential to the paradox in the way that it is in the definition of the Liar above. It does not mean that there are not cases where Curry's Paradox and the Liar Paradox coincide. There are obvious examples where a sentence can be both a Curry Sentence and a Liar Sentence, but the two paradoxes are distinct as paradoxes.

Nevertheless, the fact that it is possible for certain examples to count as both Liar Sentences and Curry Sentences illustrates the two paradoxes are closely related. It turns out, moreover, that the solution advanced later deals equally well with both of these Paradoxes. However, this does not imply that they are the same paradox. In particular, it will be shown below that there are languages which can be affected by one of these paradoxes and not the other. However, the central mechanism which gives rise to the paradox is identical for both paradoxes and so they can share a common solution. However, since the aim of this chapter and thesis is to deal with the Liar Paradox, only languages which are affected by the Liar Paradox will be explicitly dealt with.

1.2 Grammar-Only Languages

As mentioned above, the investigation into a set of criteria which differentiate languages which are affected by the Liar Paradox from those which are not will be broken into two parts. The first is an investigation into the set of necessary and sufficient criteria for a Grammar-Only Language to be affected by the Liar Paradox. That is, the aim is to develop a set of criteria which identifies those languages for which, under the assumption of certain sets of principles of reasoning, a contradiction is derivable in exactly the pattern described. As we are working with Grammar-Only languages, these principles of reasoning are not part of the definition of the language, but are rather part of either an assumed understanding of the language, or of a logical language of which the Grammar-Only language is a part.

Since it is not possible to survey all possible Grammar-Only Languages, the development of these criteria cannot proceed through a systematic survey of all affected Grammar-Only Languages. We will instead begin with an initial set of criteria which are clearly sufficient to produce the Liar Paradox. That is any language which has these criteria will be affected by the Liar Paradox. Each member of this set will be challenged individually in order to determine what is actually necessary to produce the Liar Paradox. In this way we will produce criteria which are both necessary and sufficient. We will begin with an intuitive set of criteria that is developed from the example of the Liar Paradox given in the Introduction.

The example of the Liar Paradox given above was triggered by the following sentence:

1. This sentence is not true.

Since we are concentrating on Grammar-Only Languages, we want a set of criteria which need to be satisfied if this sentence is to be grammatically acceptable. With this in mind, it seems clear that (1) is only grammatically acceptable in a language which has the following properties:

- It has a truth predicate.
- Sentences can refer to themselves.
- Truth can be correctly predicated of sentences.
- It has a negation.
- There are no restrictions on combining the negation, the truth predicate and a reference to other sentences within the scope of the truth predicate within a sentence.

If any of these five conditions is missing, then sentence (1) is potentially ungrammatical, so the set of conditions seems reasonable. Furthermore, if any language satisfies this set of five conditions, it will be affected by the Liar Paradox, since an equivalent example to (1) can be constructed. Hence these five are sufficient conditions for a language to be affected by the Liar Paradox. However, it can be shown that as written none of them are strictly necessary conditions.

1.2.1 Truth Predicate

The first condition in the set above is that the Language includes a truth predicate. This condition is in fact implicitly assumed within much of the debate about the Liar Paradox. The Liar Paradox is normally considered to be a paradox about truth, and the formal focus is on correctly defining a formal truth predicate. However, it is not necessary that a language contains a specific truth predicate. For example, consider the following sentence:

2: What this sentence says is not the case.

Although there is no mention of truth within this sentence, the paradoxical reasoning arises. If what sentence 2) says is true, then as 2) says that what 2) says is not the case, what 2) says is not the case, and hence 2) is not true. Similarly, if what 2) says is not true, then as that is what 2) says, what 2) says is actually the case. This reasoning is identical to the standard Liar reasoning, even though there is no truth predicate in the sentence (and the truth predicate can also be removed from the argument). That a language possesses a truth predicate is therefore not a necessary condition on that language being affected by the Liar Paradox.

This does not, however, mean that the Liar Paradox has nothing to do with the concept of truth. In the case of the sentence 2), the obvious comment is that although the sentence does not use a Truth *Predicate*, it still makes implicit use of the *concept* of truth, or at least a very similar concept. To claim that what some sentence says is the case is equivalent to saying that that sentence is true. The conditions on either of these assertion being correct are in at least all reasonable cases identical. Furthermore, the circumlocution “what sentence X says is the case” is one way in which the concept of truth can be clarified. Thus, although the sentence 2) does not explicitly use a truth predicate, it contains an expression which expresses the concept of truth.

It is not difficult to see that this could be classed as a necessary condition on cases of the Liar Paradox, and therefore languages which are affected by the Paradox. The structure of the paradoxical argument turns on a direct conflict between the standard conditions on a sentence being true, and the content or meaning of that sentence. This conflict is only possible if the content of a sentence can expresses something about the truth or untruth of a relevant sentence. Without this linguistic ability, the necessary conflict cannot arise and there can be no Paradox. Given that the ability to express something about the truth or untruth of a sentence requires the ability to express the concept of truth within the relevant language, it is clear that this ability is a necessary condition on the Liar Paradox affecting a particular language. That is, it appears that only Grammar-Only languages which can express a concept equivalent to truth are affected by the

Liar Paradox.

1.2.2 Sentences

While it seems necessary that a Grammar-Only language can express a concept equivalent to truth in order that the Liar Paradox can arise, it is clear that this is not enough. The second condition listed above, that "Truth can be correctly predicated of sentences", is also required for otherwise the contradiction cannot arise. The Liar Paradox turns on the ability of a sentence to identify some sentence (possibly itself) as true, when other conditions rule that it should be not true. If truth cannot be correctly predicated of sentences, then this ability does not exist in the language, and therefore the Liar Paradox cannot arise.

However, it does not follow that, as stated, this condition is a necessary condition. Something along these lines is necessary, but the statement of the condition that "Truth can be correctly predicated of sentences" makes assumptions that are not required in general. In particular, the assumption of sentences is not necessary, and not simply the assumption that sentences are truth-bearers. The inherent assumption that a language needs to contain sentences (or equivalently complex grammatical constructions) is not necessary for the Liar Paradox. The Liar Paradox can affect very simple languages, which do not contain any of the grammatical complexity of ordinary languages.

For example, consider a very simple language which includes two gestures. "Thumbs up" corresponds to saying that what has been asserted is correct. "Thumbs down" means the opposite, that what is asserted is incorrect. One could further imagine, if necessary that these signs are part of say a "Caveman" language which has certain sounds or symbols for things like "I am hungry"; or "Dangerous animal coming". Such a language would be extremely simple, yet the "Thumbs up" and "Thumbs down" symbols would have a very useful role is confirming or denying information.

However, one can further easily consider the situation where Person A gives the "Thumbs up" sign to Person B; while Person B is giving a "Thumbs down" to Person A. If what A "asserts" is true, then it means that what B asserts is true; which means that what A asserts is not true. Similarly, if what A asserts is not true; then B is correct in giving the "Thumbs down" sign; but that means that what A asserts is actually true. We have a classic Liar Paradox.

This means firstly that grammatical complexity is not necessary for the Liar Paradox to arise, since it can arise in such a simple language. We cannot therefore assume any particular complexity or type of language as necessary for a solution to the Liar Paradox. Any set of symbols, which has a potential interpretation or which means something, can potentially be affected by the Liar Paradox. Given that this is a very minimal account of what a language is (a set of symbols with a potential interpretation); there are no restrictions on what type of languages can be affected by the Liar Paradox.

Secondly, this shows that many of the debates about the correct truth bearer are not always relevant to understanding the Liar Paradox. It is not at all clear how we would classify the problematic gestures in the Paradoxical example. They are not sentences or sentence tokens on any normal definition, and while they might assert a statement or express a proposition it is not clear at all how to understand these things. One might also want to say that the relevant signs or gestures are true or not, as the case may be. In any case, something ought to be true or not true in this situation, and this is all that is necessary for the Paradox to arise: (at least many of) the symbols which make up the language must be capable of either being true or not true; or expressing something which is true or not. This is nothing remarkable, since it follows directly from the definition of a language as a set of symbols with an interpretation.

Furthermore, it is important to note that however we interpret the correct truth bearers, it does not affect whether a Grammar-Only Language is affected by the Liar Paradox. The definition of a Grammar-Only language being affected by the Liar Paradox is that it produces the Paradox under *some* understanding of the language. For any choice of truth bearer, there will be a corresponding understanding (or a corresponding set of logical assumptions) which gives rise to the Paradox. While debates about the correct truth bearer may be relevant to the correct understanding of a language, they do not affect whether a Grammar-Only language is affected by the Liar Paradox.

As a matter of convenience and terminology, sentences will be habitually referred to as the normal truth bearers in this thesis. This does not reflect any philosophical conviction, but rather reflects both the necessity of having some term to use and the formal interests of the investigation. Formal languages are typically defined so that sentences are the truth-bearers. However, it is almost always possible to substitute "sentence" for any other truth bearer and when the appropriate modifications are made, the arguments will be still valid.

1.2.3 Negation

If we assume that a necessary condition on a language being affected (in any sense) by the Liar Paradox is that it is possible to express the concept of truth within that language, it is obvious that this condition is not sufficient. The concept of truth, by itself, cannot generate the contradiction within the Liar Paradox. It can only generate the non-contradictory puzzles surrounding the Truth - Teller, that is cases such as "This sentence is true." In these cases, no contradiction can be derived, but there is no apparent way to derive the truth or falsity of the relevant sentence.

Obviously, if we have a way of expressing the concept of truth within a language, it is also necessary to have a negation within that language. The Liar Paradox turns on the fact that it necessarily follows from the content of a sentence that that sentence is not true. Without the negation, the Paradox cannot be formed. It is not necessary to assume anything substantial about the behaviour of the negation within the language. Almost any negation is sufficient for these purposes.

However, if we want to identify the truly necessary conditions, the separate existence of a negation and a way of expressing the concept of truth is not required. So long as there is a method of expressing the concept of "untruth", or of not being true, and this can be predicated of other 'sentences' then it is possible to produce a paradoxical example. Thus for example, it is possible that a language could have the ability to express the concept of Falsity without being able to directly express the concept of Truth. The Liar Paradox would still arise in this language. While the types of formal and natural languages that are commonly dealt with all include a method of expressing Truth and a negation, this is not necessary for the Paradox to arise.

It might seem as though this point is unimportant, since all of the languages that we are interested in include a negation and a way of expressing Truth. However it is important to the way that we understand the paradox, and what provides a satisfactory solution. A solution, for example, which relies on a different definition of negation cannot be a general solution, since it cannot deal with languages which do not include a negation, yet are still affected by the Liar Paradox.

The first necessary condition on Grammar Only languages being affected by the Liar Paradox is that the language includes a method of expressing untruth. Normally we would expect this to be via a method of expressing truth and a negation, but this is not strictly necessary.

1.2.4 Self-Reference

It has been often pointed out that self-reference, although included in the list of sufficient conditions above, is not necessary for the Liar Paradox. The most obvious examples are the multiple sentence versions of the Liar Paradox, where a set of sentences all refer to each other and the end result is a reference loop which is paradoxical. A simple example is the following pair of sentences:

3: The following sentence is true.

4: The previous sentence is not true.

Assuming any truth value for these sentences leads to contradiction, in the same way that sentence (1) leads to a contradiction. This means that the argument has exactly the same structure and therefore, by the characterisation above, we have a case of the Liar Paradox. This case, however, does not include any self-reference, so the possibility of self-reference is not a necessary condition for the Liar Paradox to affect a language.

This type of case, and any similar ones, suggest that cases which rely on self-referentiality are only special cases of a broader phenomenon involving extended circular reference loops. The paradox occurs because there is a set of sentences which each refer to some other sentence(s) in the set and the combination means that there is a closed loop of reference. If we follow the path of reference we eventually get back to where we started.

It was argued above that the distinguishing feature of the Liar Paradox is that a contradiction can be derived only using the sentences themselves, the semantic content of the sentences and the conditions on the sentences as declarative sentences being true; and that the first contradiction derived contradicts the initial assumption. We can see here that the condition on sentence (3) being true as a declarative sentence is that sentence (4) is true (since this is the semantic content of (3)). However, if sentence (4) is true, then by what (4) says, it must be the case that (3) is not true, from which it follows that (4) is also not true. Thus a contradiction arises without having to appeal to any other sentences or broader information, and the equivalent argument holds if we assume that (4) is not true.

This type of argument structure is only possible if we can argument back to a conclusion about the sentence we started with. Given that the steps allowed in the argument are only the sentences, their semantic content and the conditions on them being true as declarative sentences, the only available method for progressing from one sentence to another is by a sentence referring to another. In this case the semantic content (what it says) will identify a further sentence. The fact that we get back to where we started means that there is some chain of reference that leads back to the original sentence. This is only possible if circular reference loops are possible within the language.

Thus, it seems necessary for the Liar Paradox to arise that a language allows circular loops of reference - chains of reference where we get back to where we started. This condition is, however, hard to test as it is not immediately apparent from the definition of a language whether it will allow complicated structures such as circular loops of reference. However, it is easy to test firstly whether the language allows sentences to refer to other sentences; and secondly whether there are any restrictions on which sentences can refer to other sentences. If the language allows sentences to refer to other sentences and there are no restrictions on this, then the combinatorics of reference mean that (unless the languages has a very limited range of sentences) circular reference loops will almost certainly arise. The key condition that determines whether the Liar Paradox is possible is whether circular loops of reference arise. If sentences are allowed to refer to other sentences in the language, then circular loops of reference will arise unless the reference is restricted in a way to prevent this occurring.

1.2.5 Reference

More needs to be said about exactly what is meant by reference. In the examples above, reference has been assumed to work on a naming model - we refer to a sentence by inserting a name of that sentence in another sentence. This is a simple type of reference which is not representative of all possible types of reference, or even of all examples of the Liar Paradox.

Perhaps the most interesting example is the liar sentence advanced by Quine. In this case there is no explicit reference, or naming, of the sentence itself, although the sentence appears paradoxical. Quine's example is the following:

E: "yields a falsehood when preceded by its own quotation" yields a falsehood when preceded by its own quotation.[1]

Assuming that 'falsehood' is to be interpreted as meaning a false sentence,[2] then this sentence is a correct sentence of English and it is paradoxical. The paradoxical reasoning which characterises the Liar Paradox arises in the following way. If the predicate "yields a falsehood when preceded by its own quotation" is true when preceded by its own quotation, then E is true. However, E says that this predicate yields a falsehood when preceded by its own quotation, so what it says is false, which contradicts the assumption that it is true.

Similarly, if "yields a falsehood when preceded by its own quotation" actually yields a falsehood when preceded by its own quotation, then what the above sentence says is true. As the above sentence is the result of preceding "yields a falsehood when preceded by its own quotation" by its own quotation, then the predicate cannot yield a falsehood. Again we have a contradiction.

This sentence, however, does not appear to refer to any other sentences, at least in the same way that the sentences above referred to other sentences. However, if we assume as we must that 'falsehood' means a 'false sentence', the E provides a description of a sentence which it is talking about. This sentence is the sentence that is produced when "yields a falsehood when preceded by its own quotation" is preceded by its own quotation. But this sentence which is described and therefore referred to is the sentence E itself. Hence, because of the meaning of the sentence, E refers to itself.

Thus the sentence refers to itself, only via the meaning of the sentence rather than through an explicit naming of a sentence. However, this weaker sense of reference is sufficient to give rise to the Paradox. In fact, if we consider the paradoxical argument, any means of reference from one sentence to another will suffice. Thus reference in the list of criteria has to be taken in the broadest possible sentence. So long as the reference allows truth (or untruth) to be predicated of the sentence being referred to, the paradox can arise.

1.2.6 Restrictions

This final item on the initial list is "There are no restrictions on combining the negation, the truth predicate and a reference within a sentence." As with the previous items, this item is not strictly necessary. However, this is not due to substance of the item but rather to its exact wording. It has

[1] This is sometimes referred to as Quine's Paradox, see W. V. O. Quine. "The Ways of Paradox". In: *The Ways of Paradox and Other Essays*. Cambridge, Mass.: Harvard University Press, 1966, pp. 1–21

[2] Appending a predicate to a quotation yields a sentence, so we must interpret falsehood as a false sentence for this sentence to be grammatically correct.

been shown that negation, a truth predicate and sentences are not strictly necessary, so as worded this item cannot be necessary. However, the principle holds in general. So long as a language includes restrictions on the way that i) its method of expressing untruth of entities in the language and ii) its way of referring to these entities can interact, it is possible that the Liar Paradox will not arise.

To make up an example, it is possible that a language allows truth to be predicated of other sentences, but the negation of truth cannot be predicated of other sentences. In this case the Liar Paradox would not arise, even though the language satisfies the other necessary conditions. Similarly, there might be a language which allows sentences to refer to other sentences, but not in the scope of the truth predicate. The Liar Paradox is only in general possible if there are no restrictions on the way that the other necessary conditions interact so that unrestricted reference within the scope of the (un)truth predicate is possible.

1.2.7 Summary

In summary, we have developed a set of four necessary conditions for Grammar-Only Language to be affected by the Liar Paradox. If any language satisfies all of these conditions, then it will be affected by the Liar Paradox. That is, under appropriate assumptions about reasoning, truth and possibly about matters of fact, a contradiction arises. However, any language that does not satisfy all of these will not be affected by the Liar Paradox.

The list of conditions is as follows:

1. The language has a way of expressing the concept of untruth.

2. This concept can be predicated of/applied to the types of things which are used to express facts/information.

3. The types of things which are used to express facts can refer, by whatever means, to other things of this type.

4. There are no restrictions on the way that the mechanisms in the first three points can be combined. That is, there are no restrictions on the application of the concept of untruth to the types of things which express facts within the language.[3]

The phrasing of these necessary conditions is deliberately broad, so as to not to limit these conditions to any types of languages. As argued, we cannot make any assumptions about the complexity of the languages involved, or about the specific mechanisms involved. Thus, as mentioned, although the first condition will normally be constructed using a negation and a truth predicate, languages without this can still be paradoxical.

[3] The key point is that this condition allows circular reference loops to arise.

It is interesting to point out that if we conduct the same types of analysis for Curry's Paradox, then conditions 2), 3) and 4) will remain essentially the same, but the first condition will have to change. In the case of Curry's Paradox, it is necessary that there is a way of expressing the concept of truth, and that the language contains a conditional (or equivalent).

Obviously, English and other natural languages satisfy these four conditions. Moreover, there is no way that these conditions can be restricted in a natural language to prevent the Liar Paradox without significantly altering the language in question.

1.3 Logical Languages

The second major part of this chapter is to look at the necessary conditions for Logical Languages to be affected by the Liar Paradox. Given that any Logical language can be separated into a Grammar-Only part and a Logical part, the first thing to note is that the Grammar-Only part of a Logical language has to be satisfy the conditions listed above if the Logical language is affected by the Liar Paradox. If the Grammar-Only part is not affected, then no potentially paradoxical sentences can be constructed and therefore the Logical language cannot be affected. What we are interested in in this section are therefore the conditions that the Logical Part needs to satisfy in order that the Logical Language is affected.

In the Introduction, a Logical Language was defined to be affected by the Liar Paradox only if it is trivialised, that is, only if everything is provable in the language. The reason for adopting this criteria is that a number of important solutions to the Liar Paradox have been advanced that accept the validity of some contradictions. It follows that we cannot use inconsistency as the mark of an affected Logical Language, since that would rule out these solutions as valid without even considering them. However, the traditional reason for ruling out inconsistencies is that it is assumed that an inconsistency precludes anything in the language from being true, in other words, an inconsistency would render the language useless as nothing identifiably true can be asserted in it.

The obvious alternative therefore is to adopt triviality in the sense that everything is provable, since a language in which this is the case is completely useless. One cannot use it to say anything as everything is equally true and false, and everything entails everything else. Nothing identifiably true is assertible in such a language. There may be other alternatives where there is a language in which not everything is provable, but it is equally useless. For example, it may be possible to have a language in which the negation of every sentence is provable, but some sentences are not provable. This would be equally useless, if such a language exists. However, as these are only possibilities, they will be ignored.

Adopting this definition of a Logical Language being affected gives rise immediately to the first condition on a Logical Language being affected by the Liar Paradox. A Logical Language is only affected if the *ex contradictio quodlibet* principle holds for the logical part of the language. That is, only Logical Languages in which a contradiction implies everything are affected in the full sense being adopted here.

Secondly, while it is possible that a Grammar-Only language which does not have a negation or a concept that directly expresses truth can be affected by the Liar Paradox in the relevant sense, this does not hold for Logical Languages. The pattern of argument that is distinctive of the Liar Paradox begins with the assumption that a certain sentence is true (or not true) and concludes that it is not true (or true). This argument can only be expressed in a language which includes both a negation and the concept of truth. These comprise a second condition on a Logical Language being affected by the Liar Paradox.

However, simply possessing a way of representing the concept of truth within the language is not sufficient to generate the paradoxical reasoning. A key assumption in the argument is that a particular sentence is true. If this assumption does not make sense, then the argument falls over. Thus it must be the case that the concept of truth applies to the grammatical sentences in the language in the natural way: they are either true or not true. While strictly speaking, the paradox can arise if only one paradoxical sentence must be either true or not true, this would be a very arbitrary language. We will adopt the simpler condition that the sentences in the language are either true or not true, and importantly, the method of expressing the concept of true captures these accurately.

It was argued above that a distinctive feature of the Liar Paradox is that the relevant derivations proceed without needing any auxiliary hypotheses or information beyond the meanings and truth conditions of the relevant sentences. We see this clearly in the typical argument that follows from a sentence such as:

(1) Sentence (1) is not true.

For this sentence, the argument can be broken down as follows:

> If sentence (1) is true, then it follows that what it says must be true. *(Condition on Truth as a Declarative Sentence)* What it says is that it, itself is not true. *(Semantic Content)* Therefore, if (1) is true, it follows that (1) is not true. *(Derivation)* However, suppose we assume the other alternative, that sentence (1) is not true. Then what sentence (1) says is in fact true (*Semantic Content)*, from which it follows that (1) is true *(Condition on Truth as a Declarative Sentence)*. Therefore, if (1) is not true, then (1) must be true. *(Derivation)*

Without going into the details, this pattern can be seen in any other paradoxical argument that arises from the Liar Paradox. However, this fact that no auxiliary information is required imposes a clear, and intuitive, restraint on the truth conditions for sentences. If no auxiliary information is allowed, then the truth conditions on a sentence cannot rely on anything beyond the semantic content of that sentence. In other words, a sentence is true if, and only if, its semantic content holds. This is another way of expressing the T-Schema: ‘P’ is true iff P. Thus, for the Liar Paradox to bite, it must be the case that the T-Schema holds for the language. It is not necessary that the T-Schema holds within the language, but only that from the sentence “P is true”, it is possible to derive P, and vice versa. This imposes a fourth necessary condition on a Logical Language being affected by the Liar Paradox, the T-Schema holds for that language.

Once we have established that a language allows the derivation from “P is true” to P and vice versa (and the contrapositives) for all sentences, and that the language includes a negation and the concept of truth, the standard pattern of derivation will arise for relevant cases. That is, from the assumption of the truth of a sentence (or equivalent) it will follow that the sentence is not true; and vice versa. However, this is not sufficient to establish that the logical language is affected in the strong sense, as this pattern by itself does not give rise to a contradiction. We require sufficient reasoning ability to turn this pattern into an explicit contradiction, so that *ex contradictio quodlibet* applies. The required reasoning is very minimal, and is not unique. Various combinations of principles of reasoning or normal assumptions will be sufficient. We will provide only one example using a Natural Deduction framework to illustrate. However, there are a range of different possible combinations of principles that give an equivalent result.

The situation is that we have established the following two patterns of argument:

	I			II	
1	$Tr\ulcorner P\urcorner$	Assumption	1	$\neg Tr\ulcorner P\urcorner$	Assumption
2	$\vdots$		2	$\vdots$	
3	$\neg Tr\ulcorner P\urcorner$	Conclusion	3	$Tr\ulcorner P\urcorner$	Conclusion

For example, if we assume further that the relevant language includes Reduction Ad Absurdem and $\wedge$-Introduction (using Natural Deduction), then we get the following:

$$\begin{array}{lll}
1 & \vert\ \underline{Tr\ulcorner P\urcorner} & \text{Assumption} \\
2 & \vert\ \vdots & \\
3 & \vert\ \neg Tr\ulcorner P\urcorner & \\
4 & \vert\ Tr\ulcorner P\urcorner \wedge \neg Tr\ulcorner P\urcorner & \wedge\ \text{Intro} \\
5 & \neg Tr\ulcorner P\urcorner & \text{RAA ln. 1,3} \\
6 & \vdots & \\
7 & Tr\ulcorner P\urcorner & \text{See above} \\
8 & Tr\ulcorner P\urcorner \wedge \neg Tr\ulcorner P\urcorner & \wedge\ \text{Intro}
\end{array}$$

We have derived a contradiction from the inclusion of two generally uncontroversial premises.

In general, any combination of principles which allow the relevant patterns of derivation to be combined to produce a contradiction are sufficient to mean that a relevant Logical Language is affected by the Liar Paradox.

We have identified five conditions that a Logical Language must satisfy if it is to be affected in the strong sense by the Liar Paradox:

1. *ex contradictio quodlibet* holds.

2. The Logical Language includes the concept of truth and a negation.

3. Sentences in the language are either true or not true (and the concept of truth in the language captures this).[4]

4. The T-Schema holds for the language.[5]

5. The language includes sufficient reasoning to derive a contradiction from the relevant pattern.

It follows that in any Logical Language whose Grammar-Only part satisfies the conditions in the previous section, and whose Logical part satisfies these, everything is provable. However, any language for which one of these does not hold is not trivialised in the same sense. We have, therefore, a set of conditions which differentiate languages which are affected from languages which are not affected.

1.4 What is the Liar Paradox?

We have identified a set of nine conditions which are jointly necessary and sufficient to distinguish languages which are affected by the Liar Paradox from languages which are not. As argued, this means that any Logical Languages (or logico-linguistic understandings of languages) which satisfy these nine conditions for both the Grammar-Only and Logical Languages above are necessarily trivialised. The Liar Paradox can simply be taken to express this fact, that languages which satisfy these conditions are trivialised, and therefore useless; and that languages that do not satisfy these nine conditions are not necessarily useless. In this sense, the Liar Paradox is an interesting observation about different types of languages.

This interesting observation has significant consequences for our understanding of natural languages and the concept of truth in natural languages. As argued above, natural languages satisfy the four conditions on Grammar-Only languages being affected by the Liar Paradox. Given that any understanding of truth and reasoning in a natural language can be represented as a Logical Language, it follows that any such understanding that satisfies the second set of conditions will be

[4] Strictly speaking, for the paradox to bite it is only necessary that one paradoxical sentence is true or not true, but this is highly artificial.

[5] Similarly, it is only necessary that the T-Schema to hold for one paradoxical sentence for the paradox to bite, but this is again highly artificial.

inconsistent and useless. However, as pointed out previously, an intuitive understanding satisfies these principles.

The problem that the Liar Paradox produces is therefore that a correct understanding of truth and reasoning must abandon at least one of these conditions, however it is not at all clear which one can be given up. The next chapter will examine the approach that various existing solutions take with respect to these criteria.

Chapter 2

Solutions to the Liar Paradox

The previous chapter developed a precise understanding of the Liar Paradox, by identifying conditions under which it is possible. It was argued that there is a list of nine conditions which distinguish languages which are affected by the Liar Paradox from those languages which are not. Four of these conditions relate to the vocabulary and grammatical construction of the language and hence are conditions on Grammar-Only languages; the other five are conditions on Logical languages as they relate to the concepts and reasoning present within the language. Any logical language which satisfies all nine conditions is trivialised by the Liar Paradox as everything will be provable in that language.

As noted before, we can understand the Liar Paradox as the fact that any language which satisfies all nine conditions is trivial, and therefore cannot be used for anything we would want to use a language for. The formal and philosophical problems that the Liar Paradox poses are that natural languages, combined with intuitive reasoning, and some formal languages, satisfy all nine conditions. The significant problems that this paradox poses requires a solution, and this chapter will focus on solutions to the Liar Paradox, both the question of what makes a good solution and some already existing solutions.

The understanding of the Liar Paradox developed in the previous chapter makes the possible approaches to solving the Liar Paradox clear, but it does not offer anything that would help identify the correct approach. If a language does not satisfy one of the nine conditions, that language will not be affected by the Liar Paradox, and therefore will not be trivialised. In this fairly formal sense, solving the paradox is relatively easy: one simply needs to provide a language, or understanding of a language, which does not satisfy one of the conditions. While this would be a solution in the formal sense, it may not be a solution in the sense of resolving the philosophical problems that arise, and which have prompted investigation into the Liar Paradox in the first place.

The philosophical issue is that the nine conditions are not all equal philosophically, while it may be possible that each could be got around, the cost of doing so varies. For example, one could decide that the solution to the Liar Paradox could be got by removing talk of "untruth" from a language. This would prevent the Paradox, but at the cost of making the language unsuitable for many purposes, especially philosophical discussion. The difficulty is that the Liar Paradox itself does not provide any reliable guide to a satisfactory philosophical solution. The Paradox is simply a fact about the consequences of certain conditions on languages, and while these conditions demonstrate the areas that need to be considered for a solution, they do not provide any guide to the solution in themselves.

Nevertheless, the philosophical problem itself does provide some guide to the scope of possible solutions. The philosophical problem is that natural languages satisfy the four conditions on Grammar-Only languages being affected by the Liar Paradox; and our intuitive understanding of reasoning and truth satisfy the five conditions on a Logical Language being affected. So the combination of natural language and intuitive reasoning about it is trivialised and therefore unusable.

If a proposed solution relies one of the four conditions on Grammar-Only languages, then to adopt it we must alter the way we use natural languages and their grammatical rules to take this into account. Such a solution would however be unsatisfactory for at least two reasons. Firstly, it would seem heavy handed to mandate different grammatical rules for natural languages on account of a logical paradox. This is not how natural languages are used or develop and is unlikely to be heeded in general. Secondly, we use and reason with sentences that require the offending conditions regularly and consistently, without having to be careful about how we use them. Either we are mistaken about our own use, or it is possible to use natural languages consistently and the Liar Paradox does not actually affect the logical structure behind natural languages. We should obviously only concede the first of these options if there is absolutely no other option, as it calls into question a vast amount of human endeavour and thinking. A satisfactory philosophical solution should leave the first four conditions alone and find a solution in one of the remaining five conditions on logical languages.

We will therefore be looking for a solution which provides a logical language which satisfies all four conditions on a Grammar-Only language being affected, and does not satisfy at least one of the conditions on a logical language being affected. However, we can be even more specific, as natural languages definitely satisfy one of the conditions on a logical language. The relevant condition is that the Logical Language "includes the concept of truth and a negation", which natural languages satisfy. This leaves four conditions in play in the search for a satisfactory philosophical solution, as one can provide a case that each of the other conditions can be given up, although the strength of the cases varies.

The fact that cases can be made for all of these conditions demonstrates that it is not possible to provide a definitive unchallengeable solution to the Liar Paradox, as the solution must draw on factors outside the paradox. We could only claim that there was such a solution if we had complete, universally accepted philosophy of language. We cannot claim to have or to be proposing such a complete solution. Nevertheless, we can see the Liar Paradox as a useful tool in developing a broader philosophy of language, as it can be used to rule out paradoxical philosophical positions. The focus of this work is however elsewhere. We will be concentrating on developing a formal logical language that provides a philosophically satisfactory formal truth definition.

2.0.1 What conditions are in play?

The most basic condition on a solution being philosophically satisfactory, as discussed, is that a logical language which embodies the solution satisfies the conditions which any natural language must satisfy out of the nine conditions identified in the previous chapter. For reference, these nine conditions are:

1. The language has a way of expressing the concept of untruth.
2. This concept can be predicated of/applied to the types of things which are used to express facts/information.
3. The types of things which are used to express facts can refer, by whatever means, to other things of this type.
4. There are no restrictions on the way that the mechanisms in the first three points can be combined.
5. *ex contradictio quodlibet* holds.
6. The Logical Language includes the concept of truth and a negation.
7. Sentences in the language are either true or not true (and the concept of truth in the language captures this).
8. The T-Schema holds for the language.
9. The language includes sufficient reasoning to derive a contradiction from the relevant pattern.

The first four of these conditions, which the conditions on Grammar-Only languages being affected by the Liar Paradox, are all satisfied by natural languages, as they are conditions on the way that sentences can be constructed and the vocabulary of the language. So any satisfactory solutions must satisfy all of these. As argued above, any reasonable understanding of natural languages must also satisfy condition 6. Truth and negation exist in natural languages, and therefore any understanding of natural languages must include these.

This leaves four conditions which are in play in terms of a satisfactory solution to the Liar Paradox, namely:

- *ex contradictio quodlibet* holds.
- Sentences in the language are either true or not true (and the concept of truth in the language captures this).
- The T-Schema holds for the language.
- The language includes sufficient reasoning to derive a contradiction from the relevant pattern.

There are strong arguments that each of these hold, yet it is also possible to form arguments against each of these. For the moment, we will not look in depth at the reasons for holding or abandoning any of these, as this will be done in the context of solutions to the Liar Paradox in the literature. Before we look at these, however, it is important to clarify what exactly is meant by a *solution* to the Liar Paradox.

2.1 What is a Solution to the Liar Paradox?

Within the literature on the Liar Paradox, there is not only disagreement about the correct solution to the paradox, but there seems to be a great diversity of opinion as to what constitutes a solution. At one end, some authors offer a diagnosis of the incorrect assumption and demonstrate how the paradoxical reasoning in a natural language context is blocked for some examples when these are corrected. At the other end, others simply present a technical definition of a formal language in which there is a truth predicate and the paradox does not arise. These are very different projects, and neither captures fully what an adequate solution to the Liar Paradox is.

The first approach recognises that the problem that needs to be solved is primarily a question about the coherence of our use of reasoning and natural languages. Solving this requires a diagnosis of the problematic assumptions so that we can learn to reason correctly with natural languages without fear of incoherence. However, simply showing that the normal paradoxical reasoning does not arise in a small number of cases does not demonstrate that there is no incoherence within natural language and reasoning. It may have been simply moved to a different type of argument, or arise with a different type of case. Correcting the problematic assumption also requires making claims about the philosophy of language more broadly which are not often explored in this context.

Simply providing a formal solution, however, does not guarantee any solution of the overriding philosophical issues. One can construct many consistent formal systems which include a truth predicate, but the important question is whether these formal systems are appropriate as a way of understanding the logical structure of natural languages. To use the point made above, the formal systems must satisfy all of the relevant conditions that natural languages satisfy, and the mechanism that prevents the paradox in the formal system must be plausible in a natural language context. However a full formal solution allows us to decide whether the paradox has been systematically prevented, as we have techniques we can investigate the consistency of the system with.

So a comprehensive solution must combine elements of both of these approaches, offering both a natural language justification, or at least motivation, for the problems identified plus a formal definition that can show that the result truly is not affected by the paradox. The investigation of different solutions below will focus on solutions offered in the literature which are comprehensive in this sense. However, before examining these, it is worth looking further at the cases that need to be discussed. While the previous chapter discussed some more complicated examples than the most simple cases, there are further cases which are relevant to the discussion.

2.2 Cases of the Liar Paradox

We will continue to treat the following as the stereotypical case of the Liar Paradox:

1. This sentence is not true.

In the previous chapter, the following cases were also introduced:

2. What this sentence says is not the case.

3. The following sentence is true.

4. The previous sentence is not true.

However, there are many more, less artificial, examples of Liar Sentences which need to be taken into account. For example, the following case is also paradoxical:

- I am currently lying.

- Policeman (in court): "Most of what the defendant says in the witness stand will not be true."
  Defendant (says only this): "What the policeman said was true."[1]

Each of these cases introduce further complexities into the analysis of the Liar Paradox, which will become clear in the discussion of the different solutions. However, one remark can be made from a survey of the cases here. We can see from the range of sentences in the different cases here that there are no common grammatical properties shared by all of these sentences. This further strengthens the point made above that we cannot look for a solution amongst the grammatical conditions. It is not possible to find any grammatical properties that are unique to paradoxical sentences. Any satisfactory solution must look at the four logical conditions identified above.

[1] This example is adapted from one given in L. Jonathon Cohen. "Can the Logic of Indirect Discourse be Formalised?" In: *The Journal of Symbolic Logic* 22.3 (1957), pp. 225–232

2.3 Solutions in the Literature

This thesis will briefly examine seven different solutions in the literature to understand how they deal with the challenge of the Liar Paradox. The solutions chosen for investigation are a small selection of the different approaches in the literature, but they all have a well-worked through formal account to accompany the philosophical discussion. The analyses of the different solutions rely on the framework that has been developed in the previous chapter and the Introduction, as this will help highlight the exact nature of the solution. In particular, we will focus on which of the nine conditions is not met in order for the solutions to work. Of particular interest are common elements to the different approaches and similar problems that arise across multiple solutions.

2.3.1 Alfred Tarski

Alfred Tarski published the first formal truth definition in his seminal paper "Der Wahrheitsbegriff in den formalisierten Sprachen"[2] In doing so, Tarski established the modern for of this area of research and has profoundly affected the scope of subsequent research into formal truth predicates and the Liar Paradox. Tarski's solution is therefore the natural place to start any investigation into existing solutions to the Liar Paradox in the modern literature.

The clearest indication of the importance of Tarski's work is that he introduced two ideas which are now central a lot of work in modern logic and philosophy. The first idea is the concept of a formal meta-language in which concepts can be defined that cannot be defined within the normal (object) language. The second idea is the T-Schema, namely the schema that " 'P' is true if, and only if, P". Both of these concepts play a crucial role in Tarski's solution to the Liar Paradox and are indispensable elements of modern philosophy and logic. However, before turning to Tarski's formal solution, it is worth looking at the philosophical motivation and justification.

Central to Tarski's approach is his conviction that

> In [colloquial] language it seems to be impossible to define the notion of truth or even to use this notion in a consistent manner and in agreement with the laws of logic.[3]

This disagrees profoundly with most modern approaches, including the approach to the Liar Paradox that has been offered in this thesis. Admittedly, in line with Tarski it has been argued in this thesis that, if correct, the Liar Paradox demonstrates that our use of the concept of truth in natural languages is necessarily inconsistent. However, while Tarski seemed to take this as evidence of the flaws in natural languages, we have taken it as evidence of flaws in our understanding of natural languages. Tarski's response to the problems he saw with natural languages was to formulate a precise, logical, scientific language in which the concept of truth can be consistently and accurately

[2] Alfred Tarski. "Der Wahrheitsbegriff in den formalisierten Sprachen". In: *Alfred Tarski: Collected Papers*. Vol. 2. Birhäuser, 1986. The English translation was published in Alfred Tarski. "The Concept of Truth in Formalized Languages". In: *Logic, Semantics and Metamathematics*. Oxford University Press, 1956, pp. 152–278. All quotes are taken from the English translation.

[3] Tarski, "The Concept of Truth in Formalized Languages", see n. 24, p. 153

used.[4] The idea was to define a replacement for natural languages. The standard modern approach is to reform our understanding of truth and/or reasoning in natural languages to legitimise our use of natural language, particularly as it seems impossible for us to ever completely replace natural language talk of truth with a formal language.

Tarski's argument for the inconsistency of natural languages begins with a justification of the T-Schema as a principle which any truth predicate must satisfy. Natural languages, when treated as a Logical Language, automatically satisfy conditions 1, 2, 3, 4 and 6, and classical logic, which Tarski accepts, satisfies conditions 5 and 9. As Tarski is a classical logician he seems to accept the principle that every sentence is either true or false, and that a truth predicate should be able to report this, which means he accepts condition 7. Given that the last principle is the T-schema, it is unsurprising that Tarski saw natural languages as necessarily inconsistent, as combined with his account of reasoning and truth all nine conditions are satisfied and hence natural languages are necessarily inconsistent.

Tarski's specific argument for the inconsistency of natural languages turns on whether it is possible to provide a rule based method of naming natural language sentences that allows the T-schema to be satisfied. As it seems to be impossible, Tarski takes it as evidence that natural languages are inconsistent.[5] This shows that Tarski did not consider any of his logical assumptions to be contestable, and thus was focusing effectively on the conditions on Grammar-Only languages. It also highlights a distinction between natural languages and formal languages which becomes relevant later in the thesis. Where a formal language includes names as a means to refer to sentences, it does so in a precise, regimented way which normally pairs one sentence with one name. Natural languages are, on the other hand, very liberal with names and techniques for referring to other sentences. There are a wide variety of ways we can refer to a sentence, including demonstratives, definite descriptions and quotation marks, and it is possible to define new ways within natural languages. This large difference raises the question of whether an accurate account of natural languages can be fitted into the standard naming structure of a formal language.

There is another simple argument for natural languages being inconsistent that Tarski does not use, but which is useful to highlight another relevant distinction between formal and natural languages. It is obviously the case that the way that people use natural languages is often inconsistent. People often contradict themselves and others with their assertions. However, this is a different type of inconsistency to the type the Liar Paradox presents. While people often use natural languages inconsistently, it is still possible that they can be used consistently, if we are careful with what we assert. If correct, the problem posed by the Liar Paradox demonstrates that it is not in fact possible to use natural languages consistently.

This distinction between how natural languages are actually used and how it is possible to

[4] Tarski, "The Concept of Truth in Formalized Languages", see n. 24, p. 165

[5] See Tarski, "The Concept of Truth in Formalized Languages", see n. 24, pp. 154-164

use them does not exist for formal languages. The rules governing a formal language only allow provable sentences to be stated or asserted within the language. This means that if a person spoke a formal language correctly, they could only assert true sentences. It is not possible to assert a falsehood in a formal language without breaking a rule of that language. Natural languages do not have a similar distinction between legitimate and illegitimate assertions as part of the rules of the language. For example, there is no distinction between "Snow is white" and "Snow is black" as legitimate English sentences. While there are undoubtedly rules about what is and is not assertible when we use natural languages, these rules are provided by things like context and cultural expectations, not by the language itself. To put it differently, one can (and many do) assert lies in a natural language without violating any rules of the language.

Tarski's approach to the Liar Paradox was to see problems with the construction of natural languages, and therefore seek to replace them, at least for talk about truth. This is however not a realistic option, as it is not possible to transcend our use of natural languages. For example, suppose we wanted to justify moving to a particular formal language, the justification for this must occur in a natural language otherwise we could not understand the point of changing. However, if natural languages are necessarily inconsistent, then we cannot be sure that our justification is valid, and therefore that we are justified in moving to the formal language. While it is argued that a more valuable approach to resolving the problems with natural languages is to try to ascertain whether it is ever possible to use natural languages consistently, it is important look at the way Tarski constructs a formal truth definition, both due to its influence and that it highlights common issues.

The Formal Solution

Given that Tarski rejects the suitability of natural languages for a truth definition, it is unsurprising that he focuses his solution on the first four conditions in our list, the conditions on a Grammar-Only language being affected by the Liar Paradox. The key structural component to Tarski's solution that allows him to avoid paradox is his distinction between an object language and a metalanguage. For Tarski, "we must always distinguish clearly between the language *about* which we speak and the language *in* which we speak."[6] While we would normally talk *about* objects *in* the object language, we can only talk *about* an object language *in* a metalanguage. In the formal definition, Tarski only allows the definition of a truth predicate for sentences in the object language within the metalanguage. Thus, for a sentence P in the object language, $P\, is\, true$ is only defined in the metalanguage. When we look at the nine conditions above, it follows that the object language does not satisfy conditions 1, 2 and 6. That is, the object language does not contain the concepts of truth or untruth and therefore these concepts cannot be predicated of things within

[6] Tarski, "The Concept of Truth in Formalized Languages", see n. 24, p. 167

the language.

This prevents the paradox from occurring, since the type of circular reference which is central to the paradox cannot arise, as the truth predicate is only defined in the metalanguage to refer to sentences in the object language. Importantly, the truth predicate in the metalanguage is restricted so that it only applies to sentences in the object language, not to other sentences in the metalanguage. This means that the metalanguage does not satisfy condition 4, since the truth predicate cannot apply to sentences of the metalanguage - there is a restriction on the what the truth predicate can be predicated of. If we were to want a truth predicate for sentences in the metalanguage, we would have to treat the metalanguage as an object language and move to a meta-metalanguage. However, Tarski does not consider this as it is not necessary for his project of replacing natural languages with a precise formal language for truth. The ability to deal with sentences such as ' "P is true" is true' is relatively unimportant as these types of sentences are not normally necessary in a formal scientific language.

While the distinction between object language and metalanguage provides a method for preventing the Liar Paradox, it does not provide any definition of truth. There are two other key components to Tarski's formal definition which deal with the actual definition of a truth predicate. The first is the T-Schema, and the second is his recursive definition of the truth predicate.

The T-Schema, that *'P' is true iff P*, has been widely adopted as a definition of truth, however Tarski uses it in a very particular way. He introduces the T-Schema as the key point in his Convention T which provides the necessary conditions which any satisfactory truth definition must satisfy.[7] That is, for Tarski, the T-Schema provides a test which must be true of any satisfactory definition of truth. If it does not follow from a truth definition that say '"Snow is white" is true iff snow is white', then for Tarski that truth definition cannot be correct. The T-Schema does not define anything in Tarski's definition, and he takes care to show that his truth definition satisfies the T-Schema.

Tarski further provides a formal definition for a particular mathematical theory. This has advantages of style in that Tarski can assume that his formal theory includes a large amount of mathematical reasoning which is rarely included within modern formal languages due to the complexity of some of the concepts. Most relevantly, this approach allows Tarski to use mathematical concepts to offer a recursive definition of the truth predicate within the metalanguage. Thus Tarski's actual definition of the truth predicate is by recursion over satisfiability. The details are unimportant to this discussion, however they are very familiar to any modern logician, as Tarski uses the same technique as is now used for defining truth values in a model.[8]

[7] Tarski, "The Concept of Truth in Formalized Languages", see n. 24, pp. 187-88

[8] See Tarski, "The Concept of Truth in Formalized Languages", see n. 24, p. 193

Analysis

Tarski's formal definition is simple and formally elegant, and can be easily shown to be consistent. However, it has well-known problems as a philosophical account. The most famous problem is the critique offered separately by both L. Jonathon Cohen[9] and Saul Kripke[10]. The critique turns on situations that arise in natural languages where perfectly ordinary sentences can turn out to be paradoxical. These contingently paradoxical cases normally involve two ordinary assertions which, in the circumstances, turn out to set up a two sentence version of the Liar paradox. The example give above is a classic example of this phenomenon:

- Policeman (in court): "Most of what the defendant says in the witness stand will not be true."
- Defendant (says only this): "What the policeman said was true."[11]

The problem is that this type of situation cannot even be set up in a Tarskian situation as both sentences would need to be in a metalanguage with respect to the other. As an account of truth in natural languages, the Tarskian approach is limited as it cannot even formulate this legitimate pattern of natural language reference.

This point can perhaps be made even more clearly if we consider a non-paradoxical case which is nevertheless cannot be formulated in a Tarskian system. A simple case is a case where two friends each assert of the other that "The vast majority of what he says is true". Both assertions can be true, yet each sentence is in the scope of the quantification of the other. In a Tarskian system each must be in a metalanguage with reference to the other, which is not possible. These problems make it clear a Tarskian system cannot provide a satisfactory philosophical analysis of natural languages. This fact, however, would not surprise Tarski, as he did not believe that a satisfactory philosophical analysis of natural languages is possible. His aim was to replace natural language talk of truth, rather than to explain it.

This critique of the Tarskian approach provides a neat practical example of the general point that has been made several times in our analysis of the Liar paradox. Natural languages satisfy all of the conditions on a Grammar-Only language, while the Tarskian approach does not satisfy these conditions. The general point that has been made is that this means the Tarskian solution cannot provide an account of truth and reasoning in natural languages, and the critique above demonstrates one reason why this is general point holds. While Tarski offers a very elegant formal solution, it does not do anything to help resolve the primary philosophical problem that has been identified, as it cannot be applied as an account of truth and reasoning in natural languages.

[9] Cohen, "Can the Logic of Indirect Discourse be Formalised?", see n. 10, p. 226

[10] Kripke, see n. 9, p. 692

[11] This example is adapted from one given in Cohen, "Can the Logic of Indirect Discourse be Formalised?", see n. 10

2.3.2 Kripke

The paper by Saul Kripke, "Outline of a Theory of Truth"[12], contains a formal truth definition which explicitly aims to fix the some of the limitations inherent in the Tarskian approach. In particular, Kripke aimed to avoid the obvious limitation in Tarski's system, from a natural language perspective, that a truth predicate cannot be defined that applies to its own language. The distinction between object language and metalanguage in Tarski's approach explicitly prevented a truth predicate applying in its own language. Kripke's definition is explicitly an attempt to provide a definition of a truth predicate that does not have this limitation.

Kripke bases his critique of Tarski, and his analysis of the problem posed by the Liar paradox on the cases of contingent liar sentences, such as the example above. The key lesson that Kripke drew from these examples was that "it would be fruitless to look for an intrinsic criterion that will enable us to sieve out— as meaningless, or ill-formed – those sentences which lead to paradox."[13] It follows that if there is no intrinsic criterion, nothing in the sentences themselves, then there must be some external criterion we can use. Kripke identifies an external criterion in the following way:

> If ... sentences themselves involve the notion of truth, their truth value in turn must be ascertained by looking at *other* sentences, and so on. If ultimately this process terminates in sentences not mentioning the concept of truth, so that the truth value of the original statement can be ascertained, we call the original sentence *grounded*; otherwise, ungrounded.[14]

This external criterion of groundedness gave Kripke an intuitively compelling way of separating paradoxical from unparadoxical sentences, and it lent itself very easily to a formal recursive definition.

In a brief summary, Kripke's formal approach was to extend the recursive definition of truth values of sentences to include the truth predicate. Kripke's definition starts with a base set of sentences closed under the standard recursive definitions of connectives that is taken to include all true sentences which do not include a truth predicate (the facts). The truth predicate is then defined recursively on top of this set by forming a hierarchy of new sets which include "P is true" for every sentence P in the previous set. If this recursive definition is continued transfinitely, then there are guaranteed to be fixed points where the sentences in the set stabilises. Kripke took the fixed points to be complete sets of true sentences.

This formal definition captures the concept of groundedness in a natural way as only grounded sentences will ever be included in the set hierarchy. Any sentence in the hierarchy will either

[12] Saul Kripke. "Outline of a Theory of Truth". In: *Journal of Philosophy* 72.6 (1975), pp. 690–716

[13] Kripke, see n. 9, p. 692

[14] Kripke, see n. 9, p. 693-4. This concept was previously introduced in Hans G. Herzberger. "Paradoxes of Grounding in Semantics". In: *The Journal of Philosophy* 67.6 (1970), pp. 145–167

be in the base or refer to a sentence (or sentences) at a lower level in the hierarchy. This lower sentence (or sentences) will in turn either be in the base or refer to another lower sentence (or sentences). For all sentences in the hierarchy if this process is continued, they will eventually only refer to sentences in the base set of facts. In other words, every sentence in the hierarchy is grounded. Moreover, paradoxical sentences will never be included in the hierarchy as, as argued in the previous chapter, they only occur when a sentence belongs to a circular reference chain. The recursive definition of the hierarchy guarantees that no circular reference chains will be included. It follows that paradoxical sentences are never true, but are never false either. In this way Kripke formulates a consistent definition of truth in which the truth predicate can apply to sentences within its own language.

Kripke therefore offers a formal definition which is a natural extension of existing techniques in logic, and addresses one of the main concerns with Tarski's work, however there are a number of limitations to it. One significant limitation was noticed by Kripke himself, as he noted that "There are assertions that we can make about the object language which we cannot make in the object language."[15] The example Kripke gave was the sentence 'The Liar Sentence is not true'. This sentence is intuitively true since the Liar Sentence is never included in the set hierarchy of true sentences. That is, it is a consequence of Kripke's solution that Liar Sentences are not true, but while a sentence which expresses this would be grammatically acceptable in the object language, it has no truth value. This means that the object language in Kripke's solution does not satisfy condition 7, since there are sentences which cannot be said to be either true or not true. Interestingly, Kripke's solution also does not satisfy condition 8, since the T-Schema is not in general valid in the object language. It is valid for all grounded sentences, but the T-Schema for ungrounded sentences does not have a truth value.

While there is some intuitive support for the idea that a paradoxical sentence might not be the sort of thing that is truth-apt, the exact shape of the problem for Kripke is more serious. One of the key theses in Kripke's solution is that Liar Sentences are not true since they are not grounded. However, for any sentence P, if P is not grounded then "P is not true" is also not grounded. Hence for any given Liar Sentence, L, the sentence "L is not true" is not grounded and therefore not true. Thus it is not possible to assert a key thesis of Kripke's truth definition within the object language, as it is not true in the object language. As Kripke notes, to assert this key thesis it is necessary to move to a meta-language, and thus the ghost of the Tarskian hierarchy remains.[16]

Ironically, Kripke's core critique of Tarski's solution was that invoking a meta/object language distinction cannot work when we are analysing natural languages. Since Kripke's solution ultimately relies on a metalanguage to assert the relevant theses, it follows that Kripke's own solution has a similar natural limitation in its application to natural languages: it cannot deal with dis-

[15] Kripke, see n. 9, p. 714

[16] Kripke, see n. 9, p. 714

cussion of the Liar Paradox, since key theses of his solution are not true without resorting to a metalanguage.

There is another aspect to Kripke's proposed solution which limits its usefulness as a philosophical account. Kripke's approach in using his recursive definition is to construct a semantic model of a consistent truth definition within a language. The existence of a sound model demonstrates that it is possible to have a consistent truth definition within a formal language. However, Kripke does not himself offer any definition within the object language, whether by means of axioms or rules of inference. The limitations with this are clear when we consider natural languages. Kripke, if correct, has effectively shown that it is possible to have a consistent truth predicate within natural languages, however, does not give any clue as to how this truth predicate can be defined or even understood within a natural language.

Thus Kripke provides the definition of a semantic model for a consistent truth definition in which the truth sentence can apply to sentences in its own language. As such it is an improvement on Tarski as a philosophical analysis of natural languages, in particular as it does not impose any restrictions on the construction of sentences in the language. His definition does not satisfy conditions 7 and 8, and has the significant limitation that key results of the definition are not true within the relevant language. This limits its applicability as a philosophical account of natural languages, as there is no metalanguage that transcends natural languages.

2.3.3 Dialetheism

Dialetheism, the position that accepts the existence of true contradictions, is an increasingly popular response to the Liar Paradox. It has been most vigorously been advanced by Graham Priest, who is also one of the original proponents of the position.[17] Essentially, this approach argues that all of the premises in the paradoxical argument are correct, and we should therefore accept the conclusion – a contradiction – as correct. That is, for the dialetheist, any Liar Sentence is both true and false.

Judged against the nine conditions outlined previously, dialetheism is a very neat solution, as it simply requires rejecting condition 5. That is, it is simply necessary to develop a logic in which a contradiction does not prove everything, and the T-Schema can be used as the definition of truth. While to a classically trained philosopher or logician this seems bizarre, there are well worked through formal logics with the required properties, including a sound model for the logic. The most famous of these logics is Priest's LP, presented in his paper "The Logic of Paradox."[18] Thus essentially the dialetheist offers a formally simple solution in which all of our ordinary intuitions

[17] See, among others Graham Priest. "Logic of Paradox Revisited". In: *Journal of Philosophical Logic* 13 (1984), pp. 153–79; Graham Priest. *In Contradiction: A Study of the Transconsistent*. Dordrecht: Martinus Nijhoff Publishers, 1987; Graham Priest. "What is So Bad about Contradictions?" In: *The Journal of Philosophy* 95.8 (1998), pp. 410–426; Graham Priest. "Truth and Contradiction". In: *The Philosophical Quarterly* 50.200 (2000), pp. 305–319

[18] Graham Priest. "The Logic of Paradox". In: *The Journal of Philosophical Logic* 8 (1979), pp. 219–241

about truth hold, only that we must give up consistency as a prerequisite for truth.

Aside from the simplicity of the formal definition the dialetheist provides, one of the primary arguments in favour of dialetheism is the claim that it allows semantic closure. That is, all sentences within the logic have a truth value, and the assertion that these sentences have the particular truth value is a valid sentence within the language. This is a highly desirable property that we would want to hold for natural languages, if we want to be able to use natural languages in the way we normally do, particular for philosophical discussions.

The dialetheist position therefore has a number of significant advantages, which obviously must be weighed against the cost of accepting true contradictions. While there are various considerations about whether this is a reasonable approach to take, they are beyond the scope of this thesis. There is however a more specific weakness in Priest's approach in particular that has interesting parallels with a limitation in Kripke's solution.

One of the key theses of the dialetheist approach is that Liar Sentences are both true and false. However, if we go through the semantics of Priest's LP, we can see that not only are Liar Sentences both true and false, but sentences such as "The Liar Sentence is both true and false" are also both true and false.[19] While Priest does not see this as a problem, it is a weakness in the dialetheist approach. Liar Sentences are obviously problematic sentences which, according to Priest, arise at the limits of reason.[20] However, asserting that "The Liar Sentence is both true and false" is a substantive philosophical claim that dialetheism relies on. If this philosophical claim is not valid, then dialetheism cannot be correct. However, according to the dialetheist, this sentence has exactly the same truth status as obviously problematic Liar Sentences. Within dialetheism, a sentence asserting the key philosophical position of the theory is no more and no less successful as a claim about reality than a Liar Sentence.

This problem becomes more acute when we recognise some of the consequences of this position. If someone attempts to disprove dialetheism by proving that the claim "The Liar Sentence is both true and false" is false or even not true, the dialetheist can simply accept this as proving half of the thesis that this sentence is both true and false. That is, the core thesis in dialetheism is not falsifiable to the dialetheist – it cannot be disproven. Thus either the fact that the Liar Sentence is both true and false is a necessary logical truth, or it is a problematic in the same way that scientific theories which are not falsifiable are problematic.

The interesting thing about this weakness in the dialetheist position is that it mirrors the limitation in Kripke's solution. In both cases, sentences about a Liar Sentence have the same truth value as the Liar Sentence within the relevant language, even if we would interpret the sentence about the Liar Sentence as making a substantive philosophical claim.

[19] For example, see Priest, "The Logic of Paradox", see n. 3, pp. 238-9

[20] This is the theme ofGraham Priest. *Beyond the Limits of Thought*. Cambridge: Cambridge University Press, 1995

2.3.4 Barwise & Etchemendy:

Barwise and Etchemendy, in their work *The Liar: An Essay on Circularity and Truth*[21], introduce two ideas from the philosophy of language which they use to construct a coherent solution to the Liar Paradox. The first idea is the distinction between the assertion of a negation and a denial, while the second is the idea of situation semantics. While the first idea assists in their precise analyses of liar sentences, it is the second idea that plays the crucial role in preventing their solution becoming inconsistent and therefore incoherent. This analysis will focus on the second idea.

In Barwise and Etchemendy's approach, sentences are not taken to be true universally, but rather true in particular circumstances or situations. The broad outlines of this idea are familiar, however Barwise and Etchemendy implement this idea in a very precise way using a non-standard set theory developed by Peter Aczel. For Barwise and Etchemendy, the fact that a sentence is only true in a particular situation means that the truth of that sentence within that situation can only be asserted from outside that specific situation. We have to move to a broader situation of which the original is a part to be able to assert that the sentence is true in the original situation.

By imposing this requirement, Barwise and Etchemendy's approach prevents the paradox from biting since the conclusions in the argument relate to truth in a different situation to truth in the original sentence. While this hierarchical approach is reminiscent of Tarski's solution, it operates in a significantly different way since they do not impose grammatical restrictions on the valid sentences. Instead, the approach effectively uses a hierarchical approach to separate different components of the paradoxical argument to different levels in the hierarchy so the complete argument cannot run. This allows the truth predicate in a language to report the truth of other sentences in that language, if we take the truth predicate as indexed over situations. In this way it avoids many of the problems that Tarski faced.

On the list of nine conditions, Barwise and Etchemendy's approach seems to fail two and possibly more conditions. It fails condition 4, since there are restrictions on the valid combinations of sentences with the truth predicate as we need to index truth to situations. It also fails condition 7 since sentences in the language are not simply either true or not true as their truth is relative to particular situations. Many sentences do not have any truth value in many situations, as they are entirely irrelevant to the given situation. There is also an open question as to whether there is a univocal truth predicate in Barwise and Etchemendy's approach, or whether there are a large number of indexed truth predicates. If the second option is correct, then condition 6 also fails.

However, there is a big limitation with this approach since it is never possible to move to a universal situation. If we move to a universal situation, there is no broader situation to move to

[21] Jon Barwise and John Etchemendy. *The Liar: An Essay on Truth and Circularity.* New York: Oxford University Press, 1987

which means as the quarantining effect cannot occur and the Liar Paradox will arise again. However, the impossibility of moving to a universal situation in Barwise and Etchemendy's approach means that universal statements cannot be completely universal - they must be always interpreted as about a particular situation. This is in conflict with many of our deepest philosophical habits, and calls into question the coherence of any philosophical discussion of the Liar Paradox. If there is no universal situation, any conclusion we draw about the Liar Paradox cannot apply universally but only to the situation relevant to the discussion. This is presumably as broad as it is possible to make, but it cannot be universal. This means that, if correct, Barwise and Etchemendy's solution cannot hold universally but only within some sort of limited situation.

It seems difficult to accept that an approach which intends to completely resolve the Liar Paradox cannot allow that its core theses hold universally. While it does not necessarily follow that the theses are false in a broader situation than the relevant one, one can never know, as any claim about the Liar Paradox in all situations is not legitimate. Interestingly, this problem falls into a similar category to the problems with Kripke's and the dialetheist's approaches. In each case, key philosophical claims by the relevant theory about the Liar Paradox or liar sentences cannot have the intuitively correct truth status within the theory.

2.3.5 Gupta & Belnap:

Gupta and Belnap, in their book *The Revision Theory of Truth*,[22] offer a carefully worked through solution which is based on a theory of definitions and draws significant inspiration from Kripke's work. Central to this approach is an argument that circular definitions, such as in the case of the Liar Paradox, are often legitimate definitions even if they run into trouble in cases like the Liar Paradox. While the details of Gupta and Belnap's approach are beyond what can be covered here, their key strategy for avoiding the paradox is worth reviewing.

The key ideas in their approach are neatly summarised as follows:

> We recognize the Liar to be pathological, but we do not want to say that it is neither true nor false. The sentences 'the Liar is not true' and 'the Liar is not false', whether viewed as belonging to the object language or to some metalanguage are pathological in exactly the same way as the Liar is pathological. To correctly describe the semantic status of the Liar we need to appeal to notions such as "categoricalness".[23]

In terms of the conditions about, Gupta and Belnap thoroughly reject Condition 7, since they argue that there are sentences which are neither true nor not true, but are pathological. More could be said about what pathological means, but the basic idea is that something goes so wrong in the

[22] Anil Gupta and Nuel Belnap. *The Revision Theory of Truth*. The MIT Press, 1993

[23] Gupta and Belnap, see n. 22, p. 255. Categorical sentences for Gupta and Belnap are sentences which are not pathological.

definition of Liar Sentences that it is no longer for these sentences to be considered as truth-apt, they cannot have any truth status at all, not even a negative truth status.

There are a number of comments that can be made about an approach of this sort, including the fact that it seems to be depriving negation of its intuitive meaning. One would normally interpret the set of not true sentences as being the complement of the true sentences, however Gupta and Belnap posit a third set of pathological sentences. However, with regards to the philosophical strengths of this approach, there are two key aspects that are worth focusing on.

Firstly, Gupta and Belnap, in line with a number of more recent authors,[24] appeal to a concept other than truth to help make the necessary distinctions to prevent the Paradox. For Gupta and Belnap, the key concept is categoricalness since it is only categorical sentences which are truth-apt. However, as Gupta and Belnap admit, it is possible to use the concept of categoricalness to reformulate a strengthened liar paradox, which they cannot avoid.[25] This strengthened paradox does not mean, however, that their formal approach is inconsistent, as they rely explicitly on a distinction between an object language and a metalanguage.

While in Tarski's work the concept of truth was defined in the metalanguage for sentences within the object language, in Gupta and Belnap's approach the concept of truth is defined within the object language for sentences within the object language. However, the key concept in their approach of categoricalness is not defined within the object language but only within the metalanguage. The fact that this concept is defined in the metalanguage for sentences within the object language means that the strengthened paradox is avoided in exactly the same way that Tarski avoids the original paradox: the paradoxical cases cannot be formulated.

However, while the object language includes a consistent definition of truth for its own language, there are obvious limitations in apply this approach as a model for understanding reasoning and truth in natural languages. On Gupta and Belnap's approach, the coherence of the concept of truth in the object language depends crucially on a concept that cannot be formulated within the object language. If we apply this to natural languages, then it should follow that the coherence of the concept of truth in a natural language depends crucially on a concept that cannot be formulated within that language. However, Gupta and Belnap presented their arguments clearly in a natural language and defined the concept of categoricalness clearly in a natural language. This presumably means that categoricalness cannot be the relevant concept, if we wish to apply this approach to natural languages.

In fact, if we wish to apply this general style of approach to natural languages, we are left with a significant problem. This style of approach depends there existing some sort of concept that is not definable within the natural language but which underlies our concept of truth. However, it

[24] Such as Vann McGee inVann McGee. *Truth, Vagueness and Paradox: an essay on the logic of truth.* Indianapolis: Hackett Pub. Co, 1990; and Hartry Field, see below.

[25] Gupta and Belnap, see n. 22, pp. 255-6

is not clear that we can grasp a concept that cannot be formulated in a natural language, as we definitely cannot articulate it. If we cannot articulate or even grasp the concept, there is a big question as to how we can then know that this concept guarantees the coherence of our concept of truth. Yet again, like the previous solutions that have been investigated, these approaches rely on philosophical theses which cannot be true within the relevant theory or language.

2.3.6 Michael Glanzberg

The key idea to Michael Glanzberg's solution to the Liar Paradox is that the propositions that sentences which include the truth predicate express are context dependent, and that the relevant context changes through the paradoxical argument.[26] This means that when a liar sentence and its negation are both proven within a paradoxical argument, it is not the case that two contradictory propositions are expressed by the two sentences. There is therefore no contradiction and no paradox.

Glanzberg bases his analysis of the way that the context changes on an argument that what we assert something of the form 's is true', what we are saying is "something like 'the proposition expressed by sentence s is true', or more informally, 'what she said when he said s is true'."[27] This means that there is an implicit quantification and the sentence actually has the form: there is a proposition p such that s expresses the proposition p in context c and p is true. Glanzberg argues for a context change that results from the domain of quantification for the existential quantifier in the sentence changing as consequence of the argument that normally generates from the Liar Paradox.

Glanzberg acknowledges the similarities in his approach to Barwise and Etchemendy's.[28] Although his approach does not suffer the same limitation that Barwise and Etchemendy's does, there is a related and smaller limitation. The key fact about the context shift that Glanzberg relies on is that it occurs during the paradoxical argument. Thus while the Liar Sentence may not express a proposition at the start of the argument, and therefore cannot be true; by the end of the argument the domain of truth conditions has expanded so that there is a proposition to express, and what it expresses it true for the previous context.

Glanzberg's analysis, as for any solution to the Liar Paradox, is presumably to be taken as a universal conclusion about how to solve the Liar Paradox. If we have a universal conclusion about all instances of the Paradox, however, we know something about the truth status of any Liar Sentence without needing to go through the argument. However, if Glanzberg's analysis is correct, it is difficult to see how this is the case. To show this we will consider the following example:

W: W is not true

[26] For example, see Michael Glanzberg. "The Liar in Context". In: *Philosophical Studies* 103 (2001), pp. 217–251

[27] Glanzberg, "A Contextual-Hierarchical Approach to Truth and the Liar Paradox", see n. 5, p. 32

[28] See Glanzberg, "A Contextual-Hierarchical Approach to Truth and the Liar Paradox", see n. 5, pp. 38-40

We can immediately recognise W as a Liar Sentence. If we take the final conclusion of Glanzberg's approach, then we know that W will express a true proposition in the appropriate context. However, as we have not gone through the relevant reasoning, W does not express a proposition and is therefore not true. Thus it seems that we know that if we were to go through the relevant reasoning, W would express a true proposition, but because we are too lazy to, it does not express a proposition at all. The truth value of W therefore seems to depend on whether we have actually performed some act, which is counterintuitive given that the sentence W does not refer to or depend on any matter of fact. This leaves open the possibility that two perfectly rational agents can disagree about the truth value of W, and be both rationally justified in their belief.

If we look at this approach against the nine conditions above, Glanzberg's approach does not satisfy condition 6, since there is not a single concept of truth included in the language. The relevant concept depends on the relevant context. Glanzberg's approach also mirrors Kripke's approach in that he offers a complete semantics for his language, without attempting to provide a definition of truth internal to the language. In Glanzberg's definition, this has the result that key concepts are not definable within the language, and in particular, his key conclusion that the Liar Sentence expresses a proposition in some contexts and not others is not definable within the relevant language.

2.3.7 Hartry Field:

Hartry Field has recently offered a very detailed and impressively worked through solution to the Liar Paradox that is based on a generalisation of fuzzy logic semantics. Field develops a new logic which does not satisfy the Law of Excluded Middle for his solution, as he argues that there are certain cases where the Law of Excluded Middle does not hold, including the Liar Paradox and the Sorites Paradox.[29] Despite the significant interesting and original logical work in the detail of his solution, there are core philosophical ideas in Field's approach are similar to other approaches, including Gupta and Belnap's. For example, where Gupta and Belnap rely the concept of categoricalness to avoid the paradox, Field's solution turns on a concept of 'determinateness':

> in which one can say of certain sentences like the Liar that they are neither determinately true nor determinately not true. And saying this isn't just saying that their truth value is unknowable; it has real import about "non-factuality": to say that a sentence A isn't determinately true or determinately untrue commits one to *not* accepting the corresponding instance of excluded middle $A \vee \neg A$[30]

Like Gupta and Belnap, Field's solution rejects condition 7 above, as there are sentences which are neither true nor not true. Field clarifies this rejection by using the concept of determinately

[29] Hartry Field. "Semantic Paradoxes and Vagueness Paradoxes". In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 262–310pp. 262-3

[30] Hartry Field. *Saving Truth from Paradox.* Oxford University Press, 2008, p. 325

true, which gives a plausible reason as to why negation is not the complement in this case. There is nothing that can be determinately said about such sentences, and therefore we cannot say of the relevant properties whether they do or do not hold. Moreover, the concept of determinateness is expressible in the object language in Field's approach, which makes his approach plausibly applicable to natural languages.

Aside from the cost of abandoning the law of excluded middle in certain cases, there is a particular weakness within Field's approach. Field effectively avoids the ordinary Liar Paradox by moving from a concept of truth to a concept of determinate truth. This means that a strengthened paradox can be formulated in terms of determine truth. Field resolves this in turn by moving to a concept of determinately determinate truth, then to determinately determinately determinate truth and so on.[31] While these continual moves work formally, they have obvious intuitive problems in an application to natural languages as it seems alien to rely on an infinite sequence of predicates like this. A further problem is that it is necessary to treat all of these as distinct but related concepts. If we seek to unify them into a single concept, the Liar Paradox bites again and the system is inconsistent. A particular problem is that we have the expressive resources in a natural language to unify the sequence of determinately trues in to a single concept, and it we do so the paradox bites again.

Field does not take this to be a significant problem as his project operates in a similar way to Kripke's. He provides a formal model for a definition of truth that can consistently apply to its own language. He does not offer an explicit definition that might be appropriate within a natural language. As noted before, this approach has a number of limitations in terms of providing a philosophical solution to the Liar Paradox. Firstly, it offers limited philosophical insight, if we are not provided with a means by which we can define and work with a truth predicate *within* a natural language. Secondly, approaches which adopt this method often depend on certain things not being definable within an object language, which are definable within a natural language. While this may provide a nice formal model, it limits the application of these approaches to natural languages, which is where the problem arises and is most acute.

2.4 Summary

The examination of the various solutions in the literature has revealed a number of similarities across the solutions. In terms of the nine conditions above, a range of conditions were relevant to different solutions. The most common condition to not hold was, interestingly, the seventh condition, that sentences are either true or not true, and that this can be expressed within the relevant language. Unfortunately, this condition is at the top of any list of philosophical desiderata, as we would like a language in which we can express the truth or untruth of all sentences in the language.

[31] Field, "Semantic Paradoxes and Vagueness Paradoxes", see n. 3, p. 299

Even more interesting is the fact that, even in solutions which satisfied condition 7, there are almost universally sentences that arise for which we cannot assert an intuitively correct truth value. For Kripke, it is sentences such as "The Liar Sentence is not true"; for the Dialetheist, there are questions over sentences like "The Liar Sentence is both true and false"; for Barwise and Etchemendy, it is sentences about any universal assertion; while both Gupta and Belnap, and Field have sentences which are not true in an intuitive sense, but which are categorised as something else.

Also significant is that all of these solutions rely heavily on a model theoretic approach to the definition of truth in which there is a clear distinction between the model against which the semantic properties are defined and the language. In most cases, including Kripke, Gupta and Belnap, Glanzberg and Field, the truth definition was demonstrated to work in a semantic model without providing a clear definition for the model. There is an element of irony in this, as all of these different solutions are seeking to avoid the problems in Tarski's formal truth definition, while relying crucially on formal semantic techniques which were pioneered by Tarski and, in the case of the recursive semantic definition used in model theory, introduced by Tarski in his formal truth definition.

The existence of these common limitations within existing solutions to the Liar Paradox is striking, particularly as they occur across very different solutions. This suggests that there is some systematic limitation being reached, either in what can be expressed or in the techniques being used. The next chapter will examine this issue and diagnose reasons for these common limitations.

Chapter 3

On the Structure of Formal Languages

As argued in the Chapter 1, the Liar Paradox consists in the fact that any Grammar-Only language which satisfies the given set of four criteria is necessarily affected by the paradox. Furthermore, any Logical language (or any natural language in combination with relevant assumptions about truth and reasoning) in which a further set of five properties hold is necessarily inconsistent and trivialised. A successful solution to the Liar Paradox must give up at least one of these conditions, and the range of solutions presented in Chapter 2 makes it clear that it is not obvious which condition should go.

One of the most challenging aspects of evaluating different solutions to the Liar Paradox is that the paradox itself does not differentiate between the various conditions. If any of the offending conditions does not hold in a logical language, the Paradox does not arise and is therefore dealt with. There is nothing about the Liar Paradox itself that can decide between given solutions.

This means firstly that the justification for a correct solution, and the basis of the criticisms of various solutions must come from somewhere else, not simply the paradox. Thus, for example, a significant criticism of the Barwise-Etchemendy solution is not that it does not resolve the Paradox, but that it does not allow universal assertions and these are not something that we want to give up. Similarly, the dialetheist solution is criticised for its acceptance of true contradictions, and Tarski's solution is criticised for the inability to represent truth in its own language, not the ability to prevent the Paradox. However, there are no explicit generally accepted criteria in the literature against which we can evaluate proposed solution to the Liar Paradox against. Each author tends to define the criteria in a way that naturally privileges their solution.

3.0.1 The Philosophical Problem

The distinction between Grammar-Only and Logical languages, however, offers a clear analysis of the philosophical problem caused by the Liar Paradox, and from that a plausible set of criteria arises that a solution must satisfy to solve the philosophical problem. Importantly, these criteria have support within the literature. In the first chapter, it was argued that any Grammar-Only language which satisfies four criteria is necessarily affected by the Liar Paradox. That is, there is some understanding of reasoning and truth in that language which leads to a contradiction and triviality. Importantly, however, all of these four criteria are satisfied by natural languages. The philosophical problem is that our intuitive concept of truth and reasoning satisfies the further five conditions on a Logical Language being affected and therefore trivialised. This calls into question our ability to use natural languages to express a concept of truth.

A satisfactory philosophical solution to the Liar Paradox must therefore accept the four conditions on a Grammar-Only language being affected, and modify the Logical conditions. Otherwise the solution requires that we seriously modify our use of natural languages in order to talk coherently of truth within natural languages, which requires that we accept that previous use of natural languages is incoherent, at least in the context of discussions about truth. We therefore want to provide a formal language with a truth definition which satisfies these four conditions on Grammar-Only languages. Otherwise, the solution cannot resolve the philosophical problem.

It was shown, however, in the second chapter, that some solutions to the Liar Paradox work by giving up on one or more of the criteria on Grammar-Only languages. Tarski's solution, in which it is not possible to assert the truth of any sentence in the same language, is the most obvious example to this. Importantly, however, much of the research into the Liar Paradox since Tarski has been a conscious attempt to get around the restrictions that Tarski's solution imposed. The most important restriction was that, under Tarski's solution, the facts about truth for a language could only be asserted within another (meta-) language. This seems to directly contradict everything that we do and want to do with natural languages, particularly in discussion of philosophy and linguistics. There is no obvious meta-language to appeal to and if we cannot express certain truths in our normal languages then it is not clear if these truth can ever be expressed. As discussed in the second chapter, there has been a general philosophical aim to find a solution which matches natural languages more closely, and which implies that we do not have to give up on or change natural languages.

3.0.2 Semantic Closure

This aim obviously matches the goal here, which is to find a solution in which all the criteria for a Grammar-Only language to be affected by the Paradox are satisfied, since natural languages satisfy these, but which nevertheless offers an account of reasoning and truth which is not trivialised.

However, it is not good enough that we have an internally coherent account of reasoning and truth, since it is possible to have such an account which leaves out crucial information or facts. Many of the solutions discussed in the second chapter satisfy all the criteria for a Grammar-Only language, in the sense that all of the relevant sentences are grammatical. However, there are grammatical sentences which do not get the correct truth value, according to the truth definition being adopted. For example, Kripke offered a highly coherent and consistent truth definition which could not allow a central truth of that definition, namely that the Liar Sentence is not true, to be expressed and to be true within the language of the definition. Similarly, Gupta and Belnap's solution depends on the fact the the Liar Sentence is not categorical, yet this claim is not even grammatical in the relevant language. The internal coherence was bought at the cost of being able to express all the relevant truths within the language.

This failure has been expressed within parts of the literature as the failure of languages to be semantically closed. A language is semantically closed in this sense if all of the truths about the semantic properties of the language are assertable within the language. Given that the focus of this thesis is on formal languages where the most important semantic information is the truth value of sentences, we will generally restrict the concept of semantic closure to the ability of a language to correctly assert the truth value of every sentence within that language. However, in solutions like Gupta and Belnap's or Field's, concepts such as categoricalness or determinately truth are also crucial semantic concepts and should be expressible with the correct truth value within the language. The most important case to consider is normally whether for a Liar Sentence l and the truth predicate Tr, one of $Tr(l)$, $\neg Tr(l)$ or $Tr(l) \wedge Tr(\neg l)$ is provable and therefore true within the formal language. The point is that, any truth definition should imply that a Liar Sentence is either true, not true (which includes false and neither true nor false) or both true and false, and an important condition on a language being semantically closed is that this fact can be asserted for all sentences with a truth value within the language.

3.0.3 Summary

We have identified two crucial criteria against which an ideal truth definition will be measured. It must allow the definition of a logical language which is semantically closed in this sense and whose grammar-only part satisfies the four criteria listed in Chapter 1.

As is clear from Chapter 2, this ideal has not been achieved by currently proposed formal solutions. What is most interesting in this are the similarities in the way that the different solutions fail. While solutions are capable of dealing with Liar Sentences, they very regularly cannot then allow sentences about those Liar Sentences to have the desired truth value. That is, conclusions which are central to articulating the solution cannot be assigned the correct truth value within the relevant language. Given the ubiquity of this problem, it seems either to be an essential property of the Liar Paradox, or a systematic error in the way that the Liar Paradox is being examined.

As argued in the Introduction, the Liar Paradox should be treated as a paradox that involves languages, rather than only sentences or cases. If there is a systematic error, then this should be one that involves the way that languages, and particularly formal languages, are being treated. It is therefore necessary to examine the structural properties of modern formal languages to see whether they introduce any systematic problems of the sort indicated. Naturally, the structural properties that are most relevant are the assumptions which govern the semantic structure of formal languages. That means, in the modern context, it is vital to examine the consequences of assuming that Recursive Model Theory defines the semantics of formal languages with formal truth definitions and the Liar Paradox. An investigation into the consequences of assuming Model Theory is the purpose of this chapter.

The strategy to be adopted is to isolate the key structural assumptions when a Model Theoretic approach is adopted for semantics, and then to investigate the interaction of these assumptions with formal truth definitions and the Liar Paradox. This will be done by assuming that an arbitrary language which includes a formal truth definition satisfies each of the assumptions, and observing the consequences of liar, and related, sentences. It will be shown that this approach is very fruitful since the structural assumptions within Model Theory are incompatible with a philosophically ideal, semantically closed, formal truth definition.

3.1 Recursive Model Theoretic Semantics

In general terms, modern formal semantics is characterised by an approach which defines the semantic properties of sentences within a language recursively, and with respect to a model of that language. This type of approach, and the mathematics that underlies it, falls under the title of Model Theoretic Semantics. Within the scope of Model Theoretic Semantics, however, there are two key assumptions that are separable and have different consequences when we examine them with respect to formal truth definitions.

The two relevant assumptions are i) the concept of a *semantic model*, i.e. that there is some formal structure (the model) against which all of the semantic properties of sentences in a language can be defined; and ii) the assumption that these semantic properties can be defined *recursively*, that is, they can be defined initially for the most basic sentence components and then recursively for more complicated sentences based on the grammatical structure of the sentences. While these are normally two parts to the one model theoretic definition, the two assumptions will be investigated separately here since each assumption has different consequences for formal truth definitions.

The concept of a semantic model, as something against which the semantic properties (particularly the truth values) of all sentences in a language is defined, is of obvious interest for our understanding of truth within a formal language. On this method of defining semantic properties, a sentence in the language is true if it is true with respect to the model. It follows that the act of defining the truth value of sentences in a language against a model assumes that the relevant concept of truth for sentences is truth in the model. For this reason, we will investigate this concept first and look at whether defining semantic properties in this way has any effect on the successful solution of the Liar Paradox. It will be argued that no semantically closed language which includes a truth predicate can have a Model which defines the semantic properties of all the sentences of that language. That is, no semantically closed formal language is sound in the normal model theoretic sense. Recursivity will not be assumed in the analysis and so this conclusion is independent of the assumption that the semantic definition for the language is recursive.

This fundamentally important conclusion can be interpreted in two different ways. Firstly, if one assumes the necessity of the standard model theoretic approach to formal semantics, then it provides a clear proof of Tarski's conclusion that a semantically closed language is impossible.[1] On the other hand, if alternative approaches to defining the semantics of formal languages are possible, then this conclusion shows that the definition of formal semantics needs to be changed, if we wish to achieve semantic closure.

This conclusion is not only supported by a formal argument, but is supported by a conceptual analysis of logical structure of model theory. It will be argued that the structure of model theory is not consistent with some minimal conditions on the concept of truth. Thus the inability of formal definitions to achieve semantic closure is not an accident but is a necessary consequence of the conceptual structure of standard model theory. Any truth predicate that reflects the concept of truth within a model cannot allow sentences such as Liar Sentences to be completely semantically evaluated against the model since the structure of the system of reference prevents such sentences from being interpreted within the model.

The assumption of Recursivity will be investigated separately from this analysis of the concept of a model. It will also be shown that assuming a recursive definition of semantics is problematic in the context of a satisfactory solution to the Liar Paradox. There are phenomena which are essentially involved in the paradox which cannot be accommodated within a recursive semantics. It will be argued, however, that this is not inherently due to the assumption of Recursivity, but due to the assumption that sentences (or sentence types) are truth bearers and that identical sentences have identical truth values.

In order to unify the following discussion, we will universally assume that we have a formal language $\mathcal{LT}$ which includes a truth predicate Tr and a system of naming. As a matter of notation, we will let upper case letters, such as P, Q and R, be metatheoretic symbols that stand for sentences within $\mathcal{LT}$, and we will use the $\ulcorner\urcorner$ notation as a name forming device on sentences in $\mathcal{LT}$. Thus,

[1] Alfred Tarski. "The Semantic Conception of Truth: and the Foundations of Semantics". In: *Philosophy and Phenomenological Research* 4.3 (1944), pp. 341–376pp. 348-9

$\ulcorner P \urcorner$ is a symbol for a name of the sentence P. However, in order to discuss the Liar Paradox, we need to have actual cases to discuss. Actual cases necessarily involve some particular sentence within the language, and we will therefore assume that $\mathcal{LT}$ contains the following sentences: $Tr(a)$ with the name a; $\neg Tr(b)$ with the name b; $\neg Tr(b) \wedge \neg Tr(b)$ with the name c; and $Tr(b)$ with the name d. In terms of notation, we are using lower case letters (i.e. a, b, c, d) as names of sentences. Strictly speaking, these names are defined within $\mathcal{LT}$, but to avoid the confusion of assigning the same sentence two names, we will use the same lower case letter as the metatheoretic name of the sentence. Where it is important to distinguish between the letter as a name and the letter as standing for the sentence, we will be careful to clarify.

3.2 The Definition of Semantic Model

As noted above, the orthodox way to approach the definition of semantics for formal languages is to use Model Theory. The key idea of this is that as sentences within a formal language do not have any intrinsic semantic properties, all of the semantic properties of sentences within a formal language are defined with respect to an external, precisely defined structure, a Model. This means, among other things, that concept of truth for sentences in a formal language only makes sense when it is the concept of truth in the Model. Given that formal languages include a proof structure, a Model can only be used to define the semantic properties of a formal language, if every provable sentence in the language is defined to be true with respect to the Model.

This idea is simple and intuitively appealing as it seems to capture something important about formal languages. Formal languages are defined by allowing certain types of strings of symbols as legitimate, and allowing certain types of derivation patterns to be carried out on strings of symbols. The meanings of different symbols, and at times derivation rules, are not fixed by the definition of the language. Instead they are abstract symbols that can be used in different contexts to mean different things. This method of definition, which is very similar to the abstraction that occurs in mathematics, is one of the strengths of formal logic. There is no ambiguity in the reasoning allowed, yet the formal language can be used to describe any system which has the same structure. Thus, for example, modal logics are often used to study knowledge, since knowledge operators share certain structural similarities to necessity operators, even though intuitively the meanings are very different.

The concept of a semantic model turns this intuition into a precise formal definition. The semantic properties of sentences in a language not defined as part of the definition of the language, but are defined with respect to something external which can be seen to supply meanings and truth values to symbols. This is structurally similar to the way that the meaning and truth of natural language sentences are defined with reference to the real world, and the real world is obviously external to any natural language.

If we follow this angle, it does not make sense to ask about the semantic properties of the strings of symbols, since they depend on the context of application. However, when we are considering truth values, and meanings of certain logical connectives, the situation is not as straightforward. Within a formal language, certain strings of symbols are always provable from the derivation rules, and therefore must always be true. Otherwise the notion of provability does not make sense. The question then arises as to whether the truth of these strings of symbols is a fact about the formal language, and is therefore not defined against something external. There is significant debate about these issues, including whether logical consequence is a model theoretic or proof theoretic notion.

As we are focusing on Model Theoretic semantics, we only need to consider the solution to this question adopted within model theory. A model theoretic definition treats the truth values of all sentences in the same way: they are defined with respect to a model. The provable sentences, however, also dictate which formal structures are allowable models, for only structures in which the provable sentences are true can reasonably supply a definition of the semantic properties of a language. This means that formal structure can only be a model for a formal language if the structure and language have enough structural similarity.

The key point is that the semantic properties, or at least the truth values, of all sentences within a language are defined with respect to a model in Model Theoretic semantics. Truth for a formal language is normally defined, and thought of, as 'truth in a Model.'

As a point of practise, there are normally multiple, non-identical models of any formal language. This does not mean, however, that there are multiple concepts of truth, or that the concept of truth is badly defined. The requirement that all provable sentences are true in the model means that all models will agree on at least the provable sentences, so there will always be a set of sentences which are true in all models. This means that often the most useful concept of truth to work with is "true in all models". In other cases, where it is clear what sort of structure a language is meant to describe, the concept of truth that is adopted is "true in the intended model". Thus, for example, the intended model for Peano Arithmetic in various formal language is number theory on the set of natural numbers, even though there are other non-isomorphic models.

While each of these approaches are not identical, they do not represent different concepts of truth and the same considerations arise. The analysis in this chapter applies equally, whether we are considering truth in a model, all models or an intended model.

3.2.1 The Definition of a Model

There are problems that arise when a formal truth predicate is defined within a model theoretic semantics. These problems arise from the combination of any standard definition of a model with the intuitively correct properties for a formal truth predicate within model theoretic semantics.

It is important to note in what follows that we are not assuming anything about the recursivity of the model theoretic definitions. While model theory is almost universally defined recursively, recursivity and the concept of model theory are separable assumptions, which are being analysed separately in this chapter. Some of the definitions may seem odd at first, but this is because recursivity is not assumed.

To see the problems, we must begin with a definition of a model. Given that we are only investigating the core concept of a model, we need only a very generic definition. The following is a standard definition:

Definition. Let $\mathcal{L}$ be a formal language, $\vdash$ denote consequence in $\mathcal{L}$ and $\mathcal{M}$ some mathematical structure. $\mathcal{M}$ is a *model of* $\mathcal{L}$ if, and only if, the following condition holds, for any sentence P in $\mathcal{L}$:

If $\vdash P$ then $\mathcal{M} \models P$

That is, $\mathcal{M}$ is only a model of $\mathcal{L}$ if every theorem in $\mathcal{L}$ (i.e. every sentence in $\mathcal{L}$ that is provable from no assumptions) holds in $\mathcal{M}$.

While this definition captures the key idea, it leaves an important philosophical aspect of the definition implicit. P is defined as a sentence of the language, and is therefore not an element of the model $\mathcal{M}$. To be precise, it does not make sense to write $\mathcal{M} \models P$ to mean that P holds in $\mathcal{M}$. The relationship between the language and the model must be set up by means of some interpretation or valuation function which associates sentences in the language with elements of the model. For example, in Classical Logic, there are valuation functions which interpret sentences in the language as truth values in the model. This function is normally treated as a part of the definition of the model, and is left out of the supplied definition since including it would make the notation harder to follow, and it does not affect the argument in any way.

The definition also only considers one particular type of model, where we have a model of the language itself. Often it is more useful to look at models of a theory within the language. In this context a theory is simply a consistent set of sentences within the language. The definition of a model of a theory in a language is essentially identical to the above definition:

Definition. Let $\mathcal{L}$ be a formal language, $\vdash$ denote consequence in $\mathcal{L}$, $\mathcal{T}$ be a theory in that language and $\mathcal{M}_{\mathcal{T}}$ be some mathematical structure. $\mathcal{M}_{\mathcal{T}}$ is a *model of* $\mathcal{T}$ if, and only if, the following condition holds, for any sentence P in $\mathcal{L}$:

If $\mathcal{T} \vdash P$ then $\mathcal{M}_{\mathcal{T}} \models P$

The original definition is a special case of this when we have an empty theory.

It was noted above that there are occasions when "true in an intended model" or "true in all models" are more useful concepts that simply "true in a model". If we are working with truth in an intended model, then the above definitions will hold (otherwise the intended model could not be a model). Truth in all models, which is often represented formally as $\models P$ as opposed to $\mathcal{M} \models P$, is a more complicated notion. The key difference is that it is normally assumed that a Model is *complete* in the sense that every sentence in the language is assigned a truth value by the Model. That is, in a classical model, for any sentence P, only one of $\mathcal{M}_{\mathcal{T}} \models P$ and $\mathcal{M}_{\mathcal{T}} \models \neg P$ holds. When we are considering truth in all models, it is not possible that every sentence has an identical truth value in every model, so there will be many (contingent) sentences which do not have a truth value with respect to the set of all models. Nevertheless, the above definition in the form of "If $\vdash P$ then $\models P$" since "If $\vdash P$ then $\mathcal{M} \models P$" must hold for every model.

These definitions, plus the requirement that a model is complete, provide a general characterisation of semantic models. However, the key factors in an examination of formal truth definitions are the properties a satisfactory truth predicate must have in a model theoretic context.

3.2.2 The Truth Predicate within Model Theory

The aim of this investigation is to evaluate the consequences of assuming a Model Theoretic definition of semantics for formal truth definitions. The important factors that therefore need to be established are the semantic properties of any truth predicate if a model theoretic definition is adopted. In order investigate this, we will assume that we have our fixed language $\mathcal{LT}$ includes a fixed theory $\mathcal{T}$, a truth predicate Tr and has a valid model of the theory $\mathcal{M}_{\mathcal{T}}$. The theory may be a null theory, but we will allow there to be a theory as it is a more general case. This means that the principle given in the definition above, namely that "If $\mathcal{T} \vdash P$ then $\mathcal{M}_{\mathcal{T}} \models P$" is assumed to hold. We will motivate a number of different principles which must hold if the truth predicate is to capture, even extensionally, the same concept as being true in a model.

The fundamental idea of the model theoretic approach to semantics is that the semantic properties, such as truth, of sentences within a language are measured against the models of that language. Hence, a sentence P in $\mathcal{LT}$ is true on the assumption of the theory $\mathcal{T}$ just in those cases that the interpretation of P within the model $\mathcal{M}_{\mathcal{T}}$, is true in the all the models that $\mathcal{T}$ is true. This suggests that the following principle should be satisfied by a formal truth predicate:

Condition 1: $\mathcal{T} \vdash Tr\ulcorner P \urcorner$ iff $\mathcal{M}_{\mathcal{T}} \models P$

That is, for any satisfactory truth predicate, 'P is true' should be provable from $\mathcal{T}$ in exactly those cases when P is true in the model $\mathcal{M}_{\mathcal{T}}$. Assuming this, however, would imply that everything that is true is provable in the language (on the assumption of the relevant theory). This is in general simply not the case, since we are only rarely interested in theories that assign a truth value to all sentences. This condition needs therefore to be weakened, and we use one half of the biconditional in our first condition on a satisfactory truth predicate:

Satisfactory Truth Principle 1: If $\mathcal{T} \vdash Tr\ulcorner P\urcorner$ then $\mathcal{M}_{\mathcal{T}} \models P$

This obviously captures the idea of a truth predicate in a model theoretic context. If it can be proven that a sentence is true, then that sentence must hold in the model. To put this point differently, we can consider a counterexample. If there is some sentence Q, such that $\mathcal{T} \vdash Tr\ulcorner q\urcorner$ but $\mathcal{M}_{\mathcal{T}} \not\models_v Q$, then the truth predicate cannot be satisfactorily defined - for it identifies as true a sentence which is not true in the relevant model. However, this suggests that we need to add a further criteria on the satisfactory definition of the truth predicate, for we also do not want sentences to be identified as not true, which are true in the model. That is, our second condition on a satisfactory truth predicate is:

Satisfactory Truth Principle 2: If $\mathcal{T} \vdash \neg Tr\ulcorner P\urcorner$ then $\mathcal{M}_{\mathcal{T}} \not\models P$[2]

These two definitions must be satisfied for any formal predicate which can reasonably be interpreted as truth in a model theoretic setting. Thus, for any formal truth definition, if these two principles are satisfied, then the definition satisfies a minimal criterion on being a reasonable truth definition.

However, considering the definition of a truth predicate within a model theoretic framework raises the further important question. It is assumed that the semantic properties for every sentence in a language are given by the model. $\mathcal{LT}$ includes a truth predicate, so it follows that every sentence which includes a truth predicate must be interpretable within the model as the model must assign it a truth value. To put it differently, since every valid model $\mathcal{M}$ must satisfy the condition: If $\mathcal{T} \vdash P$ then $\mathcal{M}_{\mathcal{T}} \models P$; then it must also satisfy the condition: If $\mathcal{T} \vdash Tr\ulcorner P\urcorner$ then $\mathcal{M}_{\mathcal{T}} \models Tr\ulcorner P\urcorner$. However, this condition is only coherent if there is an interpretation of $Tr\ulcorner P\urcorner$ in each model $\mathcal{M}$, for otherwise $\mathcal{M}_{\mathcal{T}} \models Tr\ulcorner P\urcorner$ literally does not make sense.

Without being given any information about $\mathcal{LT}$, it is not possible to determine what the interpretation of $Tr\ulcorner P\urcorner$ in any model is for any P. The interpretation depends clearly on the relevant language and model. However, we can identify a natural condition it should satisfy. Quite simply, whenever P is true in $\mathcal{M}$, $Tr\ulcorner P\urcorner$should also be true in $\mathcal{M}$. That is:

Tr Interpretation Principle: $\mathcal{M}_{\mathcal{T}} \models Tr\ulcorner P\urcorner$ iff $\mathcal{M}_{\mathcal{T}} \models P$

If this principle is not satisfied, then the Tr predicate cannot be said to represent the concept of truth in the model. If it was not true then there would be some sentence that is true in a model, but the assertion that it is true is not true in that model. Or alternatively, there would be a sentence that is not true in the model, but the assertion that it is true is true in the model. Both of these alternatives would imply that the truth predicate is not a genuine truth predicate. Thus the interpretation of any formal truth predicate within the relevant model must satisfy this principle. We would normally expect that this implies the interpretation of $Tr\ulcorner P\urcorner$ will in some way depend on the interpretation of P, but this is not a necessary assumption.

[2] We are not assuming anything about the definition of negation, except that it is not paraconsistent.

We have therefore identified three principles which must hold within a Model Theoretic framework, if a formal truth predicate is to represent the concept of truth within model theory in any meaningful sense. These Principles are not independent, however, as Satisfactory Truth Principle 1 can be derived from the Tr Interpretation Principle, and the definition of a model. Thus although they can be independently motivated, they follow from a basic restriction on the properties a truth predicate must hold in a model. However, these Principles are not sufficient for a discussion of the Liar Paradox, since we do not yet have any rules governing the behaviour of negation within a model theoretic context. Nevertheless, they allow for a discussion of the Truth Teller within Model Theory, which is an important related phenomenon.

3.2.3 The Truth Teller

A Truth Teller Sentence is one which asserts of itself that it is true. In order to investigate the properties of a Truth Teller sentence within the model theoretic framework, we have assumed that our language $\mathcal{LT}$ includes the sentence $Tr(a)$, which has the name a. The Tr Interpretation Principle, which is meant to place a necessary condition on the interpretation of a, is the following:

Tr Int. for a: $\mathcal{M}_{\mathcal{T}} \models Tr(a)$ iff $\mathcal{M}_{\mathcal{T}} \models Tr(a)$

This is trivial but vacuous as it does not tell us anything. The two Satisfactory Truth Principles are likewise vacuous:

Satisfactory Truth Principle 1: If $\mathcal{T} \vdash Tr(a)$ then $\mathcal{M}_{\mathcal{T}} \models Tr(a)$

Satisfactory Truth Principle 2: If $\mathcal{T} \vdash \neg Tr(a)$ then $\mathcal{M}_{\mathcal{T}} \not\models Tr(a)$

They do not tell us anything more than the fact that $\mathcal{M}_{\mathcal{T}}$ is a model of $\mathcal{T}$. Obviously if we try to use any of these principles in reasoning about the Truth Teller, we simply get back to what we already know about the Truth Teller. It is true if and only if it is true, and it is not true if and only if it is not true. However, it adds a new dimension to our understanding, when we consider the question within the model theoretic context.

With model theoretic semantics, the sentence a is true (or has any other truth value) only if the model assigns a truth status to a. Moreover, for the majority of sentences of the form $Tr\ulcorner P \urcorner$, the Tr Interpretation Principle is itself sufficient to establish the truth value that the sentence should have. $Tr\ulcorner P \urcorner$ is only a valid sentence if P is a sentence in $\mathcal{LT}$. If P is a sentence in $\mathcal{LT}$ then it must have some truth status with respect to $\mathcal{M}$. The status of $Tr\ulcorner P \urcorner$ with respect to $\mathcal{M}$ will follow from this. The problem in the case of a is that its definition (as $Tr(a)$) does not fix an interpretation or truth value in $\mathcal{M}$. The definition in itself is not enough to allow the model to determine a truth status for a, or in other words the information contained in a underdetermines its truth value in the model.

This is not a problem if we are considering a fixed model $\mathcal{M}_{\mathcal{T}}$, since this model will presumably contain many sentences without a determinate truth value, and the model assigns truth values to all such sentences. However, when we consider truth in all models, a difficulty arises. Sentences which get an arbitrary truth value assigned will have one truth value in some models and a different truth value in other models. This is to be expected when the sentences are contingent, and therefore have a truth value that depends on the relevant facts. The sentence a on the other hand does not appear contingent, and it seems weird to suppose that this sentence could have different truth values in different contexts. The only ways to fix the truth value of a are an intended model, or to add some extra principle(s) to the language and/or model which assign truth values to sentences like a. Either approach requires justification, and requires appealing to information beyond that contained in a to determine a truth value.

This problem is surmountable but requires justification as to the correct extra principles or the correct intended model. In one sense, this simply confirms the observation above that the Liar Paradox itself does not dictate a solution. We must assess any solutions against criteria external to the Liar Paradox. However, this analysis reveals something of the structure of the problem, and identifies an area where more serious problems arise when consider Liar Sentences. However, in order to that we must understand the basic properties of negation within Model Theory.

3.2.4 Negation

Negation, and how to understand it is a very large topic, particularly in the context of discussions of the Liar Paradox. Many different solutions to the Liar Paradox require different assumptions about the correct properties of negation. However, as pointed our in Chapter 1, the key to solving the Liar Paradox cannot be in our understanding of negation as there are languages with no negation that can be affected by the paradox. For the sake of this analysis, it is necessary not to prejudge these discussions as much as possible. However, analysing the situation requires something and we will make the assumption that we are dealing with consistent models (at least with respect to the relevant negation). This is a non-trivial assumption in the context of discussions of the Liar Paradox, yet is widely accepted and justifiable. To be more precise, we will assume that:

Consistency of Models: For any sentence P in a language $\mathcal{L}$ with a model $\mathcal{M}_{\mathcal{T}}$, at most one of $\mathcal{M}_{\mathcal{T}} \models P$ and $\mathcal{M}_{\mathcal{T}} \models \neg P$ holds.

This assumption has significant consequences, particularly via a couple of obvious corollaries. Firstly, it follows from the Consistency of Models Principle that if $\mathcal{M}_{\mathcal{T}} \models \neg P$, then it cannot be the case that $\mathcal{M}_{\mathcal{T}} \models P$. That is, if $\mathcal{M}_{\mathcal{T}} \models \neg P$, then $\mathcal{M}_{\mathcal{T}} \not\models P$. Secondly, if we assume that a model assigns a truth value to every sentence, in other words that the model is complete, it follows that if $\mathcal{M}_{\mathcal{T}} \not\models P$, then $\mathcal{M}_{\mathcal{T}} \models \neg P$.

This principle and its corollaries, combined with the assumptions on a satisfactory definition of the truth predicate in a model theoretic framework, give us enough to begin an investigation of the Liar Paradox. This will be split into several parts.

3.2.5 The Liar Sentence itself

The first focus of investigation will naturally be the archetypal Liar Sentence b in our language $\mathcal{LT}$. As stated before, b is the sentence $\neg Tr(b)$. This sentence is the most simple form of liar sentence, and we wish to evaluate this sentence with respect to its semantic properties in the model, $\mathcal{M}_{\mathcal{T}}$. As a valid sentence within the language $\mathcal{LT}$, b should have a truth value in $\mathcal{M}_{\mathcal{T}}$. However, the Tr Interpretation Principle for b is:

Tr Int. for b: $\mathcal{M}_{\mathcal{T}} \models Tr(b)$ iff $\mathcal{M}_{\mathcal{T}} \models \neg Tr(b)$

This is obviously contradictory, and obviously means that neither $Tr(b)$ nor b (which is $\neg Tr(b)$) can hold consistently in $\mathcal{M}_{\mathcal{T}}$. However, the problem is more serious than this. Firstly, in order to bring out the reasons that b does not hold in $\mathcal{M}_{\mathcal{T}}$, we assume that b does hold in $\mathcal{M}_{\mathcal{T}}$, i.e. that $\mathcal{M}_{\mathcal{T}} \models \neg Tr(b)$. We get the following valid argument on the principles developed here:

1. $\mathcal{M}_{\mathcal{T}} \models \neg Tr(b)$ (Defn of b)

2. $\mathcal{M}_{\mathcal{T}} \not\models Tr(b)$ (Corollary to Consistency of Models)

3. $\mathcal{M}_{\mathcal{T}} \not\models \neg Tr(b)$ (Contraposed Tr Interpretation Principle i.e. if $Tr(b)$ does not hold, then b cannot hold)

That is, from the principles we have established, if we assume that b holds in the model, then it follows that it does not hold. Given that it does not make sense that a sentence both holds and does not hold in the model, our assumption must be incorrect, and so it should follow that b does not hold in $\mathcal{M}_{\mathcal{T}}$. However, if we assume that b does not hold in $\mathcal{M}_{\mathcal{T}}$, that is we assume that $\mathcal{M}_{\mathcal{T}} \not\models \neg Tr(b)$, then the following argument holds:

1. $\mathcal{M}_{\mathcal{T}} \not\models \neg Tr(b)$ Assumption

2. $\mathcal{M}_{\mathcal{T}} \not\models Tr(b)$ (Contraposed Tr Interpretation Principle,)

3. $\mathcal{M}_{\mathcal{T}} \models \neg Tr(b)$ (Corollary to Consistency of Models)

Assuming that b does not hold in $\mathcal{M}_{\mathcal{T}}$ leads to the conclusion that it must hold. This means that, for any given model, b can hold in the model if, and only if, it does not hold in the model. In other words, the basic conditions imposed on the definition of a formal truth predicate above lead to a completely contradictory characterisation of the semantic properties of the liar sentence b. If we assume that these conditions are valid, which is difficult to argue against without undermining the idea of a model theoretic semantics, the sentence b cannot consistently have any properties with respect to the model $\mathcal{M}$. That means that it cannot be consistently semantically interpreted within the model theoretic framework.

Furthermore, even if we ignore the contradictory nature of the definition, we notice that none of these lines provide any more information about the semantic properties of b than the original assumption. As in the case of the Truth Teller, the Principles in question do not allow the derivation of any useful information about the semantic status of the sentence in question. Thus b does not provide sufficient information to allow a semantic evaluation with respect to a model.

Given some innocuous assumptions about the nature of truth and negation within a model, it is not possible for the model to coherently assign a consistent semantic status or evaluation to the Liar Sentence b. This is partly because it is not possible for the model $\mathcal{M}_{\mathcal{T}}$ to provide any relevant information about b or for b to be interpreted within $\mathcal{M}_{\mathcal{T}}$. However since $\mathcal{LT}$ and $\mathcal{M}_{\mathcal{T}}$ simply represent a language with the basic semantic framework set up by a standard model theoretic definition, the structure of model theory itself impedes a consistent semantically closed solution to the Liar Paradox.

3.2.6 Sentences about a Liar Sentence

Although it is clear from the consideration of liar sentences that there are limitations inherent within the structure of model theory, the problems that commonly arise mean that it is important to investigate another type of problematic sentence within the same context, that is sentences about a liar sentence. To do this, we will adopt the same framework as the previous sections with the language $\mathcal{LT}$ with a model $\mathcal{M}_{\mathcal{T}}$, with the same governing principles. We will also consider the sentence $\neg Tr(b)$ with the name b; a sentence $Tr(b)$ whose name is d; and the sentence $\neg Tr(b) \wedge \neg Tr(b)$ with the name c.

Unsurprisingly, the semantic properties of d within this context are identical to those of b, since d is the sentence $Tr(b)$. By two applications of the Tr Interpretation Principle, it follows $\mathcal{M}_{\mathcal{T}} \models d$ iff $\mathcal{M}_{\mathcal{T}} \models Tr(b)$ iff $\mathcal{M}_{\mathcal{T}} \models b$. This means that, since b does not have a consistent semantic evaluation, d cannot have a consistent semantic evaluation within this structure, and that neither d nor its negation (however that is interpreted) can be true. Similarly, the semantics properties of c, assuming the obvious principle that $\mathcal{M} \models A \wedge A$ iff $\mathcal{M} \models A$, are identical to that of b. This time an application of the Tr Interpretation Principle with the definition of b leads to: $\mathcal{M}_{\mathcal{T}} \models c$ iff $\mathcal{M}_{\mathcal{T}} \models \neg Tr(b)$ iff $\mathcal{M}_{\mathcal{T}} \models b$. Thus c holds exactly when b does, but b cannot consistently hold. So neither c nor d can have a consistent semantic evaluation.

These two sentences, however, represent the two key options for reporting the truth value of the sentence b within a language. If b is either true or not true, then either d or c should be true. Thus the structure of model theory, since all the semantic properties are determined with respect to the model, cannot differentiate between liar sentences and any other sentences about a liar sentence since none of them can have a semantic status with respect to the model. This causes problems for the reporting of truth values within the language and hence for semantic closure.

3.2.7 The Liar Sentence and Semantic Closure

A model theoretic approach to semantics cannot consistently assign a Liar Sentence any semantic properties within the model. However, this does not prevent semantic closure since this fact may be sufficient, for example, to justify that $\neg Tr(b)$. From this it would follow that it is possible for the language to be semantically closed, for the crucial test which we are applying is whether one of $Tr(b)$, $\neg Tr(b)$ or $Tr(b) \wedge \neg Tr(b)$is justified as true within the language. However, the conditions that were justified on a Satisfactory Definition of truth within a model theoretic context prevent this from occurring.

To see this, we will first repeat the relevant Principles for reference:

Satisfactory Truth Principle 1: If $\vdash Tr\ulcorner P\urcorner$ then $\mathcal{M} \models P$

Satisfactory Truth Principle 2: If $\vdash \neg Tr\ulcorner P\urcorner$ then $\mathcal{M} \not\models P$

Tr Interpretation Principle: $\mathcal{M} \models Tr\ulcorner P\urcorner$ iff $\mathcal{M} \models P$

Consistency of Models: For any sentence P in a language $\mathcal{L}$ with a model $\mathcal{M}_{\mathcal{T}}$, at most one of $\mathcal{M}_{\mathcal{T}} \models P$ and $\mathcal{M}_{\mathcal{T}} \models \neg P$ holds.

On this basis we can prove the following theorem:

Theorem. *If a language $\mathcal{LT}$ which includes a truth predicate Tr is semantically closed, then $\mathcal{LT}$ has no complete semantic model.*[3]

Proof. Firstly, we assume that $\mathcal{LT}$ is semantically closed, and it includes the Liar Sentence b, as defined previously. It follows that either $\vdash Tr(b)$ or $\vdash \neg Tr(b)$ (for some negation in $\mathcal{LT}$). We note that if $\mathcal{LT}$ has a semantic model, then for all provable sentences within $\mathcal{LT}$, if $\vdash P$ then $\mathcal{M} \models P$. That is, P is interpretable and holds in $\mathcal{M}$.

We assume that $\vdash Tr(b)$ and that $\mathcal{LT}$ has a model $\mathcal{M}$. If follows then that $\mathcal{M} \models Tr(b)$. From the Tr Interpretation Principle, it follows that $\mathcal{M} \models b$. However we have shown above that if $\mathcal{M} \models b$, then $\mathcal{M} \not\models b$. Thus, it is not possible that both $\vdash Tr(b)$ and $\mathcal{LT}$ has a model.

Similarly, we assume that $\vdash \neg Tr(b)$ and that $\mathcal{LT}$ has a model $\mathcal{M}$. It follows that $\mathcal{M} \not\models Tr(b)$, by the conditions on negation in a model theoretic context. Then, by the Tr Interpretation Principle, it follows that $\mathcal{M} \not\models b$. From this it follows, as shown previously, that $\mathcal{M} \models b$. That is, on pain of contradiction, it is not possible that both $\vdash \neg Tr(b)$ and that $\mathcal{LT}$ has a model $\mathcal{M}$.

Hence if $\mathcal{LT}$ is semantically closed, then it can have no complete model. □

[3] Complete is here being used in the sense discussed above: a complete model is one in which every sentence in the language has a truth value.

This fundamentally important conclusion can be interpreted in two different ways. Firstly, if one assumes the necessity of the model theoretic approach to formal semantics, then it provides a clear proof of Tarski's conclusion that a semantically closed language is impossible. On the other hand, if we believe that there are alternative approaches to defining the semantics of formal languages, then this conclusion shows that we need change the way that formal semantics are defined, if we wish to achieve semantic closure. In order to work out whether this is possible, it is necessary to consider conceptually where the problems arise in Model Theory, which is the focus of the next section.

3.3 The Conceptual Structure of Model Theory

The result just proven demonstrates that adopting a Model Theoretic Semantics necessarily places limitations on what can be achieved by a formal truth definition. If we accept the necessity of Model Theoretic Semantics, then semantic closure is not possible. Any formal truth definition that aims to be semantically closed must therefore, at least, fundamentally modify some aspects of the Model Theoretic approach. Nevertheless, the success and enormous utility of Model Theory as an account of formal semantics demonstrates that, even if there are problems with it in the particular context of formal truth definitions, any attempt to resolve these problems should attempt make as few changes to the standard structure as are necessary. That is, we will attempt to adhere to a principle of minimal mutilation, changes will only be made where necessary and the theory should revert back as much as possible to the standard theory when we take out considerations about formal truth definitions.

In order to achieve this, it is necessary to have a clearer understanding of the reasons why there are problems with the orthodox semantic approach. For this purpose, it is vital to examine the conceptual structure which supports the model theoretic approach in order to identify the causes of the identified problems. When one considers the conceptual structure of Model Theory, the problems arise very intuitively, and in turn they suggest a strategy by which we might resolve them. This can be made particularly clear if we consider the relationship between formal language (or syntax) and model as analogous to the relationship between (natural) language and the world.

As has been made clear, within the Model Theoretic semantic approach, the idea is that the semantic information (at a minimum: truth) for sentences in a language is only determined by the Model(s), which is (are) separate from the syntax of the language. If we translate this into the analogy between language and world, it means that all of the semantic information, and in particular truth values, about sentences within a language are purely determined by what is the case in the world; and that the language and the world are strictly separate. Furthermore, the semantic properties of sentences within the language are fixed by how they are interpreted with respect to the world.

If we then ask the question, for example, whether a Truth Teller Sentence would be true or not on this conceptual picture, the formal problem identified above immediately arises. If the Truth Teller has any semantic properties, it must be interpreted within the world. However, a Truth Teller Sentence says that it itself is true, which is a claim about the semantic properties of an element in the language. That is, it does not say anything that can be directly interpreted within the world, since the language and world are strictly separate on the model theoretic conceptual structure. For ordinary truth ascriptions, say of the form 'D is true', this problem is resolved since the sentences make indirect claims about the way the world is via the sentence D. In other words, they are grounded. A Truth Teller, however, only makes a claim about the semantic properties of a sentence which makes the same claim about the semantic properties of the same sentence. That is, it cannot, in principle, be interpreted as about the world and since world and language are separate, it is not possible to evaluate the truth value of a Truth Teller sentence with respect to the world.

This problem obviously affects Liar and other similar sentences equally, and the reasoning involved exactly mirrors the argument given above. Furthermore, it follows that any sentences about a Liar Sentence also cannot be interpreted as about the world, and hence they have the same semantic status with respect to the world/model. Thus the model theoretic semantic structure cannot differentiate between paradoxical sentences and sentences about paradoxical sentences. This demonstrates that the conceptual structure of model theory, and in particular the way that semantic information is determined for all sentences with respect to the model, is incapable of dealing with the Liar Paradox in an intuitively satisfactory way.

The problems with the conceptual structure, in turn, illustrate why semantic closure is not possible within a model theoretic framework. Semantic closure requires that one of $Tr(b)$ or $\neg Tr(b)$ is provable within the language. However, within a model theoretic framework, if we can assert either $Tr(b)$ or $\neg Tr(b)$, then the truth value of these sentences must be definable with respect to a model. It follows that if one of $Tr(b)$ or $\neg Tr(b)$ is provable, we must be able to semantically evaluate the sentence b with respect to the model; otherwise there is a truth that is expressible within the language but not in the model, which contradicts the assumptions of model theory. However, if b is a liar sentence, this is not possible, and hence semantic closure is not possible within a model theoretic framework.

This analysis of the problems with the conceptual structure of model theory also identifies why there are problems with Liar (and related) Sentences, but not with other self referential sentences. If we take, for example, the sentence "This sentence has thirty one letters", the semantic interpretation of this sentence does not depend on the semantic evaluation of the sentence itself. It is determined by other properties of the sentence which can be identified independently of the semantic properties of the sentence - in this case by counting the letters. In the case of a Liar Sentence, the semantic interpretation of the sentence depends on being able to independently identify the semantic interpretation of the same sentence, which is impossible. Thus self-reference itself is not a problem, only the type of semantic self-reference that is exhibited in the Liar Paradox.

This fact that that self-reference is only a problem in the context of semantic predicates such as truth, indicates that, if we are to adhere to a principle of minimal mutilation, it is only necessary to change the way that our semantic account of formal languages deals with semantic predicates. How this can be achieved will be left to the next chapter, as there is the other assumption inherent in modern formal semantics which needs to be examined.

3.4 Recursivity

At the beginning of this chapter, it was argued that there are two key assumptions present in the orthodox approach to the formal definition of semantics: the concept of a semantic model and the assumption that semantic values are defined recursively. These assumptions were deliberately separated as each assumption has different consequences when we want to define a formal truth predicate. The previous sections have shown that assuming that all semantic properties are defined with respect to a model is problematic when we are trying to define a formal truth predicate. Importantly, this was shown without making any assumptions as to whether the semantic definitions are recursive. This section will look at the assumption of recursivity.

Assuming that the definition of the semantic properties of sentences is recursive is one of the most productive and important assumptions with the orthodox approach to formal semantics. That is, the assumption that the semantic properties of the atomic sentences, or (depending on the type of language) the most basic sentence components, are initially defined and the semantic properties of complex sentences are iteratively defined from the semantic properties of the most basic sentence components by defining the way that the rules of sentence construction transmit semantic values. Thus, to take a very simple example, if $\mathcal{LT}$ is a sentential logic, and P, Q and R are atomic sentences, then a recursive definition works as follows:

1. Assign truth values to P, Q and R - say T, T and F respectively..

2. Define the semantic properties of connectives. For example, $A \wedge B$ is true exactly when A and B are each true.

3. Iteratively assign truth values to complex sentences (such as $(P \wedge Q) \wedge (R \wedge Q)$) by means of the truth values defined in Step 1, and the properties of the connectives defined in step 2. Thus, in our example, we first determine that $P \wedge Q$ has value T and $R \wedge Q$ has value F. From this it turns out by a further iteration that $(P \wedge Q) \wedge (R \wedge Q)$ has value F.

When this process is repeated indefinitely, it will normally assign truth values to every possible sentence within the language.

One of the most attractive features of a recursive definition is that in order to determine the semantic properties of any particular sentence, it is only necessary to decompose the sentence into its most basic components and then use to recursive definition to calculate the semantic properties of the relevant sentence. Thus, in our simple example, if one wishes to work out the truth value of $(P \wedge Q) \wedge (R \wedge Q)$, one simply decomposes it and notices that it contains the atomic sentences P, Q and R. From the basic definition we can determine the truth value of these sentences, and then use the definition of $\wedge$ to calculate the truth value of $(P \wedge Q) \wedge (R \wedge Q)$.

It has already been shown that the very concept of a model against which all the semantic properties of sentences in a language are defined prevents a completely semantically closed formal truth definition. We will, for the purposes of examining recursivity, ignore this conclusion and look purely at the consequences of assuming a recursive semantic definition. The conclusions reached here will apply beyond model theoretic semantics to any recursive semantics. We will assume that we have the language $\mathcal{LT}$ and the sentences a, b, and c as defined above ($Tr(a)$, $\neg Tr(b)$ and $\neg Tr(b) \wedge \neg Tr(b)$ respectively).

This process of decomposition that can normally be used in a recursive definition, however, does not work when we examine sentences such as a and b. The semantic properties of the sentence a, on a recursive definition, depend on the semantic definition of the Tr predicate and the semantic properties of a. This sets up an immediate circularity, which means that the sentence a cannot be decomposed into atomic sentences, or any sort of basic sentence components which have defined properties. The Liar Sentence b suffers from exactly the same problem - its semantic properties depend on its own semantic properties within a recursive definition. Therefore an ordinary recursive definition cannot assign any semantic properties to Sentences like a and b.

This observation has been often made before, and is the basis of the Kripkean approach to formal truth definitions. Kripke (and Herzberger before him) labeled those sentences which can be decomposed into basic sentence components with defined semantic properties "Grounded" sentences.[4] His method of definition was, on this basis, to allow grounded sentences to be assigned semantic properties according to an ordinary recursive semantics. For ungrounded sentences he defined a third truth value for "ungroundedness", and allowed the recursive definition to continue transfinitely, in order to have a complete definition of it. However, the application of Kripke's definition was limited by what he referred to as "the ghost of the Tarskian hierarchy".[5]

The essential problem that Kripke identified with his definition is that sentences about a Liar Sentence are also ungrounded and therefore on his definition have the same semantic properties

[4] See Hans G. Herzberger. "Paradoxes of Grounding in Semantics". In: *The Journal of Philosophy* 67.6 (1970), pp. 145–167; Saul Kripke. "Outline of a Theory of Truth". In: *Journal of Philosophy* 72.6 (1975), pp. 690–716

[5] Kripke, see n. 9, p. 714

as the Liar Sentence. This problem is one that necessarily follows a recursive approach to the definition of semantics. On this type of definition, the semantic properties of a sentence, such as c, which refers to another sentence will necessarily depend on the semantic properties of the sentence it refers to, in this case b. In fact, the semantic properties of c, that is $\neg Tr(b) \wedge \neg Tr(b)$, have to be purely determined by the properties of $\neg$, Tr, $\wedge$ and b. However, if a recursive definition cannot assign any semantic properties to b, then it equally cannot assign any semantic properties to c. This means that a recursive definition semantics cannot semantically differentiate between a liar sentence and any sentence about a liar sentence, such as c.

This is a serious problem since, according to an intuitive examination of a sentence such as c, its semantic properties cannot be identical to those of b. We will assume that $\wedge$ is defined as in the example above, that is, $A \wedge B$ is true exactly when A and B are each true. This means that, on a recursive definition, c is true, if and only if, $\neg Tr(b)$ is true, i.e. when b is not true. As we have assumed that $\mathcal{LT}$ is consistent, c cannot have the same semantic properties as b. However, as stated, a recursive semantic definition cannot semantically differentiate between b and c so it must assign them the same semantic properties and therefore truth value. Thus a recursive semantic definition cannot capture the logical relationship between c and b, and therefore cannot allow a semantically closed truth definition.

It is important to note neither the sentence c nor the connective $\wedge$ are unique in this respect. There is a long list of sentences which cause the same problem. Some examples include: $\neg Tr(b) \vee \neg Tr(b)$; $(P \supset P) \supset \neg Tr(b)$; and even $\neg Tr(b)$ - with a name other than b (if that is possible). Thus, if we assume that we have a consistent language with a recursive semantic definition, then that language cannot contain a semantically closed truth definition. The assumption of Recursivity, independently of any other assumption about the nature of formal semantics, prevents a semantically closed truth definition.

3.5 Sentences as Truth Bearers

While we have shown that a recursive definition of semantics cannot properly capture the logical relationship between various sentences, there is a weaker assumption that also has similar problems. One of the consequences of assuming a recursive semantics is that every identical string of symbols within the same language necessarily has the same semantic properties and hence the same semantic value. This is because two identical strings of symbols have the same syntactic structure and the same atomic sentence components, and therefore under a recursive semantics must have the same semantic properties. This means, to put it in the language of philosophy of language, that sentences (or sentence types) are the truth bearers if we assume a recursive semantics. This assumption within formal languages is reflected, for example, in the assumption that every sentence has a unique, or canonical, name, and we can use that name exclusively to refer to the sentence. Assuming that sentences bear truth values is a weaker assumption that recursivity, since it is possible to have a non-recursive semantics where sentences bear truth values.

Assuming that sentences are truth bearers, and that every sentence has a singular canonical name, has similar consequences to assuming recursivity for the definition of a truth predicate. In order to demonstrate the consequences of this assumption, we will assume that it is false and show that assuming that sentences are the truth bearers is a non-trivial assumption that has problematic consequences. We will therefore consider the situation where there are two instances of the sentence $\neg Tr(b)$ in the language $\mathcal{LT}$ which we will identify by different names, b and g, and in order to test whether they potentially have different properties. If the assumption that sentences bear truth values does not have any consequences for formal truth definitions or is trivial, then we can unproblematically assume that $b = g$.

However, when we consider the definitions of b and g, then it should not be expected that we can assume $b = g$ without suffering any adverse consequences. If we consider the instance of the sentence $\neg Tr(b)$ with the name b, then it is obviously a classic liar sentence. However, if we consider the sentence $\neg Tr(b)$ with the name g in isolation from the definition of b, then it is a perfectly unproblematic sentence whose truth value depends on the truth value of b (whatever that is). Thus the reference structure of these two sentences is different, since one sentence is self-referential and the other is not, even though they are different instances (or tokens) of the same sentence. This difference leads a significant problem, which demonstrates that a formal language for which sentences are truth bearers cannot be semantically closed.

We will first consider the sentence $\neg Tr(b)$ with the name g. By definition, g should be true if, and only if, the sentence b (whatever it is at this point) is not true. However, if one assumes sentences are the truth bearers, then $b = g$ and b and g must should have identical semantic properties. Given that in a consistent semantically closed formal language, exactly one of $Tr(b)$ and $\neg Tr(b)$ is true, it follows, as above, that a language in which sentences bear truth values cannot consistently include a truth predicate and be semantically closed. This is the identical problem to the situation for recursivity.

This fact, that any formal language which assumes that its semantics is recursive or that assumes that sentences bear truth values cannot be semantically closed, partly explains why semantic closure has not been achieved in any formal definition that currently exists. For every existing truth definition assumes at least one of these principles, with the majority of definitions assuming recursivity and hence both. The question that this raises is what a non-recursive semantic definition for a formal language in which sentences do not bear truth values could look like.

3.6 Summary

The existence of systematic limitations in modern formal truth definitions as shown in Chapter 2 has turned out to be unsurprising, given that all of these definitions assume a standard recursive, model theoretic semantic definition. Both the assumption of recursivity, and the assumption that truth is to be defined with respect to a model inherently limit what can be achieved with a formal truth definition. Interestingly, the reasons that each of these assumptions cause the limitations is independent, so we have two separate problems that need to be addressed.

For this reason, consideration of the method that can resolve these problems will be split into two parts. The first part, addressing the problems within Model Theory, will cover chapters 4 and 5. Chapter 4 will look at how to address the problem and chapter 5 will provide a formal definition to show how it can be done. The consideration of recursivity and the assumption that sentences are truth bearers will be left to chapter 6. It is a less complicated, and more controversial change, and builds on the work in chapters 4 and 5. Given the two problems are separate, the solutions are also separate and each can be implemented without the other.

Chapter 4

Outline of a New Approach

As noted before, a philosophically satisfactory formal solution to the Liar Paradox requires the definition of a logical language which can satisfy the grammatical conditions of natural languages. That is, the logical language must include its own truth predicate, and there are no restrictions on applying this predicate to the sentences in the language. Moreover, a philosophically satisfactory formal solution should satisfy the condition of Semantic Closure, that is, for any sentence in the language, it is possible to construct sentences which assert that the relevant semantic properties (or at least truth values) for the original sentence hold and that these sentences in turn have the correct truth values. In the previous chapter, it was demonstrated that adopting a standard Model Theoretic semantics for a formal language is inconsistent with semantic closure. The structure of standard Model Theory prevents the satisfactory evaluation of the truth value of certain sentences within the relevant language. A philosophically satisfactory formal solution to the Liar Paradox is therefore not possible if we accept a standard Model Theoretic semantics.

Usefully, two key assumptions were identified in the previous chapter that individually prevent a formal language with a standard Model Theoretic semantics being semantically closed. The first assumption was that the semantic properties of all the sentences within a language are defined directly against the semantic model. The second assumption was that sentence types are truth bearers, or that identical syntactic entities have identical semantic properties, which is necessary for the standard recursive definition of semantic properties. Given that each of these assumptions prevents semantic closure by themselves when we are dealing with the Liar Paradox, a satisfactory solution to the Liar Paradox is only possible if our understanding of the semantic structure of formal languages can be changed.

The two assumptions that need to be questioned are independent of each other, and therefore need to be investigated separately. The second assumption, that sentence types are truth bearers, has already been challenged in the literature on the Liar Paradox,[1] with varying degrees of success.

[1] For example, see Brian Skyrms. "Definitions of Semantical Reference and Self-Reference". In: *Notre Dame Journal of Formal Logic* XVII.1 (1976), pp. 147–148; Laurence Goldstein. "'This Statement is Not True' is Not

Moreover, as argued in the Introduction, changing the account of the correct truth bearer does not address the problems in differentiating between the paradoxical and non-paradoxical, which is crucial to solving the Liar Paradox. For these reasons, discussions of this assumption will be left to Chapter 6.

The focus of this chapter is on the structural assumption within Model Theory that the semantic properties of all sentences are defined solely with respect to the relevant Model. Given that this assumption has been shown to be incompatible with a philosophically satisfactory solution to the Liar Paradox, the aim of this chapter will be to develop an alternative understanding of the semantics of formal languages that avoids this assumption. Once this is achieved, the formal implementation of this alternative understanding will be explored. It should be noted that just as the relevant assumption is independent of any particular flavour of modern logic, the alternative approach developed will also be independent. This means that, if successful, the new approach and the resulting truth definition will be applicable to any formal system of logic and any formal language.

The development of the approach will also follow an important principle, namely that since the Liar Paradox does not directly affect the vast majority of sentences within a language, any solution should directly affect as few sentences as possible. This means that the approach adopted will be to refine the standard Model Theoretic approach to semantics, rather than to entirely replace it. This is not meant to represent any deep philosophical commitment to the correctness of Model Theoretic Semantics as the correct semantics for formal or any languages. Instead it is more conceptually illuminating, and easier formally, to see where the standard approach needs to be changed, rather than attempting an entirely new approach.

The strategy for developing an alternative approach, and an alternative truth definition, is to begin by considering what the concept of Truth should actually represent in a formal language. From this we can consider the nature of the truth predicate and therefore how to understand the structure of truth and the definition of the truth predicate.

4.1 Truth in Formal Languages

The concept of Truth is one of the more controversial concepts in the history of philosophy, and it is beyond the scope of this project to resolve the debates about the nature of Truth or the correct theory of Truth. Our focus is more narrow: the concept of Truth in formal languages, and how we are to understand and work with this concept. This narrowing of focus reduces the scope of the problem, since what we are interested in this context is what it is for a sentence to be true. While in one sense this problem cannot be answered without an answer to the broader questions

True". In: *Analysis* 52.1 (1992), pp. 1–5; Alan Weir. "Token Relativism and the Liar". In: *Analysis* 60.2 (2000), pp. 156–170 about what truth is, there is a formal or structural sense in which this problem can be addressed independently of any Theory of Truth. That is, we can develop a clear sense of certain structural facts about truth that are neutral with respect to the large philosophical debates about truth.

4.1.1 What does it mean for a sentence to be true?

A sentence, read purely as string of symbols, does not in itself possess any truth value. It is only when the string of symbols has, or is assigned, a meaning that it makes sense for a sentence to have a truth value. The meaning of a sentence is a major determining factor in the truth value of a sentence, as it provides truth conditions for the sentence. That is, the meaning of a sentence determines conditions under which the sentence will be true.

A sentence is therefore true whenever its truth conditions (as determined by the meaning) hold, or are the case. What it exactly means for truth conditions to hold relates directly to the broader questions of Truth which are beyond this work. However, the structure outlined here requires that there is something against which the truth conditions of a sentence can be tested to determine whether they hold or not. In order that the concept of truth be meaningful and not circular, this something must be independent of the sentence in question. There must be something outside the sentence (and perhaps the language) against which the truth of the sentence is determined.

Thus, for example, the truth of “It is raining outside” is tested against the current external weather conditions; the truth of “Bachelors are married men” is tested against the system of meanings of English words; and the truth of “2+2=4”, depending on one’s account of mathematical truth, is tested again an external mathematical reality, a socially constructed definition, or its derivability from a set of self-evident axioms. The truth of a sentence is always tested against something external to the sentence itself, and this thing which determines truth values is usually external to the language the sentence is expressed in.

Model Theoretic Semantics offers a formalisation of this principle, with the simplifying assumption that what the truth of sentences is tested against is completely external to the language. In this case, if we have a formal language $\mathcal{L}$, with a model $\mathcal{M}$, and a sentence p in $\mathcal{L}$ is true exactly when it is the case that $\mathcal{M} \models p$. That is, a sentence p in $\mathcal{L}$ is true exactly when its meaning (interpretation) generates truth conditions that are decidable in $\mathcal{M}$ and moreover they hold in $\mathcal{M}$. The model $\mathcal{M}$ is by definition external to the language $\mathcal{L}$, which illustrates the way that Model Theoretic Semantic is a particular formalisation of this fundamental principle that the truth of sentences is determined (via the sentence’s meaning) by something external to the sentence.

4.1.2 The Truth Predicate

This fact that the truth of a sentence is decided by whether the truth conditions of that sentence, as determined by the meaning, hold or not against something external to the sentence allows us to explore the necessarily properties of any satisfactory truth predicate. A satisfactory truth predicate must reflect this fact in the way it is defined, otherwise it cannot correctly be a truth predicate.

Now, say we have a sentence a in some language, if a is true then the truth conditions given by the meaning (or interpretation) of a must hold. Thus, asserting that "a is true" asserts a particular type of relationship between the sentence and something external to that sentence. Exactly how we are to understand this relationship of the truth conditions of a sentence 'holding' is not a matter that will be investigated here. Different theories of truth will give different answers to this.

However, this gives a clue to the problems faced by attempts to define a truth predicate in formal language. The sentence a is a syntactic entity, and on the normal approach to the semantics of formal languages, the external thing which determines truth values is the Model. The assertion that "a is true" is therefore the assertion of a particular type of relationship between an element of the language and the Model. To be more precise, the assertion that "a is true" would normally be understood as the assertion that the interpretation of the sentence a in the Model holds. From a structural point of view, one would therefore think that the assertion that "a is true" could therefore only be made in a language which includes both the model and the original language. The attempt to define the truth predicate in the original language, as formal truth definitions typically do, seems therefore to attempt to assert the relationship between two things, one of which (the Model) is not expressible within the language.

This problem is not fatal to the project at hand. Indeed, if it were fatal then the use of the truth predicate in natural languages would also be doomed, as we normally measure the truth of sentences against things that are not directly representable in language. However, it shows that we must understand the truth of sentences which include a truth predicate differently to normal sentences. We can see this in natural languages, where we approach assertions such as "a is true" and their truth value differently to other sentences. For example, let a be the sentence "The kettle is hot". In order to evaluate the truth value of a, we identify the conditions under which a would be true, namely the relevant kettle is hot, as evidenced by a number of possibilities such as the steam rising out of it, or that it feels hot if we put our hand on or near it. We can then test to see whether these states of affairs exist which enables us to judge whether the truth conditions of a hold, and therefore to evaluate the truth value of a.

In evaluating the truth value of the sentence "a is true", however, we cannot directly test its truth conditions against a relevant external thing. For the truth conditions of "a is true" are that the truth conditions of a hold. We must therefore first evaluate the truth value of a, and then use this to evaluate the truth value of "a is true". The evaluation of any sentence of the form "a is true" is therefore a two step process, we must first evaluate the truth value of the sentence a and then we can evaluate the truth value of "a is true".

The existence of this two step process reveals a problem with standard Model Theoretic Semantics. Definitions of Model Theoretic Semantics normally assume that every sentence has an interpretation in the Model, and therefore that the evaluation of the truth value of every sentence occurs directly in one step against the Model. The evaluation of sentences such as "a is true" does not function in this way, as there is the two step process, which is not possible under the normal assumptions of Model Theory.

4.1.3 Understanding the Liar Paradox

This two step process is crucial for understanding the Liar Paradox, for the existence of this extra step creates the 'space' in which the Paradox works. In essence the Liar Paradox occurs because there are certain sentences for which this two step process can never complete.

A requirement on the definition of a truth predicate is that there is some means of referring to sentences within the scope of the truth predicate. Typically in a formal setting names of sentences are used. As soon as there is this difference between a name (or other means of referring) and the sentence it identifies, the combinatorics of the situation mean that it is possible that there are sentences which include their own name, or at least sentences which refer (by means of names) to another sentence which includes the name of the original sentence. This possibility of circularity is equally possible if we use descriptions or some other means of referring to sentences.

This circularity is, however, fatal when it occurs in the scope of the truth predicate. Suppose we have two sentences, b and c, and b includes "c is true" as a part and c includes "b is true" as a part. According to the two step process outlined above, the truth value of b depends on evaluating the truth value of c. However, the truth value of c depends in turn on evaluating the truth value of b. Evaluating each sentence depends on being able to move to the second step of the process for the other sentence and therefore evaluating the other sentence. However, because there is this two step process for the truth predicate, and because of circularity of reference, this is an example of a situation where the evaluation never reaches the second step in the process. No sentence whose truth value is evaluable can be found, and the two step process fails. The truth value of these sentences cannot be evaluated.

This breakdown of the two step evaluation process occurs in the case of the Liar Paradox, and explains the way that any consideration of the Liar Paradox starts heading round in circles. The situation in the case of the Liar Paradox is more pathological, since not only does this two step process never complete, but the truth conditions generated at different steps in the process contradict each other. Both of these factors demonstrate that there is no way of definitively evaluating the truth value of a liar sentence through the normal two step process for sentences which include a truth predicate.

4.1.4 What does it mean for a sentence to be *not* true?

Although the standard process for evaluating the truth value of sentences which include a truth predicate breaks down in certain cases, it does not mean that we cannot say anything about the truth value of such sentences. For the breakdown of this process demonstrates definitively that these sentences cannot be true. There is nothing external to the sentence which verifies the truth conditions of the sentence. Instead the truth conditions of the sentence lead us into an infinite loop which can never be escaped. Thus although sentences such as b and c are not evaluable on the standard two step process for sentences which include the truth predicate, this fact that they are not evaluable in this sense means that we can conclude that b and c are not true.

This means that there are two different ways that we would judge a sentence to be *not* true. A sentence is obviously not true if its truth conditions do not hold. Thus for example a is not true if the kettle is not hot and which could be verified by me touching the relevant kettle in my bare hands and my hands not feeling hot or getting burnt. However, the case above involving the truth predicate shows that there are cases where the truth conditions of a sentence do not allow us to test the truth of the sentence, as nothing testable is ever identified.

This conclusion is of fundamental importance in tackling the Liar Paradox. Since there are multiple ways in which a sentence can fail to be true, any formal truth definition needs to represent these multiple ways. The standard understanding of truth, the T-Schema, only allows for sentences to be not true if what they say is not the case. That is, the T-Schema only allows for the first way of sentences not being true identified above. It does not allow for the failure of the standard evaluation process for sentences which include a truth predicate. As the Liar Paradox turns exactly on this type of situation, the T-Schema needs to be revised if a satisfactory formal truth definition is to be achieved.

4.1.5 The Strengthened Truth Schema

It has just been shown that there are at least two ways in which a sentence can fail to be true in a formal language which includes a truth predicate. A correct formal definition of truth must correctly reflect these different ways that it is possible for a sentence not to be true, however, the T-Schema cannot. The T-Schema, that 'p' is true iff p, provides a single condition on the truth of a sentence, and therefore a single condition on a sentence being not true. The Liar Paradox demonstrates clearly that, in at least special limit cases, there are at least two ways in which a sentence can fail to be true. The second condition is that the sentence fails the standard truth value evaluation procedure. For the sake of identifying this easily, we will say that such sentences are not evaluable.

While we have sketched an intuitive account of what Evaluability represents, namely that the normal truth value evaluation procedures, we have not given any precise definition. This will be investigated in a later section, as here we are only interested in exploring the use of Evaluability as a second condition on the truth of a sentence. As argued above, this is necessary to deal with the Liar Paradox, and the T-Schema needs to be strengthened to include this extra condition. Adding this condition in gives the following Strengthened version of the T-Schema:

Strengthened T-Schema: ‘P’ is true iff (P and ‘P’ is evaluable)

That is, a sentence P is true in exactly the cases when P (what P says holds) and P is evaluable. Or to translate it into a formal language $\mathcal{LT}$, we assume that there is a new predicate $Eval\ulcorner P\urcorner$ which identifies the evaluable sentences and get the following:

Str. T-Schema: $Tr\ulcorner P\urcorner \equiv (P \wedge Eval\ulcorner P\urcorner)$

This Strengthened T-Schema captures very neatly the idea that there are two ways that a sentence can fail to be true, and therefore that there are two conditions which need to be satisfied for a sentence to be true. If we treat this as a definition, then a sentence P is true exactly when P and P is evaluable. If either of these fail, then it is not true. As a note on the notation adopted here, we have used capital letters such as P to represent sentences and lower case letters will represent names. The $\ulcorner\urcorner$ notation here is, as is standard, a name-forming device.

While this Strengthened T-Schema has been developed in response to a conceptual need, it has some very elegant formal properties. Firstly, it reduces to the normal T-Schema in the case when the sentence is evaluable. If it is the case that “P is evaluable”, then the Strengthened T-Schema is equivalent to “ ‘P’ is true iff P”. Given that ordinary sentences are assumed to be evaluable, the Strengthened T-Schema fulfills the criteria of minimal mutilation, since it reduces back to the normal T-Schema in all unproblematic cases.

Secondly, if evaluability is appropriately defined, it provides a very neat method of formally resolving the Paradox. We will consider an archetypal formal liar sentence, say the sentence $\neg Tr(b)$ with the syntactic name b, as an illustration. We will assume that $Eval(p)$ is defined so that $\neg Eval(b)$ is a theorem for b and that these sentences are defined in a Classical Logic. The Str. T-Schema for b is therefore:

Str. T-Schema for b: $Tr(b) \equiv (\neg Tr(b) \wedge Eval(b))$

However, since we have assumed that $\neg Eval(b)$, the following derivation holds:

1	$Tr(b) \equiv (\neg Tr(b) \wedge Eval(b))$	Str T-Schema
2	$\neg Eval(b)$	Assumption
3	$\neg(\neg Tr(b) \wedge Eval(b))$	Classical Derivation
4	$\neg Tr(b)$	Modus Tollens lines 1,3

That is, if $Eval$ is defined appropriately, it follows from the Strengthened T-Schema that $\neg Tr(b)$. However, this does not license the conclusion that $Tr(b)$, which occurs if we assume the standard T-Schema. On the standard T-Schema, $Tr(b) \equiv \neg Tr(b)$ once we have proven that $\neg Tr(b)$, it follows immediately that $Tr(b)$. However, on the Strengthened T-Schema, we can only conclude $Tr(b)$ if both $\neg Tr(b)$ and $Eval(b)$ hold. Only one of these hold, and therefore the standard argument to the contradictory conclusion is blocked.

To put it differently, a sentence of the form $A \equiv (\neg A \wedge B)$ can be consistently true in classical logic if both A and B are false. That means that the Strengthened T-Schema for b [$Tr(b) \equiv (\neg Tr(b) \wedge Eval(b))$] can be consistently true in classical logic if both $Tr(b)$ and $Eval(b)$ are false, which is what follows from our definition. While this does not prove that a formal system which includes the Strengthened T-Schema as a truth definition is consistent, it shows that it is classically consistent with an archetypal liar sentence. This is a remarkable result, that a simple truth definition such as the Strengthened T-Schema allows a classically consistent solution to at least archetypal liar sentences.

The flexibility of the Strengthened T-Schema can be further demonstrated by the following derivation. Suppose we have a formal language which has a Truth predicate and an Evaluability predicate, but the *Eval* predicate has not (yet) been defined. That is, we cannot say of any sentence whether it is Evaluable or not. Despite this, if we use the Strengthened T-Schema as a truth definition for the sentence b as defined above, the following holds in a standard natural deduction system for classical logic:

1	$Tr(b) \equiv (\neg Tr(b) \wedge Eval(b))$	Str. T-Schema
2	$\quad Tr(b)$	assume
3	$\quad \neg Tr(b) \wedge Eval(b)$	⊃E 1,2
4	$\quad \neg Tr(b)$	∧Elim 3
5	$\quad Tr(b) \wedge \neg Tr(b)$	∧Intro 2,4
6	$\neg Tr(b)$	¬ Intro 2-5
7	$\neg(\neg Tr(b) \wedge Eval(b))$	Modus Tollens 1,2
8	$\neg\neg Tr(b) \vee \neg Eval(b)$	DeM 3
9	$Tr(b) \vee \neg Eval(b)$	Double Negation 4
10	$\neg Eval(b)$	Disj. Syll 2,5

That is, even if *Eval* is not defined, it follows from the Strengthened T-Schema that both $\neg Tr\ulcorner b\urcorner$ and $\neg Eval\ulcorner b\urcorner$. The definition of *Eval* is not crucial for deriving this conclusion. Moreover, so long as the definition of *Eval* is such that it agrees with the Strengthened T-Schema on which sentences are not Evaluable, we can expect that the resulting system would be consistent. This means that within a particular language, if every sentence which is paradoxical upon the introduction of the T-Schema is not evaluable, then the introduction of the Strengthened T-Schema will ensure that the system remains consistent. This in turn provides a very simple criterion on the definition of a satisfactory evaluability predicate, namely that every normally paradoxical sentence is not evaluable. That is, if we define truth using the Strengthened T-Schema, and all of the ‘normally paradoxical’ sentences are not evaluable, the definition will be consistent.

These results demonstrate that the conceptual conclusion that there needs to be two conditions on the truth of a sentence is very promising from a formal sense, and promises the possibility of a much improved formal truth definition. In particular, the Strengthened T-Schema can potentially provide a classical, consistent definition of a truth predicate within its own language. The effectiveness of such a definition depends, however, on a satisfactory definition and explanation of the concept of Evaluability that is being applied.

4.2 Evaluability

The previous section has shown that introducing the concept of Evaluability as a second condition on the Truth of sentences is a promising formal approach to the Liar Paradox. Moreover this concept was introduced from a conceptual analysis of the concept of Truth and Truth Predicates, not merely as a formal device. This concept could therefore play the key role in a satisfactory formal truth definition. Before this is attempted, however, the exact concept needs to be clarified, particularly as the concept of a sentence *not* being evaluable has been introduced more clearly than the concept of Evaluability.

The basic reason for introducing this concept is clear. It was observed that the process for evaluating the truth value of sentences which contain a truth predicate is different to other declarative sentences. The process of evaluating the truth value of a sentence which contains a truth predicate requires us to first evaluate the truth value of the sentence which is being referred to in the scope of the truth predicate, and then evaluate the original sentence. As pointed out before, there are sentences for which this process fails to complete. Given that the standard process for evaluating the truth value of such sentences fails, it follows, in some sense at least, that these sentences are not evaluable.

However it is not clear how to precisely understand what it means for a sentence to not be evaluable. For, as argued above, we can say something definite about the truth value of sentences which are not evaluable, they are definitely not true. But presumably that would make them evaluable because we can now evaluate their truth value.

There are two different concepts of evaluability at play here, and they need to be carefully differentiated. In the first case, we were discussing evaluability in the sense of whether the *standard process* for evaluating the truth value of the sentence succeeds. By standard process, we mean the process for evaluating the truth value of a sentence as determined by its truth conditions. In the case of sentences which contain a truth predicate, their truth conditions depend on the truth value of another sentence or other sentences. The standard process for evaluating these sentences therefore involves first determining the truth value of the other sentence(s), which is not possible in certain cases. These sentences are not evaluable in the sense that their truth conditions are not evaluable. Thus the first sense of evaluability is evaluability of a sentence's truth conditions.

In the second case, the concept of evaluability at play is the question whether it is possible to work out the truth value of a sentence at all, whether or not the truth conditions for that sentence are evaluable. Thus, for sentences such as Liar Sentences, their truth conditions are not evaluable, yet this means that they must be not true, and so they are evaluable in this second, more general sense.

Every meaningful sentence is evaluable in this second sense. For if a sentence is meaningful, then it has determinate truth conditions. If it has determinate truth conditions, then it must be the case that either the truth conditions can be evaluated or they can not. If they can be evaluated, then the sentence has a truth value, and the sentence is evaluable in the second sense. If the truth conditions cannot be evaluated, then the sentence is not true, and the sentence is again evaluable in the second sense.

As far as the Strengthened T-Schema, and any formal definition of truth, is concerned, it is the first sense of evaluability that is relevant. So to be more precise, the Strengthened T-Schema should read: 'p' is true iff (p and the truth conditions of 'p' are evaluable). Since the ordinary declarative sentences that we are mostly concerned about are assertions about facts, it is not possible that the truth conditions of these sentences can be not (at least in principle) evaluable.[2] The only sort of sentences we have found which may have truth conditions that are not evaluable are sentences which contain a truth predicate. While it may be possible in other contexts that there are other types of sentences whose truth conditions are not evaluable, the focus of this project is on the Liar Paradox and we will therefore only consider its application to the truth predicate and ordinary declarative sentences. We therefore need to identify the conditions under which the truth conditions of a sentence of these types are or are not evaluable. Since this is the only concept of evaluability that is relevant for the following discussion, unless it is made clear otherwise, the concept of evaluability being applied below is equivalent to the truth conditions being evaluable.

4.2.1 Groundedness

As discussed so far, the concept of Evaluability is very similar to the concept of Groundedness, as identified by Herzberger[3] and Kripke[4]. This is the concept that sentences which contain a truth predicate are grounded if they eventually refer only to sentences which do not include a truth predicate, that is to sentences expressing facts. The motivation behind this is that it is the facts that determine the truth values of sentences, and therefore sentences which refer (eventually) to

[2] As a matter of fact, it may not be possible for humans to evaluate them. However, they are in principle evaluable.

[3] See Herzberger, "Paradoxes of Grounding in Semantics", see n. 14

[4] See Kripke, see n. 9

facts must have a truth value. That is, to put it in terms of Evaluability, the truth conditions for sentences which refer (eventually) to facts are evaluable, and therefore these sentences must have a truth value. Grounded sentences are therefore evaluable sentences.

Herzberger and Kripke, however, go further and claim that grounded sentences are the only sentences that can have a truth value. This means, in terms of evaluability, that sentences which refer (eventually) to facts are the only ones whose truth conditions are evaluable. This is a significant claim that does not follow from the fact that grounded sentences are evaluable, and is not justifiable.

For a sentence "a is true", its truth conditions require us to first determine the truth value of a in order to determine the truth value of "a is true". If the only way that a can have a truth value is that it eventually refers only to facts, then the Herzberger/Kripke position on Groundedness is justified. However, we identified a sentence above which does not refer to any facts but which was identified as not true - the Liar Sentence. Given that the Liar Sentence is not true, the truth conditions of the sentence "The Liar Sentence is not true" are evaluable. The Liar Sentence has a truth value and therefore we can work out the truth value of "The Liar Sentence is not true". It is in fact true. This means that there are evaluable sentences which are plausibly not grounded.

This occurs since, although the Liar Sentence's truth conditions are not evaluable, the sentence itself is evaluable in the broader sense of evaluability outlined above, and therefore has a truth value. This means in turn that the truth conditions of the sentence about the Liar Sentence are evaluable, and therefore this sentence about the Liar Sentence is evaluable even though it is not grounded. Grounded sentences are therefore a subset of Evaluable sentences.

4.2.2 When is a sentence not evaluable?

Given the role of the Evaluability predicate in the Strengthened T-Schema, the most crucial part of defining the Evaluability predicate is ensuring that it identifies the correct sentences as not evaluable. This means that we need to be able to identify more precisely the sentences whose truth conditions are not evaluable. As noted above, for this project we are assuming that sentences which contain a truth predicate are the only sentences which can be not evaluable in this sense.

The core feature of the truth conditions of sentences which contain a truth predicate is that they generate a two step process. For a sentence of the form "a is true", we must first evaluate the truth value of a and then we can evaluate the truth value of "a is true". This means that we must run a truth value evaluation process for a first, and then separately run the process for "a is true". Running the process for a gives us three possible types of conclusions: i) the truth conditions of a are evaluable and a has truth value Z; ii) the truth conditions of a are not evaluable and therefore a is not true; or iii) cannot tell if the truth conditions of a are evaluable and have to run the process for some other sentence(s). In the first two cases, we have some determinate information about the truth of a and therefore the process for "a is true" will succeed. The truth conditions of a in these two cases are therefore evaluable.

In the third case, upon repeating the process possibly multiple time we will normally eventually only reach sentences that fall into either i) or ii), and therefore this information will transmit back up the line and "a is true" will be evaluable. However, it is possible that it will turn out that "a is true" is not evaluable. We will look at the situations in which a sentence will turn out to be not evaluable next.

A sentence is not evaluable if its truth value evaluation process cannot successfully complete. Given that among the sentences we are considering, this can only occur for sentences which include the truth predicate, we can focus on these sentences. The evaluation process for these sentences involves progressing to the sentence(s) referred to and evaluating the truth value of these sentences. As argued above, when we move to these other sentences, we will either get a definite truth value of some sort, or we will be required to move to further sentences. This pattern of a sentence referring to other sentence(s) which refers to further sentence(s) sets up what we will refer to as a chain of reference.

A chain of reference will halt if a sentence in the chain does not refer to any further sentences, that is if it is a sentence which does not include a truth predicate. This means that the chain of reference for a particular sentence will be finite if it eventually refers to only sentences which do not include truth predicates, that is to sentences about facts. That is, if the chain of reference for a particular sentence is finite, then that sentence is grounded.

Now if the chain of reference for a sentence is finite, the evaluation process will succeed immediately for those sentences on the end of the chain, and will therefore all of the sentences in the chain will be evaluable. Sentences that are not evaluable must therefore have, at a minimum, infinite chains of reference.

In an infinite chain of reference, there must be either a finite number of sentences infinitely repeated, or an infinite number of sentences. If there is an infinite number of sentences, the situation must obviously include something like the following:

S_1	S_2 is true.
S_2	S_3 is true.
S_3	S_4 is true.
$\vdots$	$\vdots$

In this case, none of the sentences can be evaluable, since the evaluation process can obviously never halt. There is no way to determine a truth value for these sentences by following the chain of reference, as the truth conditions on each sentence require us to do.

If there are a finite number of sentences infinitely repeated, then there must be a loop in the chain and the chain continually repeats each sentence in the loop. The reason for this is that for any repeated sentence, that sentence must refer to the same sentence(s) every time it appears in the chain. Therefore the section of chain after a repeated sentence must be identical after every time it repeats. It follows that there must be fixed loops in the chain that keep reoccurring. The following is an example:

T_1 T_2 is true.

T_2 T_3 is not true.

T_3 T_1 is true.

The chain of reference in this case will follow the pattern: $T_1, T_2, T_3, T_1, T_2, T_3, ...$ indefinitely. Obviously in this case, none of these sentences can be evaluable as there is no way of following the chain of reference and getting outside of the loop. Thus sentences which belong to a loop of reference like this are also not evaluable. Liar Sentences belong to this category.

There is, however, another possibility (aside from combinations of these two) which will generate an infinite chain of reference. For example, we can take the previous example, and add a sentence which is not part of the loop, but refers to one of the sentences in the loop:

U T_2 is not true.

T_1 T_2 is true.

T_2 T_3 is not true.

T_3 T_1 is true.

The chain of reference which starts at U will be infinite, but in this case, it does not mean that U is not evaluable. If we take the evaluation process for U, we must determine the truth value for T_2. When we run the evaluation process for T_2 we can see that T_2 belongs to a circular loop of reference, and therefore T_2 is not evaluable and hence not true. This however means that the evaluation process for U can succeed, and therefore that U is evaluable and in fact true.

We have therefore identified two situations in which a sentence is not evaluable, and each of these depend purely on the chain of reference of the sentence in question. The first is that the chain of reference generates a loop which the original sentence is a part of, and the second is that the chain of reference is an infinite chain which has infinitely many unique sentences in it. There is a major computational difference between these two situations which is relevant to what follows. In the first case, one needs to only go through a finite number of steps until one reaches the original sentence in the chain. In the second case, there is an infinite chain and the type of finite computation available for the first step is not possible. One needs to be able to reason through quantification and/or induction to be able to conclude that this case holds. These difference affect how evaluability can be defined in different formal languages.

4.3 Application to Formal Languages

We have developed a conceptually justifiable approach to the definition of Truth in the Strengthened T-Schema which depends on the concept of Evaluability which has been clarified in the previous section. The challenge now is to apply these to our understanding of the semantics of formal languages, and to define them formally in a way that captures the concepts as set out above. Fortunately there are no major obstacles to achieving this.

The key observation in achieving this is that the distinction between evaluable and not evaluable sentences depends, for the types of sentences we are interested in, purely on the chains of reference of sentences. That is, if we know what sentences refer to which other sentences, it will be possible to compute Evaluability. This information about reference is, at least in formal languages, normally defined as part of the language syntax, and there are therefore no conceptual barriers to representing this information in the relevant syntactic language. The Strengthened T-Schema, moreover, is easily definable as an axiom in a formal language, if Evaluability is definable in the language. So there are no large technical barriers to achieving this type of definition, and more details will be discussed in the next section.

There is, however, the remaining conceptual problem of how to understand the semantics of formal languages which include a truth predicate. The first point is to note that, as argued above, the truth conditions of sentences which include a truth predicate are different to ordinary declarative sentences. The truth conditions of sentences which include a truth predicate relate to whether the truth conditions of another sentence are satisfied, not to whether some state of affairs holds. It is therefore reasonable to expect that the formal semantics for these sentences are different to ordinary sentences.

Moreover, for all sentences which do not include a truth predicate, we have not identified any good reasons for abandoning the standard model theoretic semantics, so it should be kept for these sentences. For sentences which include a truth value, on the other hand, we have shown that their semantic value depends on both the semantic value of the sentence(s) referred to and whether or not the sentence is evaluable. Since evaluability is a property that depends on reference which is defined within the language in formal languages, the truth value of the sentences which include a truth predicate cannot be determined by a model in the standard sense. Normally, models do no contain any information about what sentences in the language refer to other sentences

This leaves two options, the model must be extended to include information about sentences in the language and what sentences refer to what other sentences, or sentences which include a truth predicate must remain uninterpreted and therefore are not measurable against the model. Thus, on the first option, facts about what sentences refer to what other sentences must be included in the model, and therefore the evaluability of sentences is testable against the model. On the second option, the facts about what sentences to refer to other sentences is purely encoded in the syntax of the language and therefore evaluability and truth (as defined by the Strengthened T-Schema) are properties that are not interpretable in the model as the model does not contain the relevant information.

Both approaches are justifiable, and the choice does not impact the definition of the syntax. Moreover, on both options the truth and evaluability predicates are not interpretable in the standard model for a language. On the first option, these predicates are interpretable within a model which comprises the standard model plus extra information about the reference structure of the language. On the second option, these predicates are simply not interpretable. As noted, the choice between these does not impact on the definition within the language, or the properties of the definition within the language. Therefore no definitive choice will be presented here. The key point, on both options, is that the truth and evaluability predicates are not interpretable on the standard models for formal languages.

4.3.1 Defining an Evaluability Predicate

The analysis of Evaluability above identified two conditions under which a sentence is not evaluable, either the sentence is part of a circular reference loop, or part of an infinite reference chain that never loops. While both of these conceptually lead to a sentence not being evaluable, only one of these is involved in the Liar Paradox - the circular reference loop. So long as we do not have quantification over infinite sets of sentences, as in Yablo's Paradox, the case of an infinite reference chain is not paradoxical. We will therefore focus on defining an Evaluability Predicate that deals with circular reference loops, and will leave the infinite case to one side. This has the major technical advantage of allowing a truth definition that does not require quantification, which broadens the applicability of the definition. The infinite case can obviously be included in languages which include quantification and its basic definition will be discussed here.

The primary focus of the definition of Evaluability is whether a sentence is part of a circular reference loop. In order to use this concept for a formal truth definition, we need to be able to test whether this is the case within the relevant system of reasoning. In existing formal logics, it is not possible to test this, since their definition does not allow both the necessary information to be expressed and reasoning about this information to occur within the language. As a historical note, this fact that existing formal languages are not capable of dealing with this type of reasoning is a little curious. It is widely acknowledged that circular (or self) reference is one of the causes of the Liar Paradox, yet there has been little effort put into being able to represent this information as part of a formal definition of truth. Nevertheless, the necessary concept of reference is intuitively simple and it is not difficult to provide a precise formal definition.

4.3.1.1 Direct Reference

To illustrate this, we will use an arbitrary language $\mathcal{L}$, which includes sentences $A, B, C....$, and a standard notation to identify names of those sentences within $\mathcal{L}$: $\ulcorner\urcorner$. These brackets can be read either as a name-forming device, in the way that quotation marks are used in English, or as a meta-theoretic notation which stands for whatever the actual name of the relevant sentence in the object language is. In order to develop a definition of reference, we suppose that the language $\mathcal{L}$ includes a number of predicates which take the names of sentences as their arguments, say $P\ulcorner\urcorner$, $Q\ulcorner\urcorner$, $R\ulcorner\urcorner$.... We can then say that a sentence, A, *directly refers* to another sentence, B, exactly when A includes at least one of the predicates $P\ulcorner\urcorner$, $Q\ulcorner\urcorner$, $R\ulcorner\urcorner$... as a part and an argument of that predicate is the name $\ulcorner B\urcorner$. That is, A, *directly refers* to B, exactly when one of A includes at least one of $P\ulcorner B\urcorner$, $Q\ulcorner B\urcorner$ or $R\ulcorner B\urcorner$etc as a part of the sentence A.

To give a short example to illustrate this definition: Say we interpret $R\ulcorner\urcorner$ as meaning that " is an atomic sentence", then the sentence "B is not an atomic sentence and it is snowing" directly refers to B, while the sentence "It is snowing and the grass is frozen" does not. This matches what we mean by reference, and importantly for formal definitions, can be determined from the syntax of the sentence.

A number of useful facts follow from this definition. Firstly, there will be many sentences that do not directly refer to any sentence, at a minimum all of the sentences which do not include one of the relevant predicates. As in our example, "It is snowing and the grass is frozen" does not directly refer to any sentence, as no sentence is mentioned within the sentence. Secondly, even if the sentence includes a relevant predicate, direct reference is not generally reflexive or transitive. So for example, if A is the sentence "B is not an atomic sentence" and B is the sentence "C is an atomic sentence", A directly refers to B and B directly refers to C, but A does not directly refer to C. Thirdly, a sentence may directly refer to many other sentences, and if the language allows predication over names, then it may directly refer to an infinite number of other sentences. Fourthly, a standard Liar Sentence such as $\neg Tr\ulcorner B\urcorner$ with the name B directly refers to itself. This can be seen since $Tr\ulcorner\urcorner$ is a predicate which takes names of sentences as arguments, and $Tr\ulcorner B\urcorner$ is a part of B. Therefore B directly refers to B.

The concept of direct reference picks out all of the other sentences that a given sentence mentions directly in its syntax. We need however a broader definition of Reference to be able to deal with the Liar Paradox in general, since it often arises by means of a sentence referring to another sentence which may (directly) refer back to the first sentence.

4.3.1.2 N-Reference

One can use the definition of direct reference to inductively define an indexed notion of reference to identify the cases where a sentence refers to a sentence which itself refers to another sentence. We will say that a sentence, a, *1-refers* to another sentence, b, exactly when a directly refers to b. Further, a sentence a *n-refers* to a sentence b, if there is a sentence c, such that a *(n-1)-refers* to c and c directly refers to b. Thus, in the example above, A 2-refers to C, since A 1-refers to B and B 1-refers to C.

That this definition corresponds to our intuitive concept of Reference can be seen in the fact that it provides a neat definition of Groundedness. A sentence a is *grounded*, if there is some n such that a does not n-refer to any sentences. That is, whatever sentences a directly refers to in turn directly refer to other sentences which eventually refer to sentences which do not refer to any other sentences. In other words, there is some m such that the sentence a m-refers only to sentences which do not refer to other sentences, i.e. statements of direct fact. This corresponds to the definition of Groundedness.

For the purposes of this project, this concept of Reference allows a very natural definition of a circular reference loop, namely: A sentence a belongs to a *circular reference loop* iff there is some n such that a n-refers to a. This definition is very natural, and allows us to prove a number of useful results.

From these definitions, we can show a couple of easy but important lemmas:

Lemma 1. *If a m-refers to b, and b n-refers to c, then a (m+n)-refers to c.*

Proof. If b n-refers to c, then there is some sentence $d_{\text{n-1}}$ such that b (n-1)-refers to d_{n-1} and d_{n-1} 1-refers to c. This process can be iteratively repeated to find a series of sentence d_1, d_2, ..., d_{n-1} such that b 1-refers to d_1, each sentence d_i 1-refers to d_{i+1}, and d_{n-1} 1-refers to c.

Now, since a m-refers to b, and b 1-refers to d_1, it follows by the definition of n-reference that a (m+1)-refers to d_1. By iteratively repeating this process, it is necessarily the case that a (m+i)-refers to d_i, and therefore that a (m+n-1)-refers to d_{n-1}. However, since d_{n-1} 1-refers to c, it follows that a (m+n)-refers to c. □

Lemma 2. *If a belongs to a circular reference loop, there are infinitely many n such that a n-refers to a.*

Proof. If a belongs to a circular reference loop, then by definition a n-refers to a, for some n. However, that means that a 2n-refers to a, since by 1, if a n-refers to a and a n-refers to a, then a (n+n)-refers to a. By mathematical induction it follows that a (kn)-refers to a for all natural numbers k. □

Lemma 3. *It is not possible that a sentence is grounded, and that it belongs to a circular reference loop.*

Proof. If as sentence a belongs to a circular reference loop, there is some m such that a m-refers to a. Thus for any b that a k-refers to, for k<m, since a m-refers to a, and a also k-refers to b, it follows by Lemma1 that a (m+k)-refers to b. By repeating this process it follows that for every natural number j, a (jm+k)-refers to b. This means, in particular, that there can be no number i, such that a does not i-refer to any sentence.

If, on the other hand, a sentence a is grounded, there is some n such that a does not n-refer to any sentences. This obviously contradicts the consequences of a belonging to a circular reference loop. Hence if a belongs to a circular reference loop, a cannot be grounded, and vice versa. □

However, it is possible to have ungrounded sentences which do not belong to a circular reference loop. A good example is the following two sentences: c which is $\neg Tr\ulcorner d\urcorner$ and d which is $Tr\ulcorner d\urcorner$. c 1-refers to d, and d n-refers to itself for all n. Therefore c n-refers to d for all n. It, however, never refers to itself, and hence c does not belongs to a circular reference loop. This fact is important, since it allows us to distinguish potentially paradoxical sentences, such as the Truth Teller sentence d, from sentences about them, and hence give them different semantic properties.

4.3.1.3 Reference Sets

While the definition of a circular reference loop given above allows us to easily prove things metatheoretically, it has a particular formal limitation: it is an existence statement that can be technically hard to falsify. In order to prove that a sentence is not part of a circular reference loop, we have to show that there is no n such that a n-refers to a, which may require checking an infinite number of cases.

For this reason, we will introduce 2 further definitions, the "n-reference set" and the "complete reference set". The *n-reference set* of a sentence a is the set of all sentences that a m-refers to for all m less than or equal to n. Thus, the 2-reference set of a is the set of all sentences that a 1-refers to and 2-refers to. This set will be empty in many cases. It also has the property that in most cases it stabilises. That is, after some n, all of the n-reference sets for larger n are identical. When this occurs, we will label this stabilised set the complete reference set.[5]

Using these concepts, we can offer an equivalent definition of a circular reference loop, namely that a sentence a belongs to a *circular reference loop* iff there is some n such that a belongs to the n-reference set of a. This means, moreover, that if a has a complete reference set, then a belongs to that set. However, these concepts also allow us to more easily compute when a sentence does not belong to a circular reference loop. If there is a complete reference set for a sentence a, and a

[5] If we allow infinite sets, this will always exist.

does not belong to the completed reference set of a, then a cannot belong to the n-reference set of a for any n. Therefore, a does not belong to a circular reference loop.

The concept of a Complete Reference Set also enables us to provide a precise definition of the other case considered above, where there is an infinite non-repeating reference chain. If there is an infinite reference chain that does not loop, then there can be no finite n such that the n-reference sets stabilise. Each stage in the infinite reference chain will introduce a new sentence to the n-reference set. Moreover, if there is no infinite reference chain, then there must be a finite n, such that no new sentence is referred to, and therefore the n-reference sets stabilise. Thus a sentence is part of an infinite non-repeating reference chain, if, and only if, there is no finite Complete Reference Set for that sentence.

4.3.1.4 Evaluability

Assuming that the vocabulary of a formal language is increased somewhat, there is no information within the definition of a sentence being part of a circular reference loop or part of an infinite reference chain that is not representable within a language. The most radical change required is that some basic set notation and axioms are required, however, we only need some basic axioms.

Assuming we have a language in which we can represent the required concepts, it is now possible to give a simple definition of Evaluability that covers both possibilities defined above:

Definition. A sentence a is *evaluable* iff a has a finite Complete Reference Set and a is not a member of its Complete Reference Set.

Thus we have a definition of evaluability that depends purely on features that are definable within the syntax, and which is possible to compute in a language which contains the relevant concepts. However, testing for whether a sentence has a finite Complete Reference Set involves a number of concepts that cannot be defined in the more simple systems of logic.

We will therefore introduce a more restricted notion of *finite evaluability.* The idea of this concept is that a sentence is finitely evaluable if we can determine that it is evaluable in finite number of computational steps and without quantification or induction. Similarly, a sentence is finitely not evaluable, if we can determine that it is not evaluable in a finite number of computational steps and without quantification or induction.

The concept of finite evaluability is co-extensive with the concept of evaluability.

Lemma 4. *A sentence a is evaluable if, and only if, it is finitely evaluable.*

Proof. Assume a sentence a is evaluable. Therefore, by definition, it has a finite Complete Reference Set. As its Complete Reference Set is finite, it is possible to calculate this set in a finite number of steps. We need only calculate the n-reference set for the n where the reference sets cease to grow. Once we have calculate the Complete Reference Set, we need only one more step of checking whether a is a member of the Complete Reference Set, which by definition, it cannot be. We can therefore compute that a is finitely evaluable.

Similarly if a is finitely evaluable, then we must be able to compute that a is not a member of its Complete Reference Set in a finite number of steps. This implies that the Complete Reference Set is finite. Thus a is evaluable. □

The sentences which are finitely not evaluable are, however, not co-extensive with the not evaluable sentences. Obviously, there will be sentences which do not have finite Complete Reference Sets, and these cannot be computed finitely. However, it is possible that a sentence does not have a finite Complete Reference Set, yet is finitely not evaluable. The reason follows from the following Lemma:

Lemma 5. *If a sentence a is a member of one of its n-Reference Sets, then a is not evaluable.*

Proof. Assume that a is a member of its k-Reference Set, for some fixed k. By the definition of n-reference set, the k-reference set is a subset of every (k+i)-reference set, for any i>0. This means, in particular, that if a has a Complete Reference Set, a must be a member of its Complete Reference Set and therefore a is not evaluable.

If a does not have a Complete Reference Set, it cannot have a finite Complete Reference Set, and therefore again it is not evaluable. □

However, it is always possible to compute finitely whether a sentence is a member of a n-reference set. So if a sentence is a member of an n-reference set, then it is finitely not evaluable. On the other hand, if a sentence if finitely not evaluable, then it must have a finite Complete Reference set, and be a member of that Complete Reference set. That, however, means that there is some j, such that the sentence is a member of its j-reference set.

We can therefore offer the following definitions:

Definition. A sentence a is *finitely evaluable* iff a has a finite Complete Reference Set and a is not a member of its Complete Reference Set.

If we are in a language which only allows finite sets, we alter the definition to remove the first clause on the right hand side. This however, removes the equivalence:

Definition. A sentence a is *finitely evaluable,* if a is not a member of its Complete Reference Set.

We can, moreover, offer a definition of finitely not evaluable:

Definition. A sentence a is *finitely not evaluable,* if a is a member of its k-Reference Set, for some finite k.

Both of these definitions are formally definable in a language with finite sets but without any quantification or induction. Thus the concepts of evaluability and finite evaluability are in principle definable in a very wide range of logics. This suggests that, if the Strengthened T-Schema in combination with these definitions functions as outlined above, these will provide a formal truth definition for the same wide range of logics.

4.3.2 Are these definitions adequate?

There are a number of different criteria against we should evaluate the potential adequacy of these definitions. Firstly, and most importantly for the project, is whether they are sufficient to block the Liar Paradox. Secondly, we need to check that other sentences, outside of the paradoxical sentences are also given the correct truth value. Given that we are only offering a template for a new formal truth definition, and not an actual formal definition in this chapter, these discussions can only be indicative.

These definitions of evaluability and finite evaluability can only block the Liar Paradox, in combination with the Strengthened T-Schema, if all possible paradoxical sentences are come out as not evaluable. A core feature of the Strengthened T-Schema is that the standard T-Schema holds for evaluable sentences. If any paradoxical sentences are evaluable, then the paradox will arise in the normal way. It was, however, argued in the first chapter that one of the defining features of the Liar Paradox is that it can only be generated when the paradoxical sentences belong to a circular reference loop. The definitions of Evaluability and Finite Evaluability are defined, however, so that every sentence which belongs to a circular reference loop is not evaluable. Hence, the standard mechanism by which the Paradox bites is prevented on these definitions. This does not prove consistency, that cannot be done without a more explicit definition, but it demonstrates that the standard paradoxical argument form will not hold if we adopt these definitions.

Every evaluable sentence will, on these definitions, get the same truth value as it would under a different truth definition, since the normal T-Schema holds for all of these sentences. However, there are some esoteric sentences which are identified as not evaluable on these definitions although intuitively they seem true. One example is:

$k \qquad (a \supset a) \vee \neg Tr\ulcorner k \urcorner$

This sentence would normally be true as one disjunct is a tautology. However, on this definition it is not evaluable as it is part of a circular reference loop. It will therefore come out as not true on the Strengthened T-Schema. The definition of evaluability outlined above is not sensitive to the internal syntax of sentences, whereas our intuitions here with regards to truth are that the truth of one disjunct overrides the non-evaluability of the other disjunct.

This problem could certainly be resolved by a more fine-grained definition of evaluability. This will not be undertaken here however for a couple of reasons. Firstly, it is not an easy project and a satisfactory definition of evaluability of this form will require more sophisticated concepts that we want to allow for our approach. The exact definition will also be highly dependent on the logic involved. In classical logic, for example, the interdefinability of the connectives needs to be adequately dealt with but other logics do not have this. Secondly, it makes consistency much harder to justify and prove, as one needs to be very careful that no paradoxical sentences with a non-standard sentence structure are counted as evaluable. Thirdly, the ultimate aim of this definition is to show that the Strengthened T-Schema and a plausible concept of evaluability will provide a consistent truth definition. There is undoubtedly room for a more fine grained definition of evaluability, if it can be found.

In any case, we will see in future chapters that the situation for sentences as these is not as bad as it appears here. The results of Chapter 6, in fact, will suggest an alternative way to deal with these sentences which unifies their treatment with paradoxical sentences. We will come back to sentences such as these in future chapters.

4.4 Summary

On the basis of the two simple observations, a new approach to the formal definition of truth has been developed in this chapter. This definition of truth, if implementable, potentially offers a classically consistent solution to the Liar Paradox without any restrictions on the sentences which can be assigned truth values.

The two observations were that i) the truth of a sentence is determined against something external to that sentence, and ii) the truth of sentences which contain a truth predicate are ordinarily assigned a truth value through a two step process. Following the second observation through means that there must be sentences which cannot be evaluated using this ordinary process since the process can never complete. Liar Sentences are one type of example of these sentences. This, however, imposes a second condition on a sentence being true to the ordinary one expressed in the standard T-Schema, namely the question of whether it is evaluable through the ordinary means. If it is not evaluable through these means, then it cannot be true.

This in turn gives rise to a new principle governing the definition of a truth predicate, the Strengthened T-Schema:

Str. T-Schema: $Tr\ulcorner P\urcorner \equiv (P \wedge Eval\ulcorner P\urcorner)$

This definition makes use of an "Evaluability" predicate, and a couple of closely related ways to define this predicate were motivated. The different definitions are appropriate to systems of logic with different internal powers of expressibility. The exact definition to be adopted will naturally depend on the language involved, and as it involves explicit reasoning about the patterns of reference defined in the language, this will normally required extra vocabulary in the formal language. Examples will be given in the next two chapters of how this can be achieved for a Classical Sentential

Logic.

Most usefully, the analysis in this chapter also provides a test by which we can judge whether a definition of Evaluability will satisfactorily deal with the Liar Paradox. The key feature of the Strengthened T-Schema is that if a sentence is not evaluable, then a Paradox cannot be derived in any of the normal ways as the Standard T-Schema does not apply. This will be used in the following chapters as the basis of a consistency proof, so it is useful to state this more precisely:

Definition. An evaluability predicate $Eval$ in some language $\mathcal{LT}$ is *satisfactory* if, and only if, for any sentence a such that $Tr\ulcorner a\urcorner \equiv a \vdash \perp$ (i.e. a contradiction follows from the ordinary T-Schema), then $\vdash \neg Eval\ulcorner a\urcorner$ (i.e. a is not evaluable).

While this will ultimately guarantee that the definition is consistent, the key motivation in this project is to achieve as great a level of Semantic Closure as possible. The Strengthened T-Schema promises to provide a very high level of semantic closure as there are no restrictions on the sentences it applies to. However, the key test for semantic closure is whether the truth value of every sentence within the language is truthfully reportable within the language. That is, for every sentence, if we can metatheoretically identify its truth value from the truth definition in the language then the statement that asserts that that sentence has the relevant truth value is true in the language.

That is, a language $\mathcal{LT}$, which includes a truth predicate Tr, if a sentence b is true (or not) from the definition, then the sentence $Tr\ulcorner b\urcorner$ (or $\neg Tr\ulcorner b\urcorner$) is provable within $\mathcal{LT}$. In the context of this investigation, the most important test is whether for a Liar Sentence $\neg Tr\ulcorner l\urcorner$ (with name l), the correct one of $Tr\ulcorner l\urcorner$ or $\neg Tr\ulcorner l\urcorner$ should be provable in $\mathcal{LT}$. The Strengthened T-Schema does not guarantee that this level of semantic closure will occur, as will be shown in the next chapter.

The method of defining truth motivated in this chapter will be used to provide formal truth definitions for Sentential Classical Logic in the next two chapters.

Chapter 5

First Formal Definition

5.1 Introduction

In the previous chapter a new approach to the formal definition of a truth predicate was developed on the basis of the observation that the truth of a sentence depends on something other than that sentence, and a careful analysis of the properties of a truth predicate. This new approach holds significant formal promise as it was shown that a classically consistent definition of truth in its own language, which does not restrict the application of the truth predicate to any sentences, ought to be possible. The aim of this chapter is to prove that such a definition is in fact possible, by providing a consistent formal truth definition for Classical Sentential Logic. Naturally, if such a definition is possible for Classical Sentential Logic, it is possible for any classical logic and arguably any logic whatsoever, since no features of the definition depend crucially on Classical Logic.

The approach outlined in the previous chapter depends on defining truth with the Strengthened T-Schema, which requires the definition of an Evaluability predicate. Evaluability, in turn, depends on the pattern of reference set up by the naming conventions, or more generally, what refers to what within the scope of relevant predicates. These concepts of evaluability and reference are not definable within ordinary Classical Sentential Logic, so it will be necessary to define a new language which adds vocabulary and definitions to deal with these concepts. Given the role that the concept of reference plays in the Liar Paradox, it is highly plausible that a good definition of truth must deal with this concept in its definition. Thus there is nothing counter-intuitive about adding extra vocabulary to deal with the extra concepts required.

5.1.1 Conditions on an ideal Truth Definition

Importantly, if this extended language is to provide a definition of truth for the original language, it must not change the character of the original language. So, for example, none of the definitions which govern the logical connectives can be changed or added to; and no sentences should have a truth value in the extended language when they have a different truth value in the original language. We will use the term of an *adequate* definition to identify truth definitions in extended languages which satisfy these properties, and we can define this concept more precisely. Suppose we have our original formal language $\mathcal{L}$, and its formal truth definition is defined within the extended language, $\mathcal{LT}$. Then we have the following definition:

Definition. A language $\mathcal{LT}$ which extends a language $\mathcal{L}$ provides an *adequate truth definition* for $\mathcal{L}$ if the following three conditions hold for any sentence P expressible in $\mathcal{L}$:

1. If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash P$.

2. If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash Tr\ulcorner P\urcorner$. (where $Tr\ulcorner\urcorner$ is the truth predicate)

3. $\mathcal{LT}$ is as consistent as $\mathcal{L}$.[1]

That is, the definition is adequate, if every theorem in the original language is also a theorem in the extended language, every theorem in the original language is true in the new language, and the extended language does not introduce any inconsistencies that were not in the original language. If these conditions hold, then we are justified in saying that the predicate $Tr\ulcorner\urcorner$ provides an adequate truth definition for $\mathcal{L}$. Adequacy, as defined here, is a minimal, very weak set of criteria and only demonstrates that a proposed definition can reasonably be a truth definition. It does not guarantee that a truth definition is good or philosophically satisfying. Many existing truth-definitions satisfy these criteria. For example, if we interpret a Tarskian meta-language as an extension of the original language, then the Tarskian Meta-language will satisfy the definition of an adequate truth definition.

However, one ultimate philosophical aim in this project is not to construct truth definitions for one language within another, but to construct a truth definition which is adequate for its own language. We can almost use the same definition, and replace $\mathcal{L}$ with $\mathcal{LT}$ (or vice versa). However, doing that does not quite work, since both the first and third conditions become tautological. While the first condition is not necessary, we need to alter the third condition slightly. The point of this condition is that the truth definition does not introduce a contradiction into the language.[2] This means that we need to compare the consistency of the whole language with the consistency of the T-free part. That is, all of the language that does not include the truth predicate. In that case we get the following:

[1] This means that if $\mathcal{L}$ is consistent, $\mathcal{LT}$ is too. But if $\mathcal{L}$ is paraconsistent, then $\mathcal{LT}$ only contains the same type of inconsistencies as $\mathcal{L}$.

[2] Or at least, in the case of certain paraconsistent languages, a contradiction that was not already there.

Definition. A language $\mathcal{LT}$ provides an *adequate truth definition for itself* if the following two conditions hold for any sentence P expressible in $\mathcal{LT}$:

1. If $\mathcal{LT} \vdash P$, then $\mathcal{LT} \vdash Tr\ulcorner P \urcorner$. (where $Tr\ulcorner\urcorner$ is the truth predicate)

2. $\mathcal{LT}$ is as consistent as the T-free part of $\mathcal{LT}$.

The other large philosophical motivation for this project is the idea of semantic closure. That is in the context of truth definitions, it is possible to assert the correct truth status of every sentence in the language itself. Put more precisely, for a sentence p, another sentence which asserts the correct truth status for p (e.g. $Tr(p)$) is provable in the language itself. The ultimate philosophical aim is to develop a formal language which is semantically closed in this sense and which includes an adequate truth definition for itself.

While the previous chapter provided the template for a formal truth definition which has the potential to both adequate and semantically closed, we cannot we cannot give a definition for a language $\mathcal{L}$ in general, either of an adequate extended language $\mathcal{LT}$, or of a definition in the language itself. A truth definition based on the template from the previous chapter depends necessarily of the properties of $\mathcal{L}$. In this chapter, however, we will only be supplying a truth definition using Classical Sentential Logic as a base logic, which we will label as $\mathcal{SL}$. $\mathcal{SL}$ does not contain sufficient vocabulary to express the required concepts outlined in the previous chapter, so we have to define the truth definition in a extended language, which we will label $\mathcal{SLT}_1$. This language will be defined to offer a truth definition for both $\mathcal{SL}$ and itself. That is, the truth predicate will apply to sentences in its own language, and if adequate, the truth definition will be classically consistent.

5.1.2 Evaluability in $\mathcal{SLT}_1$

In terms of defining $\mathcal{SLT}_1$, there is one main question that we need to answer with reference to the blueprint offered in the previous chapter. Two different formal concepts of evaluability were offered, evaluability and finite evaluability. The definition of the more general concept of evaluability contains an implicit quantification which is not removable, and the definition $\mathcal{SL}$ does not allow quantification, so we will use the concept of finite evaluability to define a formal Evaluability predicate in $\mathcal{SLT}_1$. That is, we will use the following definition:

Definition. A sentence a is *finitely evaluable,* iff a is not a member of its Complete Reference Set.

It must be noted that this definition only works if our language is restricted so that it only includes finite sets of names, which is reasonable for $\mathcal{SLT}_1$ since it does not include any quantification. We must, therefore, include only finite sets (of names) in our definition of $\mathcal{SLT}_1$, which means that only finite Complete Reference Sets are possible in $\mathcal{SLT}_1$. Moreover, since we will be using finitely evaluable for the definition of the formal Evaluability predicate in $\mathcal{SLT}_1$, we must use the concept of finitely not evaluable for the negation of the formal concept. That is:

Definition. A sentence a is *finitely not evaluable,* iff a is a member of its k-Reference Set, for some finite k.

The reason for this, as discussed previously, is that there are cases where we can decide in a finite number of steps that a sentence is not evaluable because it belongs to a circular reference chain, but that sentence does not have a finite Complete Reference Set.

A disadvantage of using the concepts of finitely evaluable and finitely not evaluable is that these concepts do not exhaust the possibilities. There are possible sentences which do not have a finite Complete Reference Set, so the definition of finitely evaluable cannot apply, and they are not part of a circular reference loop. However, it is not possible to deal with this sentences without either quantification or a concept of infinity within the language, and these cannot be introduced into $\mathcal{SL}$ without fundamentally changing the logic. None of the sentences are paradoxical in the cases that we are considering in this thesis.

With these concepts in place, we can provide a formal definition of a truth predicate for Classical Sentential Logic. To make the definition easier to follow, the definition and proofs will be only be provided for the Logic without any substantive theories. The definitions and proofs work in the same way if we introduce a consistent Theory into Classical Sentential Logic. For the sake of clarity of presentation, however, we will only consider a truth definition for the bare logic.

5.2 Classical Sentential Logic with Truth

5.2.1 Basic Definitions

Let $\mathcal{SL}$ be a Classical Sentential Logic. We let $P, Q, R, ...$ stand for atomic propositions and $\{\neg, \wedge, \vee, \supset, \equiv\}$ be the set of connectives defined in the normal way. We will use $A, B, C, ...$ as metatheoretic propositional variables. We assume that SL is defined as a standard Fitch style natural deduction system.[3] It contains the following introduction rules:

	¬I		⊃I	∨ I		∧ I
1	P	1	P	P	j	P
⋮	⋮	⋮	⋮	$P \vee Q$	⋮	⋮
m	$\bot$	m	Q		k	Q
n	$\neg P$	n	$P \supset Q$		m	$P \wedge Q$

[3] Choosing a Natural Deduction system to work in admittedly restricts the logical machinery available. However, it has two significant advantages here: the resulting system can be more easily used to understand natural languages; and the aim is to show how a formal truth definition is possible in a very unlikely logical system.

And the following Elimination Rules:

$\neg$E	$\supset$E / MP	$\vee$ E	$\wedge$ E
1 $\mid \underline{\neg P}$	j P	1 $P \vee Q$	$\underline{P \wedge Q}$
$\vdots$ $\mid \vdots$	$\vdots$ $\vdots$	i $\mid \underline{P}$	P, Q
m $\mid \bot$	k $\underline{P \supset Q}$	$\mid \vdots$	
n P	m Q	j $\mid R$	
		k $\mid \underline{Q}$	
		$\mid \vdots$	
		l $\mid R$	
		m R	

We will extend $\mathcal{SL}$ to a language $\mathcal{SLT}_1$ which includes a truth definition. This requires adding certain new vocabulary, in particular some predicates ($Eval()$, $Tr()$, $Ref_n()$ and $REF()$) and a system of reference that allows one to refer to sentences within the language $\mathcal{SLT}_1$. For this language we will define a calculus of names to handle reasoning about the system of reference. This will include a new class of individual constants $(p, q, r, ...)$ which are names; finite sets of names, $\{\}$; and rules governing these.

5.2.2 Grammatical Sentences

Firstly, we need to define what a grammatical sentence in $\mathcal{SLT}_1$ is. We let $P, Q, R, ...$ stand for the same atomic propositions in $\mathcal{SLT}_1$ as in $\mathcal{SL}$, and the set of (propositional) connectives are naturally defined by the same derivation rules. We will use the letters $a, b, c, ...$ as metatheoretic variables for names.

Definition. A *basic proposition* in $\mathcal{SLT}_1$ is any string which takes one of the following forms:

- P for P an atomic proposition;
- $Eval(a)$ for some name a
- $Tr(a)$ for some name a
- $Ref_i(a) = \{b, c, d, ..., f\}$ for a name a, a finite number of names $b, c, d, ..., f$ and a number i (or replacing Ref_i with REF)
- $Ref_i(a) = Ref_1(b)$ for some names a and b and a number i (or replacing Ref_i with REF)

We can use basic propositions to define the grammatical sentences of $\mathcal{SLT}_1$ recursively. We will let $A, B, C, ...$ function as metatheoretic sentential variables within $\mathcal{SLT}_1$.

Definition. A *grammatical sentence* in $\mathcal{SLT}_1$ is either

- a basic proposition;
- $\neg A$ for some grammatical sentence A.
- $A \vee B$; $A \wedge B$; $A \supset B$; or $A \equiv B$ for some grammatical sentences A and B.

Now we assume that the atomic names $p, q, r, ...$ are assigned to sentences within $\mathcal{SLT}_1$ in some way. The method of assignment is not here important, only that it assigns at least every relevant sentence a unique name. The condition of the uniqueness of the name is a stronger condition than normal, but is necessary in this context. As argued in Chapter 3, the name of a sentence can have an effect on its truth value and potentially have paradoxical consequences. This possibility will be discussed in the next chapter. Hence in order to ensure that no problems arise in this chapter, we will only allow sentences to have one name. As a matter of convention we will let the metatheoretic name variable a be the name of the sentence A; b the name of B and so on.

Furthermore, for notational convenience, especially in the definition of axiom schema, we will adopt a further metatheoretic convention. The $\ulcorner\urcorner$ brackets are used as a notational variant for names, where the name of the relevant sentence has not been specified. Every sentence in $\mathcal{SLT}_1$ has a defined name within the class p, q, r, ..., however we have not specified which name which sentence has. Where we want to refer, metatheoretically to the name of, say $P \wedge \neg Q$, we will use the notation $\ulcorner P \wedge \neg Q \urcorner$ as it refers to whichever member of the class of names belongs to that sentence. Similarly, in the Axiom Schema, something like $\ulcorner \neg A \urcorner$ will be used to identify the name of the relevant sentence of the form $\neg A$. In this context, the $\ulcorner\urcorner$ need not be read as a name-forming device, but rather simply a a metatheoretic convention to pick out a name that is fixed by the definition of the language but that has not been identified meta-theoretically.

5.2.3 Name Calculus

The Name Calculus is a set of rules and axioms that allow us to reason about the concept of Reference introduced in the previous chapter, and therefore to be able to define Evaluability. The basic elements of the Name Calculus are the class of names, p, q, r, ..., finite sets of names $\{a, b\}$ including the empty set $\{\}$, a series of predicates $Ref_i()$, and a further predicate REF. As noted before, a, b, c,... will be used as metatheoretic sentential variables. In order to reason about sets of names, our system includes set union $\cup$, inclusion in a set $\in$, and equality between sets $=$. All are defined in the standard way, and equal sets are always substitutable.

The series of predicates Ref_i is not a single predicate but an infinite series of predicates, which primarily take names as an argument and define sets. We will however extend its definition to take sets of names in an obvious way. The Ref_i predicates are defined recursively over the construction of sentences as follows:

1. $Ref_1(a) = \{\}$ if the sentence denoted by a is a basic proposition but not of the form $Tr(b)$.
2. $Ref_1(\ulcorner Tr(b) \urcorner) = \{b\}$
3. $Ref_1(\ulcorner \neg A \urcorner) = Ref_1(\ulcorner A \urcorner)$
4. $Ref_1(\ulcorner A \vee B \urcorner) = Ref_1(\ulcorner A \urcorner) \cup Ref_1(\ulcorner B \urcorner)$
5. $Ref_1(\ulcorner A \wedge B \urcorner) = Ref_1(\ulcorner A \urcorner) \cup Ref_1(\ulcorner B \urcorner)$
6. $Ref_1(\ulcorner A \supset B \urcorner) = Ref_1(\ulcorner A \urcorner) \cup Ref_1(\ulcorner B \urcorner)$
7. $Ref_1(\{a, b, ..., c, d\}) = Ref_1(a) \cup Ref_1(b) \cup ... \cup Ref_1(c) \cup Ref_1(d)$
8. $Ref_{i+1}(a) = Ref_i(a) \cup Ref_1(Ref_i(a))$

The Ref_i predicates define the concept of n-Reference Sets discussed in the previous chapter. Importantly, the $n+1$ set includes the n set, which means that in many cases the set will stabilise after some i. This allows us to define a Complete Reference Set ($REF()$), where it is finite:

REF Axiom $[Ref_i(a) = Ref_{i+1}(a)] \supset [REF(a) = Ref_i(a)]$

That is, once the set of sentences in an n-Reference set stabilise, no more new sentences can be added. Therefore we have the complete set of sentences that the original sentence refers to. The predicates can be used to define introduction rules for $Eval$ predicate:

$Eval$ 1 $\quad a \in Ref_i(a) \vdash \neg Eval(a) \quad$ for some i

$Eval$ 2 $\quad \neg(a \in REF(a)) \vdash Eval(a)$

Thus, as per the definitions of finitely evaluable above, a sentence is not evaluable if it belongs to its own n-Reference set for any n; and is evaluable if it does not belong to it own complete reference set. It should be remembered that $\mathcal{SLT}_1$ only allows finite sets, so there will be some sentences which are not classified as either evaluable or not evaluable.

Since the $Eval$ predicate identifies the sentences that we want it to as evaluable and the others as not evaluable, we can use it to define the Tr predicate by the axiom schema:

Strengthened T-Schema: $Tr(a) \equiv A \wedge Eval(a)$[4]

[4] or equivalently $Tr\ulcorner A \urcorner \equiv A \wedge Eval\ulcorner A \urcorner$

Thus we have a definition which corresponds closely to the blueprint in Chapter 4. This should mean that if sentences are evaluable, then the Tarskian T-Schema, that is $Tr(a) \equiv A$, holds and if sentences are not evaluable, then it follows that $\neg Tr(a)$. This in turn, if the blueprint is correct, will guarantee consistency. Naturally, the proofs are below.

An immediate criticism of this definition is that it appears circular. We have defined Tr in terms of $Eval$; $Eval$ in terms of Ref_n; but Ref_n depends on Tr. This does not, however, undermine the definition. The definition of Tr in terms of $Eval$ is a definition of the truth value of $Tr(a)$. It says that the truth value of $Tr(a)$ depends on the truth values of A and $Eval(a)$. Similarly, the definition of $Eval$ in terms of Ref_n provides the truth values for $Eval$. However, $Ref_n(a)$ does not have a truth value, but is only a notational variant for a finite set of names. Furthermore, the definition of Ref_n in terms of instances of the Tr predicate does not in any way depend on the truth value of Tr. It only depends on the name/sentence which falls within the scope of the Tr predicate. From a formal point of view, this is a purely syntactic property. Thus the definition of the truth value of Tr depends on which names fall within the scope of that predicate in the relevant sentences, but that is what we should expect.

5.2.4 Basic Results

Before we move to an analysis of any paradoxes and the consistency of $\mathcal{SLT}_1$, we will first note some basic properties of the system. These will be explored in the form of a series of Lemmas about $\mathcal{SLT}_1$. The first establishes one of the properties that need to be satisfied if $\mathcal{SLT}_1$ is to be an adequate truth definition for $\mathcal{SL}$.

Lemma 6. *Every theorem of $\mathcal{SL}$ is also a theorem of $\mathcal{SLT}_1$.*

Proof. If A is a theorem of $\mathcal{SL}$, then there is a proof of A within $\mathcal{SL}$. However, all the derivation rules of $\mathcal{SL}$ are valid in $\mathcal{SLT}_1$. Therefore there will be a proof of A within $\mathcal{SLT}_1$. □

The next Lemma shows that all sentences which do not include Truth Predicate are evaluable. This shows that the definition follows from the intuitions established in the previous chapter. It is only sentences with a Truth Predicate which have a different truth value evaluation procedure which leads some sentences into paradox. It is therefore only these sentences which should be affected by the formal concept of evaluability.

Lemma 7. *Every T-free sentence is evaluable.*

Proof. Let A be a T-free sentence. That is, A does not contain any instances of the Tr predicate. From the definition of grammatical sentences, this means that either A is a Basic Proposition which is not of the form $Tr(b)$ or it is composed of Basic Propositions of this form. For any Basic Proposition P in A, $Ref_1(\ulcorner P \urcorner) = \{\}$. It follows from the recursive definition of Ref_i that $Ref_1(\ulcorner A \urcorner) = \{\}$.

From this fact, we can prove that A is evaluable:

1	$Ref_1(\ulcorner A \urcorner) = \{\}$	
2	$Ref_2(\ulcorner A \urcorner) = [Ref_1(\ulcorner A \urcorner) \cup Ref_1(Ref_1(\ulcorner A \urcorner))]$	Defn of Ref_{i+1}
3	$Ref_2(\ulcorner A \urcorner) = [\{\} \cup Ref_1(\{\})]$	Substitute 1 into 2
4	$Ref_2(\ulcorner A \urcorner) = [\{\} \cup \{\}]$	Defin of Ref()
5	$Ref_2(\ulcorner A \urcorner) = \{\}$	Set Union
6	$Ref_2(\ulcorner A \urcorner) = Ref_1(\ulcorner A \urcorner)$	Substitute 1 into 5
7	$REF(\ulcorner A \urcorner) = Ref_1(\ulcorner A \urcorner)$	REF Axiom and MP on 6
8	$REF(\ulcorner A \urcorner) = \{\}$	Substitution from 1 in 7
9	$\neg(\ulcorner A \urcorner \in REF(\ulcorner A \urcorner))$	Defn of $\in$
10	$Eval(\ulcorner A \urcorner)$	Eval 2

Thus every T-free sentence A is evaluable. □

The next Lemma simply shows that if a sentence is evaluable, then the ordinary T-Schema holds, and all normal intuitions about truth are correct.

Lemma 8. *$Eval(a) \vdash Tr(a) \equiv A$.*

Proof.

1	$Eval(a)$	Hypothesis
2	$Tr(a) \equiv (A \wedge Eval(a))$	Strengthened T-Schema
3	$\mid\ \underline{Tr(a)}$	Assume
4	$\mid\ A \wedge Eval(a)$	$\supset$Elim 2,3
5	$\mid\ A$	$\wedge$Elim 4
6	$Tr(a) \supset A$	$\supset$Intro 3-5
7	$\mid\ \underline{A}$	Assume
8	$\mid\ A \wedge Eval(a)$	$\wedge$Intro 1,7
9	$\mid\ Tr(a)$	$\supset$Elim 2,8
10	$A \supset Tr(a)$	$\supset$Intro 7-9
11	$Tr(a) \equiv A$	$\equiv$ Intro 7,11

□

The next Lemma illustrates the power of the Strengthened T-Schema, for it establishes formally the conclusion in the previous chapter that if a sentence is not evaluable, then it is not true.

Lemma 9. $\neg Eval(a) \vdash \neg Tr(a)$

Proof.

1	$\neg Eval(a)$	Hypothesis
2	$Tr(a) \equiv (A \wedge Eval(a))$	Strengthened T-Schema
3	$\quad\mid A \wedge Eval(a)$	Assume
4	$\quad\mid Eval(a)$	$\wedge$Elim 3
5	$\quad\mid Eval(a) \wedge \neg Eval(a)$	$\wedge$Intro 1,4
6	$\neg(A \wedge Eval(a))$	$\neg$Intro 3-5
7	$\neg Tr(a)$	Modus Tollens 2,6

□

The following Theorem establishes that $\mathcal{SLT}_1$ indeed provides a truth definition for $\mathcal{SL}$ since every sentence that is a theorem in $\mathcal{SL}$ is true in $\mathcal{SLT}_1$.

Theorem 10. *For any sentence A, if A is a theorem of $\mathcal{SL}$, then $Tr(a)$ is a theorem of $\mathcal{SLT}_1$.*

Proof. If A is a theorem of $\mathcal{SL}$, then A is T-free, since there is no Truth Predicate within $\mathcal{SL}$. Therefore, by Lemma 7, A is evaluable, i.e. $Eval(a)$ is true. Since A is a theorem, this means that $A \wedge Eval(a)$ is true. Therefore, by the Strengthened T-Schema, so is $Tr(a)$. □

Thus $\mathcal{SLT}_1$ is a reasonable extension of $\mathcal{SL}$, and the truth definition satisfies the first two conditions on it being an adequate truth definition for $\mathcal{SL}$. The third condition of consistency must wait until after the treatment of the relevant paradoxes. Since Classical Logic is consistent, we must show that $\mathcal{SLT}_1$ is also consistent.

5.3 Paradoxes

Given the limited expressive resources of $\mathcal{SLT}_1$, and in particular the existence of only four predicates, we have a limited range of Liar Sentences to consider. In particular, we do not need to worry about Quinean Liar Sentences, since the concepts involved are not expressible in $\mathcal{SLT}_1$. Thus we will focus on the archetypal, self-referential Liar Sentence in $\mathcal{SLT}_1$ is a sentence, $\neg Tr(l)$, which has name l.

Theorem 11. *The sentence $\neg Tr(l)$ (which has name l) does not immediately produce a contradiction in $\mathcal{SLT}_1$.*

Proof. The properties of the sentence l in $\mathcal{SLT}_1$ are determined by the Strengthened T-Schema, since no other Axiom Schema or derivation rules are defined for the Truth Predicate.

Therefore the question of whether l is consistent in $\mathcal{SLT}_1$ is equivalent to the question of whether the sentence: $Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$ is consistent in $\mathcal{SLT}_1$.

Firstly, we can note that a sentence of the form $A \equiv \neg A \wedge B$ is consistent in Sentential Logic. If we consider it in terms of truth tables, if the truth values of A and B are both false, the truth value of $A \equiv \neg A \wedge B$ is true. Therefore if $Tr(l)$ and $Eval(l)$ are both false, the sentence $Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$ is consistent with Sentential Logic.

The consistency of the sentence $Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$ within $\mathcal{SLT}_1$ is supported by the fact that we can prove both $\neg Tr(l)$ and $\neg Eval(l)$ from the Strengthened T-Schema using only the rules of $\mathcal{SL}$. This means that $Tr(l)$ and $Eval(l)$ are presumably both false.

Firstly, we prove that $\neg Tr(l)$:

1	$Tr(l) \equiv (\neg Tr(l) \wedge Eval(l))$	Str. T-Schema
2	$Tr(l)$	assume
3	$\neg Tr(l) \wedge Eval(l)$	$\supset$E 1,2
4	$\neg Tr(l)$	$\wedge$Elim 3
5	$Tr(l) \wedge \neg Tr(l)$	$\wedge$Intro 2,4
6	$\neg Tr(l)$	$\neg$ Intro 2-5

We can also prove $\neg Eval(l)$:[5]

1	$Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$	Str. T-Schema
2	$\neg Tr(l)$	See Previous Proof
3	$\neg(\neg Tr(l) \wedge Eval(l))$	Modus Tollens 1,2
4	$\neg\neg Tr(l) \vee \neg Eval(l)$	DeM 3
5	$Tr(l) \vee \neg Eval(l)$	Double Negation 4
6	$\neg Eval(l)$	Disj. Syll 2,5

These proofs are simple applications of the rules of Sentential logic. If Sentential Logic is consistent, the model theoretic argument above means that no contradiction can be derived purely from the Strengthened T-Schema (for the Liar Sentence in question) using the rules of $\mathcal{SL}$.

Furthermore, there are no other rules or axioms which govern the behaviour of the Truth Predicate, and therefore the conclusion that $\neg Tr(l)$ cannot be directly contradicted in $\mathcal{SLT}_1$. However, there are other rules governing the behaviour of the *Eval* predicate. If these rules contradict the conclusion, then the Liar Sentence will be inconsistent, otherwise it should be consistent. That means if $Eval(l)$ is a theorem, it is inconsistent, but if $\neg Eval(l)$ is a theorem it is consistent.

We therefore compute the value of $Eval(l)$:

[5] It can be proved independently of the previous proof, but the proof is more involved.

1	$Ref_1(l) = Ref_1(\ulcorner Tr(l)\urcorner)$	Defn of Ref (n.3)
2	$Ref_1(\ulcorner Tr(l)\urcorner) = \{l\}$	Defn of Ref (n.2)
3	$Ref_1(l) = \{l\}$	Substitutivity 1,2
4	$l \in Ref_1(l)$	Defn of $\in$
5	$\neg Eval(l)$	Eval 1

Therefore the *Eval* rules are consistent with the conclusions drawn from the Strengthened T-Schema and the rules of $\mathcal{SL}$. If the *Eval* rules and $\mathcal{SL}$ are consistent, then no contradiction can be derived in $\mathcal{SLT}_1$ from the sentence $\neg Tr(l)$ with the name l. □

As can be seen from the proof, a contradiction is not derivable because of the *Eval* predicate. If we just have the normal T-Schema, once we can prove that $\neg Tr(l)$, it immediately follows that $Tr(l)$. However, with the Strengthened T-Schema, it only follows that either $Tr(l)$ or $\neg Eval(l)$. Given that the point of *Eval* is to identify which sentences are not evaluable, and hence when the normal T-Schema cannot hold, this is reasonable.

Formally, the *Eval* predicate prevents the derivation of the contradiction directly from $Tr(l)$, without having to compute whether the sentence is evaluable or not. This has a couple of important consequences. Firstly, it means that the burden of preventing a contradiction moves from the definition of Tr to the definition of *Eval*. If *Eval* is defined so that it is false for all the sentences which produce a contradiction when we assume the normal T-schema, then no contradiction can be derived in $\mathcal{SLT}_1$. Secondly, this shows that the basic approach to resolving the Liar Paradox adopted here is sound. If the system turns out to be inconsistent, then there must be some sentence for which *Eval* is true, but from which a contradiction can be derived from the normal T-Schema. This means that we have not defined *Eval* correctly, as there is some paradoxical case that we did not notice. Changing the definition of *Eval* does not require a great change in our conceptual apparatus, and given the complex nature of the ways large sets of sentences can refer to each it would not be a serious problem if our initial definition failed to take into account all the possibilities.

However, given the central role that *Eval* has in preventing the Liar Paradox, it is possible that the Revenge Problem will bite here in terms of the *Eval* predicate. The most likely sentences through which it could bite are the sentences $\neg Eval(f)$ with name f and the sentence $\neg Tr(g) \wedge \neg Eval(g)$ with name g. However, neither of them produce a paradox.

Fact. *Neither f (which is $\ulcorner \neg Eval(f)\urcorner$) nor g (which is $\ulcorner \neg Tr(g) \wedge \neg Eval(g)\urcorner$ produce a paradox.*

Proof. We begin with f. f is T-free, so by Lemma 7, f is evaluable (i.e. $\vdash Eval(f)$). Therefore, by Lemma 8, $\vdash Tr(f) \equiv \neg Eval(f)$. However, since $\vdash Eval(f)$, it follows that $\neg Tr(f)$ is provable, and f is not true. Given that f says of itself that it is not evaluable, and it is evaluable this is the intuitively correct solution.

We now turn to g. From the definition of Ref, it follows that $Ref_i(g) = \{g\}$ for all $i \geq 1$. It follows therefore that g is not evaluable. Thus, by Lemma 9, $\neg Tr(g)$ is provable. Although what g says is correct, it does not produce a paradox from here.

The Strengthened T-Schema is: $Tr(g) \equiv (\neg Tr(g) \wedge \neg Eval(g)) \wedge Eval(g))$. Although $\neg Tr(g) \wedge \neg Eval(g)$ is provable, $Eval(g)$ is not, so we are in the identical situation to l. Thus if l is consistent then g is consistent. □

The proof that multiple sentence versions of the Liar Paradox are not evaluable and hence not true is essentially the same as the proof for l. These cases will be included within the proof below that $\mathcal{SLT}_1$ is consistent. However, there another type of Paradox to deal with first, namely Curry's Paradox. Thus we consider the sentence $Tr(k) \supset Q$ whose canonical name is k, and where Q is arbitrary.

Lemma 12. *The sentence $Tr(k) \supset Q$ with name k is not paradoxical.*

Proof. We begin by observing the following derivation:

1	$Tr(k) \equiv ((Tr(k) \supset Q) \wedge Eval(k))$	Str T-Schema
2	$Tr(k)$	Assume
3	$(Tr(k) \supset Q) \wedge Eval(k)$	MP ln.1,2
4	$Tr(k) \supset Q$	& Elim ln.3
5	Q	MP ln.2,4
6	$Tr(k) \supset Q$	$\supset$Intro ln.2-5

Thus the sentence $Tr(k) \supset Q$ (i.e. k) is provable within $\mathcal{SLT}_1$. Normally, the paradox arises because from a proof of k, we can derive $Tr(k)$ and therefore Q. However, in $\mathcal{SLT}_1$we can only derive $Tr(k)$ if $Eval(k)$ is derivable. Therefore we will compute the value of $Eval(k)$:

1	$Ref_1(k) = Ref_1(Tr(k)) \cup Ref_1(Q)$	Ref Rule 7
2	$Ref_1(Tr(k)) = \{k\}$	Ref Rule 3
3	$Ref_1(k) = \{k\} \cup Ref_1(Q)$	Substitutivity ln 1,2
4	$k \in Ref_1(k)$	Defn $\in$
5	$\neg Eval(k)$	Eval 1

Thus $\neg Eval(k)$ is derivable and hence by Lemma 9, $\neg Tr(k)$ is derivable. Therefore, Q cannot be proven within $\mathcal{SLT}_1$. □

Thus the definitions within $\mathcal{SLT}_1$ deal equally well with the Liar Paradox and with Curry's Paradox. We have not shown that $\mathcal{SLT}_1$ is consistent, just that the ordinary methods by which an inconsistency arises have been blocked. Unless there is some other means for deriving an inconsistency, then $\mathcal{SLT}_1$ will be consistent (on the assumption that $\mathcal{SL}$ is consistent). However, showing this requires a far more detailed proof.

5.4 Consistency

The standard method in formal logic for demonstrating the consistency of a formal system is to prove the system is sound with respect to a (consistent) model. That is, one proves that there is a model of the relevant formal system, and since the model is consistent and the provable sentences are a subset of the true sentences in the model, the formal system must be consistent. A key motivation for the development of $\mathcal{SLT}_1$ however was the negative result in Chapter 3, namely that a semantically closed formal language cannot be sound with respect to a model. If this is the case, and $\mathcal{SLT}_1$ is semantically closed in something close to the relevant sense, then it would follow that $\mathcal{SLT}_1$ cannot have a model, and therefore we cannot prove consistency using the standard method.

The argument in Chapter 3, however, relied on a couple of assumptions which do not, or need not, hold for $\mathcal{SLT}_1$. Those assumptions are the Tr Interpretation Principle ($\mathcal{M} \models Tr(p)$ iff $\mathcal{M} \models p$) and the assumption that a model must be complete (i.e. every sentence in the language has a truth value in a model). Given that one can read the Tr Interpretation Principle as a type of traditional T-Schema, and the T-Schema does not hold in $\mathcal{SLT}_1$, it is not surprising that the Tr Interpretation Principle does not hold. In fact, $\mathcal{SLT}_1$ cannot have a consistent model if the Tr Interpretation Principle holds, since on that principle it must be true, for b as defined above, that $\mathcal{M} \models Tr(b)$ iff $\mathcal{M} \models \neg Tr(b)$.

The obvious move in this context is to drop the Tr Interpretation Principle from Chapter 3, and replace it with a 'Strengthened' Tr Interpretation Principle, namely: $\mathcal{M} \models Tr(p)$ iff ($\mathcal{M} \models p$ and p is evaluable). To do this, however, is to undermine the philosophical basis of a semantic model. The idea is meant to be that the semantic model defines what is means for sentences to be true or not true. If we accept the introduction of the extra clause around evaluability, which depends on syntactic information about names, then we seem to be admitting that the model for $\mathcal{SLT}_1$ cannot completely define what is and is not true.

We can however treat a model purely as a mathematical device for proving consistency via soundness, and this is all that we need here to prove consistency. If we do this, however, we can consider defining the concept of evaluability within the model, so that sentences of the forms $Eval(p)$ and $Tr(p)$ have a truth value within the model. Evaluability, however, is a property that depends purely on which sentences have which names. In formal languages, the definition of what sentences have what names normally belongs to the syntactic definition of the language. It is not part of the normal definition of a model. If we are to have a model of $\mathcal{SLT}_1$, which means we have a model in which the 'Strengthened' Tr Interpretation Principle holds, then the model must include information about what sentences are in $\mathcal{SLT}_1$, and what names those sentences have. This is not difficult to do, and a model of this sort will be presented below. Nevertheless it comes at the cost of blurring the neat philosophical distinction between the syntactic language and the model that has existed. On this approach, syntactic entities are necessary for semantic properties and therefore must be included within the model.

The second relevant assumption in the argument in Chapter 3 is the assumption that every sentence must have a truth value in the model. While the role of the model is to fix the semantic values of the sentences within the language, it is possible that there may be sentences whose value cannot be fixed directly against the model but require some (non-recursive) process in the language to fix their semantic value. Given that the problems identified in Chapter 3 are with the semantic interpretation of the Tr predicate, a natural candidate for a set of sentences who do not have a truth value in the model are sentences (or at least some sentences) which include the Tr predicate. The idea here would be that since for the T-free part of the language provability and truth must coincide, then one can use the provable sentences to ensure that the correct sentences which include a Tr predicate are provable.

If we take this approach, that the T-free part of $\mathcal{SLT}_1$ has a model, but all of $\mathcal{SLT}_1$ does not, we do not have to include information about sentences within the semantic model. However, we cannot use soundness as a consistency proof. Nevertheless, a consistency proof is still possible. This proof is however more intricate and less philosophically illuminating so it is presented in the Appendix.

It is also possible to take the approach that the T-free part of $\mathcal{SLT}_1$ plus a defined set of sentences which include the Tr predicate has a model, while the whole of $\mathcal{SLT}_1$ does not. The most obvious way to make this idea work is to allow use the type of model Kripke defined to let all grounded sentences in $\mathcal{SLT}_1$ have a truth value in the model, and then use the approach in the Appendix to prove the whole language is consistent.

For ease of exposition, and for the philosophical illumination it sheds on the structure of $\mathcal{SLT}_1$, we will focus on building a model for all of $\mathcal{SLT}_1$. It should be remembered that this relies on a broader concept of semantic model than is normally adopted, since information about sentences and their names must be a part of the model.

5.4.1 Building a Model for $\mathcal{SLT}_1$

We will construct the model for $\mathcal{SLT}_1$ out of two smaller models for fragments of the language. The first sub-model of $\mathcal{SLT}_1$ will obviously be the standard model for $\mathcal{SL}$, since all of the theorems of $\mathcal{SL}$ are theorems of $\mathcal{SLT}_1$. The second sub-model will be a model that deals with the definition of Evaluability within the model. To do this, we will construct a model of the Name Calculus part of $\mathcal{SLT}_1$. This sub-model will be entirely separable from the model of $\mathcal{SL}$ since the only logical connective that gets used within reasoning within the Name Calculus is $\supset$, and that is to allow a conditional definition. The model of the Name Calculus can include this without needing the full definition of sentential logic. Once we have a model for the Name Calculus, the model for $\mathcal{SLT}_1$ can be built from the union of this model with the model for $\mathcal{SL}$. The key challenge to that is integrating the two in a way that is consistent with the Strengthened T-Schema.

5.4.1.1 A Model for the Name Calculus

A model for the Name Calculus is not difficult to construct. An illuminating way to think about it is as a collection of points with names and arrows which connect points. The points are names of sentences, and an arrow from one point to another means that the sentenced named by the first point refers to the sentence named by the second point. In the context of $\mathcal{SLT}_1$, this means that the name at the second point occurs within the scope of the Tr predicate within the sentence named by the first point.

Points may have have none, one or finitely many arrows leading away from them, and every point will have many arrows pointing towards it. For any p, there are many sentences, such as $Tr(p)$, $\neg Tr(p)$, and $Tr(p) \wedge Tr(q)$, that refer to it. As every name in $\mathcal{SLT}_1$ identifies one sentence (and vice versa) these arrows are well defined in the model. The set $Ref_i(p)$ is the set of points which one can get to by following up to i arrows from p. The set $REF(p)$ is the complete set of points that one can get to by following arrows from p. If we follow the definitions through, a sentence/point in this model will be evaluable if it has a finite REF set, and p itself is not in the REF set. That is, there are finitely many points accessible from p and one cannot follow an arrow away from p and end up back at p via other arrows. We will refer to a series of arrows which follow from each other at points as a path.

If p has a finite REF set, this means that one can only reach a finite number of other points (i.e. sentences) from p. This can occur under two different situations. The first is when every path from p is finite. This means that if we follow an arbitrary sequence of arrows from p, we will always reach a point which does not refer to any other sentences. These sentences which do not refer to other sentences are sentences which do not have a Tr predicate in them. In this first case, p will be grounded. The second possibility is that there is a path from p which continues indefinitely, but which eventually only repeats points which have already occurred on the path. In this case, there will be a circular loop in the arrows. Naturally there will be sentences that have finite REF sets, and that have some paths which terminate and others which lead to circular loops.

On the first possibility, if every path from p is finite, then p itself cannot be in the REF set of p, so p will be evaluable, by the definition. This part of the definition of evaluability corresponds to the definition of groundedness. On the second possibility, if there is a path from p which has a circular loop, p can only be evaluable if p is not in this loop. Otherwise, p will be in the REF set. This case, which is also evaluable on the definition, where p refers to sentences in a circular reference loop but is not itself part of the definition, goes beyond the concept of groundedness. It is this case which plays a crucial role in resolving the Paradox in a philosophically satisfying way.

This definition of Evaluability corresponds to the more general definition of Evaluability defined in Chapter 4, not to the concept of Finitely Evaluable defined in $\mathcal{SLT}_1$. We are not restricted to finite reasoning within the Model, and it is easy to check that the definitions of Evaluable coincide. Using this definition ensures that everything is either evaluable or not evaluable within the model, which means that every sentence will have a truth value within the model. It is easy to see that this model is consistent. There cannot be a situation where a sentence/point is both evaluable and not, since one can either follow arrows away from a point and back to it, or one cannot. Therefore the model of the name calculus is consistent.

5.4.1.2 A Model for $\mathcal{SLT}_1$

We will construct the model for $\mathcal{SLT}_1$ from the two sub-models by using the two sub-models to define a base and then use a recursive definition on top of the base. This procedure is similar to the recursive definition introduced by Kripke, but also draws heavily on approaches such as Herzberger's where he extended Kripke's definition.[6] Whereas Kripke used the facts in the base, which effectively corresponds here to the model of $\mathcal{SL}$, we will use a more complicated base which also includes separate sets of true and false sentences, as in Herzberger's approach. We will generate the model for $\mathcal{SLT}_1$ by the following steps:

The Base

1. Include all of the true sentences from the models of $\mathcal{SL}$ and the Name Calculus in the set $\mathcal{T}_0$ in the base; and include all of the false sentences from the models of $\mathcal{SL}$ and the Name Calculus in the set $\mathcal{F}_0$ in the base.
2. For any sentence for which $\neg Eval(p)$ is true in the base (i.e. in $\mathcal{T}_0$), add $Tr(p)$ to $\mathcal{F}_0$.
3. Close both $\mathcal{T}_0$ and $\mathcal{F}_0$ by using the following recursive definitions for the connectives in $\mathcal{SL}$:
   (a) If P is in $\mathcal{F}_0$ (or $\mathcal{T}_0$), add $\neg P$ to $\mathcal{T}_0$ (or $\mathcal{F}_0$).
   (b) For all other connectives use the standard recursive definitions.

The Recursive Definition

1. We begin with $\mathcal{T}_n$ and $\mathcal{F}_n$, and add elements to form $\mathcal{T}_{n+1}$ and $\mathcal{F}_{n+1}$ according to the following.
2. We add $Tr(p)$ to $\mathcal{T}_{n+1}$ (or $\mathcal{F}_{n+1}$) for every p that is in $\mathcal{T}_n$ (or $\mathcal{F}_n$), unless p is not evaluable (i.e. unless $\neg Eval(p)$ is true in the base).[7]

[6] See Hans G. Herzberger. "Notes on Naive Semantics". In: *Journal of Philosophical Logic* 11 (1982), pp. 61–102

[7] For the definition of $\mathcal{T}_1$ and $\mathcal{F}_1$, $\mathcal{T}_{n-1}$ and $\mathcal{F}_{n-1}$ are not defined and so are taken to be empty sets.

3. $\mathcal{T}_{n+1}$ and $\mathcal{F}_{n+1}$ are closed by using the same recursive definitions of the connectives as above.

If we take the union of all levels on this definition, we will get a model for $\mathcal{SLT}_1$and every sentence in $\mathcal{SLT}_1$ will have a truth value. For every sentence in $\mathcal{SLT}_1$ is either evaluable or it is not. If it is not evaluable, it will be false according to the model. If it is evaluable, there are two possibilities, it is grounded or it is not. If it is grounded then it will appear in the hierarchy in exactly the same way as all grounded sentences appear in Kripke's hierarchy. If it is not grounded, but is evaluable, then it must refer to a sentence which is not evaluable as it belongs to a finite reference loop. However, that not evaluable sentence is false and included in the Base, and therefore the evaluable sentence will have a derived value in the hierarchy.

We have, therefore, a model in which every sentence has a truth value. If this model is consistent, and it is genuinely a model, i.e. everything that is provable in $\mathcal{SLT}_1$ is true in this model, then $\mathcal{SLT}_1$ is consistent.

5.4.2 Is the model consistent?

We will follow the two stage construction of the model and use mathematical induction in order to show that the model is consistent. The key thing in showing that the model is consistent is showing that the sets $\mathcal{T}_i$ are consistent, that is there is no sentence such that both P and $\neg P$ are in $\mathcal{T}_i$. If the $\mathcal{T}_i$ are consistent then the $\mathcal{F}_i$ will also be consistent, as the negation rules guarantee that any inconsistency in one is transferred to the other. To do this, we will show that the base ($\mathcal{T}_0$) is consistent, and then show that if at any level n, $\mathcal{T}_n$ is consistent, then at level $n+1$, $\mathcal{T}_{n+1}$ is also consistent.

The two sub-models are obviously consistent, so we must first check that adding $\neg Tr(p)$ to $\mathcal{T}_0$ for every sentence which is not evaluable preserves consistency. All of the sentences in $\mathcal{SL}$ and the Name Calculus are T-free, hence they are evaluable. This is because a T-free sentence does not refer to another sentence and hence is grounded. This means the set of sentences that are true in the sub-models and the set of not evaluable sentences are mutually exclusive sets. Therefore, there will be no sentence q that is both true in one of the two sub-models, and the sentence $\neg Tr(q)$ is added to $\mathcal{T}_0$ by Step 2 of the definition. This means that $\mathcal{T}_0$ is consistent before the closure over the connectives.

The only connectives that could cause trouble is negation, since the recursive rules for the other connectives cannot introduce a contradiction. The key point about the negation rule, is that it ensures that the negation of every false sentence is true and that the negation of every true sentence is in $\mathcal{F}_0$. The issue therefore is whether there is some sentence which is in $\mathcal{F}_0$, but whose negation is inconsistent in $\mathcal{T}_0$; or some sentence in $\mathcal{T}_0$, whose negation is inconsistent in $\mathcal{F}_0$. We know that the submodels are consistent and do not share vocabulary, therefore the only possibility must involve the sentences of the form $\neg Tr(p)$ which were introduced into $\mathcal{T}_0$ for p that are not evaluable. However, as argued, sentences that are not evaluable and sentences from the submodels are mutually exclusive. Therefore the negation rules cannot introduce any inconsistency. This means that $\mathcal{T}_0$ is consistent.

We will now assume that T_n is consistent for some n. The second step of the recursive definition introduces sentences of the form $Tr(p)$ into $\mathcal{T}_{n+1}$. If T_n are consistent, the introduced sentences will be consistent, which means that the introduced sentences will also be consistent when closed under negation. Closing under the other connectives in classical logic cannot introduce an inconsistency if the starting set of sentences is consistent.

This means that if $\mathcal{T}_n$ is consistent, $\mathcal{T}_{n+1}$ can only be inconsistent if one of the introduced sentences contradicts a sentence that already has a truth value in level n (i.e. is in either $\mathcal{T}_n$ or $\mathcal{F}_n$). We will go through the different cases identified with regards to Evaluability to check that none of these can introduce an inconsistency.

The different cases all depend on the a key insight. In the model for the name calculus, the reference structure for each sentence is represented by arrows leading from that sentence, which sets up a complicated network when all sentences are considered. The important aspect to the network when considering evaluability is the tree leading from any particular sentence. If the tree and all paths in the tree are finite, the sentence is grounded. If the tree includes finitely many sentences, but has infinite paths (i.e. loops), it is evaluable if the original sentence is not in any loop and not evaluable if the original sentence is included in a loop. If the tree includes infinitely many sentences, it is not evaluable.

Not evaluable sentences are the easiest case to deal with. $Tr(p)$ for a p that is not evaluable is not introduced at any level apart from the base, due to the condition on the second step. Hence these sentences cannot introduce an inconsistency in the recursive definition.

We will consider the situation with grounded sentences next. For every grounded sentence, there will be a longest path in the tree leading from it. For a T-free sentences, that path length will be 0, and every sentence whose longest path length is 0 will be T-free, for any sentence that includes the truth predicate will refer to another sentence and so will have a non-zero maximum path length. This means that any sentence with maximum path length 0 will be in the Base.

Any sentence with maximum path length 1 will have at least one part of the form $Tr(p)$ where p is T-free, and no parts of the form $Tr(q)$ where q is not T-free. This means that every sentence with maximum path length 1 will be introduced into the model by the recursive definition in level 1, i.e. $\mathcal{T}_1$ and $\mathcal{F}_1$, and not in any earlier level. One can easily see that this pattern will continue, and that all sentence with maximum path length n will be introduced at level n, and not at any earlier level.

Now since $\mathcal{T}_{n-1} \subset \mathcal{T}_n$, the introduction rule ($Tr(p)$ for every p in $\mathcal{T}_n$), every sentence of the form $Tr(p)$ in $\mathcal{T}_n$ will be 'reintroduced' into $\mathcal{T}_{n+1}$. Since these are sets this does not affect any properties of the sets. The key thing to focus on are the sentences that are introduced by the recursive definition at level n, which are not introduced at any previous level. We know that every sentence with maximum path length n in the name calculus model are introduced at level n. However, we also know that every sentence with maximum path length less than n are introduced at a smaller level and those with maximum path length greater than n are introduced at a higher level. This means that the 'new' introductions at any level n are exactly those with maximum path length n (for grounded sentences). This means that no grounded sentence introduced at level n can be in contradiction with a sentence introduced at a lower level, since the sentences with smaller maximum path length are a distinct set of sentences. This demonstrates that grounded sentences cannot introduce an inconsistency.

The final category to consider are sentences that are evaluable, but are not grounded. Similar considerations apply to grounded sentences, except that we measure the maximum length to a not evaluable sentence in which none of the sentences in the path are themselves not evaluable. In this case, the Base will contain all sentences with maximum path length 1, and level 1 will introduce all sentences with maximum path length 2 and so on. However, as in the case with grounded sentences, the set of sentences introduced at each level is distinct, and so no sentence can be introduced at a level which contradicts a sentence at a lower level. Therefore no sentence that is evaluable but not grounded can introduce an inconsistency in the recursive hierarchy.

This exhausts the cases, and hence it follows that $\mathcal{T}_{n+1}$ is consistent, if $\mathcal{T}_n$ is consistent. By induction, every level of the hierarchy is consistent and the model is therefore consistent.

5.4.3 Is every provable sentence true in the model?

We must again follow the two stage construction of the model to show that every provable sentence is truth in the model.

The first step is to note that as the models for $\mathcal{SL}$ and the Name Calculus are part of the base, all provable sentences in these two parts of $\mathcal{SLT}_1$ are automatically in the model. The only part of $\mathcal{SLT}_1$ that is not in one of these is the Strengthened T-Schema. If we can show that the Strengthened T-Schema holds for every sentence in $\mathcal{SLT}_1$, the fact that we close the model under the recursive definition for the connectives at each level means that all of the classical consequences from the Strengthened T-Schema will also hold in the model. This will mean that everything provable in $\mathcal{SLT}_1$ is true in the model.

Every sentence in $\mathcal{SLT}_1$ is either evaluable or it is not evaluable. For every non-evaluable sentence b, both $\neg Tr(b)$ and $\neg Eval(b)$ are true in the base, so $Tr(b) \equiv (b \wedge Eval(b))$ is true by the recursive definition of the connectives since both sides are false. This means that the Strengthened T-Schema is true in the base for every sentence that is not evaluable.

If c is evaluable, $Eval(c)$ is true in the base and the standard T-Schema will hold as $Eval(c)$

is true. Also it means that either c is grounded, in which case every path from it ends with a T-free sentence, or that at least one path ends in a circular loop which does not contain c. If c is grounded, then every path ends with a T-free sentence, but every T-free sentence in $\mathcal{SLT}_1$ is a member either of $\mathcal{SL}$ or the Name Calculus. All the T-free sentences will have a truth value fixed in the base, and the recursive process will only transmit this up to c and $Tr(c)$. The recursive definition will mean that either c and $Tr(c)$ are both true or are both not true. If c is not grounded, and has a path that ends in a circular reference loop, all of the sentences in that circular loop will be not evaluable and hence not true in the base. The relevant values will transmit up the levels and recursive definition will mean that either c and $Tr(c)$ are both true or are both not true. Again in both cases the Strengthened T-Schema will hold as $Eval(c)$ is true.

This means that the Strengthened T-Schema is true in the model for every sentence in $\mathcal{SLT}_1$ and hence every provable sentence in $\mathcal{SLT}_1$ is true in the model.

5.5 Final Results

We have shown that $\mathcal{SLT}_1$ includes a consistent truth definition within a classical sentential logic that does not place any restrictions on which sentences the truth predicate can apply to. That is, any sentence can be within the scope of the truth predicate, and the resulting sentence will have a truth value. This makes this a very promising formal truth definition, particularly for philosophical applications, but we need to test it against the criteria on a good definition that were defined above.

Fact 13. *$\mathcal{SLT}_1$ provides an adequate truth definition for $\mathcal{SL}$ in the sense defined above.*

Proof. The definition of an adequate truth definition included three conditions. The first was: If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash P$. This first condition is proven by Lemma 6. The second condition was that: If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash Tr\ulcorner P\urcorner$. This is proven by Theorem 10. The third condition, that $\mathcal{LT}$ is as consistent as $\mathcal{L}$, follows immediately from the consistency proof. □

However, the same does not apply for $\mathcal{SLT}_1$ itself.

Fact 14. *$\mathcal{SLT}_1$ does not provide an adequate truth definition for itself.*

Proof. While $\mathcal{SLT}_1$ satisfies the consistency condition, the first condition, namely "If $\mathcal{LT} \vdash P$, then $\mathcal{LT} \vdash Tr\ulcorner P\urcorner$" does not hold. In particular, we know that $\mathcal{SLT}_1 \vdash \neg Tr(b)$, for b the liar sentence defined above. However it does not follow that $\mathcal{SLT}_1 \vdash Tr\ulcorner\neg Tr(b)\urcorner$ since $\ulcorner\neg Tr(b)\urcorner$ is identical to b and it is not the case that $\mathcal{SLT}_1 \vdash Tr(b)$. This could only be the case if $\mathcal{SLT}_1$ was inconsistent. □

Thus while $\mathcal{SLT}_1$ provides an adequate truth definition for $\mathcal{SL}$, and allows every sentence which includes a truth predicate to have a truth value, it does not provide an adequate truth definition for itself. In other words, $\mathcal{SLT}_1$ is not semantically closed in the desired sense. The form of this problem suggests that Kripke's problem has simply been pushed up a level. Although we can now assert that the Liar Sentence is not true; we cannot assert that it is true that the Liar Sentence is not true. To this problem, we could take Kripke's solution and suggest that at this point we still need a ghost of the Tarskian hierarchy, and have to resort to a metalanguage. However, another solution to this will be explored in the next chapter and involves changing the way we understand define names within a formal logic. For the moment though, it is enough to note that although $Tr\ulcorner\neg Tr(L)\urcorner$ is not true within the system, $Tr\ulcorner\neg Tr(l) \wedge \neg Tr(l)\urcorner$ is a provable theorem.

5.6 Summary

In this chapter we have provided a consistent and adequate truth definition for Classical Sentential Logic. Most remarkably, this was achieved without changing any features of the underlying logic and simply by adding extra machinery to deal with the reference relations between sentences and the truth predicate. The success of this extra machinery means that it is clear that an equivalent definition can be added to any more sophisticated logic.

The key to this definition is the approach to the semantics of the language, which includes the definition of truth. Truth was taken to be about the relationship between syntactic entities and semantic facts, which means that it depends on more than one condition. One of these conditions is traditionally semantic and the other traditionally syntactic, however both are required for a successful definition of a truth predicate. This overlap of the traditional separation meant that, in the approach adopted here, information that was traditionally syntactic is required within the semantic model of the language. While this approach undercuts traditional formal practise, it accords with natural language practise. If one wants to work out the truth value of "The first sentence in this chapter is true", the information about what sentence is being referred to is relevant semantic information and necessary to working out the truth value of the sentence. Including the information about what sentences refer to what in the model is simply including this type of information.

The truth definition presented in $\mathcal{SLT}_1$ is a viable candidate for solving the philosophical problem generated by the Liar Paradox. It is consistent, and the language satisfies the four conditions on a Grammar-Only language being affected by the Liar Paradox, as natural languages do. The truth definition is not tied to a particular logic, which is also a positive. However, it fails to be fully semantically closed, as there are sentences which intuitively ought to be true but which cannot be true in $\mathcal{SLT}_1$. This problem will be dealt with in the next chapter.

Chapter 6

Second Formal Definition

The formal truth definition offered in Chapter 5 is remarkable in that it allows a consistent and plausible definition of truth within a classical logic. Moreover, it does not require any restrictions on grammatically acceptable sentences, and allows all grammatically acceptable sentences to have a determinate truth value. This means it is a language whose Grammar-Only part satisfies the conditions on a Grammar Only language being affected by the Liar Paradox. Moreover, any sentence can be referred to within the scope of the truth predicate without contradiction. Thus Chapter 5 offers a Logical Language whose Grammar Only part fulfills the philosophically desirable conditions and the Logical Language is not affected in the stronger sense. Importantly, there is nothing in extension of $\mathcal{SL}$ to $\mathcal{SLT}_1$ that depends on working with a classical logic, so the definition can be adapted to any formal system of logic. We have a formal truth definition which consistently satisfies the linguistic conditions on a solution to the Liar Paradox, which means that the assumptions about truth embodied in this truth definition are plausibly the correct ones for our understanding of natural languages. However, as noted in the previous chapter, this definition does not achieve everything that an ideal formal definition should achieve.

A key aim for a formal truth definition is that it achieves semantic closure. That is, the correct semantic properties of every sentence in the language can be stated (by true sentences) within the language. Unfortunately, this does not hold in $\mathcal{SLT}_1$ as there are sentences which have certain semantic properties, but it is not possible to assert those properties within the language. For example, given the sentence b which is $\neg Tr(b)$, it follows in $\mathcal{SLT}_1$ that that b is not true, i.e. we can prove that $\neg Tr(b)$. However, it does not follow that an assertion that " 'b is not true' is true" is true. That is, it is not possible to assert the truth of this conclusion within $\mathcal{SLT}_1$. The problem is that the sentence which expresses " 'b is not true' is true" in $\mathcal{SLT}_1$ is $Tr(b)$, which is false because $\neg Tr(b)$ is provable. While we can assert that 'b is not true', we cannot assert the semantic fact that it is true that "b is not true" within the language. This problem is obviously a limitation in the application of this approach to natural languages. If it is the case that b is not true, we would expect the sentence " 'b is not true' is true" to be true. Otherwise we would not have correctly captured the concept of truth. This means that there is still a gap between the formal concept of truth definition in $\mathcal{SLT}_1$ and our understanding of the concept of truth.

Fortunately as a point of analysis, in terms of overcoming this limitation, the definitions in Chapter 5 are based on only half of the analysis in Chapter 3. In Chapter 3, two main types of problems were identified with the standard model theoretic approach to semantics in the context of the Liar Paradox. The first was the definition of truth purely with respect to a model, which was addressed in Chapters 4 and 5 by including certain facts about the reference structure of the language in the definition of truth through the concept of evaluability. However, the second type of problem was the inability of a language with a recursive semantics, or in fact a language that assumes that sentence types are truth bearers, to be semantically closed.

The truth definition in the previous chapter is, at least in a strict sense, a non-recursive definition. The truth value of a sentence does not purely depend on the semantic properties of the components of the sentence, but also on the name of the sentence and the interaction between the name and the general reference structures in the language. The model defined as part of the consistency proof, however, shows that the semantic properties of sentences can be defined recursively, so long as facts about evaluability are taken to be primitive. In any case, $\mathcal{SLT}_1$ was defined so that each sentence (i.e. grammatical string of symbols) had only one name, and therefore it follows that all sentences of the same type in $\mathcal{SLT}_1$ must have the same truth value. The analysis of Chapter 3 predicts that this means that $\mathcal{SLT}_1$ cannot be semantically closed, which has been confirmed above. Therefore, the analysis in Chapter 3 shows that if we want to achieve semantic closure, then we need to modify or replace $\mathcal{SLT}_1$ and do away with the assumption that sentence types are the truth bearers.

6.1 Is this necessary?

Before setting out to find a way of modifying $\mathcal{SLT}_1$ so that sentence types are no longer the truth bearers, it is important to consider whether this is a plausible account of similar situations in natural languages. In terms of the philosophical motivation of this project, these is little point pursuing a truth definition which assumes that sentence types are not truth bearers if natural languages demonstrably make this assumption. The debate about the correct truth bearers in natural languages is a vigorous and detailed debate that cannot be covered here, beyond saying that there is no universal consensus on the matter. The fact that there are live options in the debate which argue that sentence types are not truth bearers is perhaps sufficient justification for pursuing this part of the project. It could therefore be a reasonable position to take, and if it can be shown that a formal language which takes something else as truth bearers can be semantically closed, then that would be evidence that sentence types are not the primary truth bearers.

We can, however, offer more direct evidence that the type of change being considered to the formal languages is a reasonable one in the context of considering natural languages. One way of seeing this is to consider the following situation:

> The country of Xodarap has in recent history had a change of government. The previous President was a highly mendacious and rarely, if ever, told the truth. The current President, in contrast, is transparently honest, and made this honesty the cornerstone of his campaign. His rallying cry during the campaign was the assertion that "What the current President of Xodarap is saying is not true." Given the legacy of his predecessor, the current President legislated that if the President of Xodarap ever uttered a falsehood, he must be removed from office.
>
> One day, the President fell ill and was gripped by a terrible fever. After lying in bed almost all day, unable to speak and barely able to move. Suddenly, and with great energy, he sat up straight and defiantly repeated his campaign rallying cry: "What the current President of Xodarap is saying is not true." The illness then regained control of his body, he sunk to his bed and could not speak for the rest of the day.
>
> The situation is further complicated by another comment that was made at the same time on the same day. The President's doctor, after assessing the President's peculiar fever, pronounced that, in his medical opinion the nature of the fever will mean that "What the current President of Xodarap is saying is not true."

The most important task, for our purposes here, is to consider the truth status of the doctor's assertion. Given that both assertions were made at the same time, then the doctor's assertion will be true if the President's assertion is not true and vice versa. However, the doctor made an assertion of exactly the same sentence type as the President, which appears to show that these two assertions of the same sentence type cannot have the same truth value. It is commonly accepted that sentences of the same type which contain indexicals will differ in truth value, since the meaning of the sentence will depend on the meaning of the indexical in the context. In this case, however, while there is an indexical in the sentence, 'What the current President of Xodarap is saying', the indexical refers to exactly the same thing in both assertions, and hence there is no difference in meaning due to indexicality. Thus this natural language example provides a situation where the same sentence type does not always have the same truth value which is not captured by indexicality. It is reasonable to pursue a formal solution which allows sentences of the same type to differ in meaning, as has been proposed.

It might be objected that this analysis has only used a very simplistic conception of sentence types, and say that accounts of sentence types which take into account some role for context would not have the same problem. Firstly, the point of this example is only to show that questions about sentence types arise in the context of the Liar Paradox and that it is plausible to look at alternatives to sentence types as truth bearers in this context. Secondly, and more importantly, standard formal languages implicitly adopt this simple conception of sentence types that identical strings of symbols have identical truth values, and moreover there is normally no defined concept of context in formal languages that can play a role. If a more sophisticated account of sentence types is necessary to deal with this type of situation in natural languages, then an alternative approach must also be adopted in a formal language, if we are to provide an account of truth and reasoning in natural languages within a formal language.

The fact that situations such as that with Xodarap provide situations where it is plausible to move away from sentences types as truth values means that it is reasonable to look for an alternative approach to formal languages so that sentence types are not (always) truth bearers, at least when considering the Liar Paradox.

6.2 Truth-bearers

The idea being advanced is to work with a formal definition in which sentence types can have different truth values, at least in certain circumstances. Given the success of $\mathcal{SLT}_1$ in addressing the Liar Paradox, and the fact that the formal definition in $\mathcal{SLT}_1$ addresses one of the key issues identified, we propose to modify $\mathcal{SLT}_1$ so that it is not assumed that identical sentence types have identical truth values. The immediate question is then what should play the role of truth bearer. While the normal alternatives to sentence types as truth values are propositions or sentence tokens, as argued in the Introduction each of these options is problematic in this context, at least on a standard account of either.

On the one hand, standard understandings of propositions are not fine-grained enough to distinguish between the statements made by the President and the Doctor in the example above. Each person has uttered the same sentence in virtually the same context and would normally be taken to have expressed the same proposition. On the other hand, any account which takes sentence tokens as truth bearers has to account for the following problem, particularly in a formal setting where context does not play much of role. Argument and derivations typically assume that if we assert a new sentence token of a type already proven, then that new token is still proven or true. If this is not always the case since different tokens can have different truth values, then it is necessary for an account that uses tokens to be able to account for when such a move is valid and when it is not. To put it differently, tokens of the same type normally have the same truth value, how can we know when they do and when they do not? While this problem may not be insurmountable, it requires a rethinking of the whole approach to proof in a formal context.

The story of the President, however, offers a hint of a different approach to understanding truth values. The sentence "What the current President of Xodarap is saying is not true" as the *President's* assertion seems to have one truth value, while the same sentence as the *Doctor's* assertion seems to have a different truth value. The person who uttered the sentence, at least in far as this affected the place of the sentence within the reference structure, affected the truth value. The fact that one sentence is identified as the *President's* assertion means in the context that that sentence is self-referential. The fact that the other sentence is the *Doctor's* assertion takes it out of the circular reference loop, which means that it can potentially have a different truth value.

If we take the concept of evaluability as defined in previous chapters, the distinction between the President's assertion and the Doctor's assertion is clear. The sentence, as the President's assertion, is not evaluable as it belongs to a circular reference loop: the sentence refers to itself as that is the only relevant sentence. However, the sentence, as the Doctor's assertion, is evaluable as it refers to the President's assertion which refers to itself. The Doctor's assertion is not in a circular reference loop, and so is evaluable.

While in this particular example the key fact is who asserted the relevant sentence, other examples show that the important information is the method we use to refer to or identify a particular sentence. For example, if we have a piece of paper with a single sentence written on it, the following are only a few examples of the ways we can identify the same sentence: "The first sentence on this page"; "The only sentence on this page"; "What is written on this page". If, moreover, that sentence reads "The first sentence on this page is not true", then the definite description used to identify the sentence appears to affect the truth value of the sentence. While the sentence itself, i.e. "The first sentence on this page is not true", is paradoxical; there does not seem to be any reason to suppose that "The only sentence on this page is not true" should be problematic. This definite description identifies a clear sentence and no circular reference arises. This is the same sort of situation as in the Xodarap example. In both cases, the way of identifying sentences affected the possible truth values that particular sentences could hold. It follows therefore that we need to look at the combination of sentence, with the means of identifying the sentence, as a truth bearer.

Within formal definitions, the standard means by which sentences are identified and referred to is through names. Sentences are assigned names of some sort, and those names are used to refer to the sentence. Analogously to the Doctor and the President and the definite descriptions, the name of a sentence will determine whether it belongs to a circular reference loop or not. The obvious step for a formal language is therefore to treat the combination, or pair, of a sentence and a name as the truth bearer. That is, it is sentences as identified by names that truth is predicated of. In $\mathcal{SLT}_1$, where every sentence has one unique name, this distinction is meaningless as there is no real distinction between a sentence and its name.

As pointed out above, in natural languages there are many cases where we can refer to the same sentence, and even the same sentence token, in different ways. To take the example of the piece of paper with a single sentence written on it, the following are only a few examples of the ways we can identify the same sentence token: "The first sentence on this page"; "The only sentence on this page"; "What is written on this page". These need not be strictly names in the way we understand them in a formal language, but they play the same role as names in the formal language. Moreover, as in the example given above, if that sentence reads "The first sentence on this page is not true", we can see the same type of behaviour as in the Xodarap example. The sentence under one method of reference is problematic as it sets up a circular reference. To use the terminology of the last couple of chapters it is not evaluable. However, the same sentence token considered under the other methods of reference are not problematic and are evaluable. This will be taken to mean that there is a difference in truth value here.

The difference in truth value depends on the method of referring to a sentence, which in a formal language is its name. Truth values therefore depend both on the sentence and, at least in a formal context, the name being used to refer to the sentence. In order to implement this in a formal language, we will therefore take sentence/name pairs as the basic truth bearers in the following formal definition.

6.3 Preliminaries for a Formal Definition

Since we are adopting the convention that the primary truth bearers are combinations of a name and a sentence, it necessary to allow single sentence types (or grammatically correct strings of symbols) to have multiple names. Moreover, it is possible that different names for the same sentence type will differ in whether they are evaluable and therefore have different truth values. Importantly, this possibility provides a way around the semantic closure problem identified above with $\mathcal{SLT}_1$.

Suppose now that we have two sentence-name pairs, one being the sentence $\neg Tr(b)$ with the name b, and another being the same sentence $\neg Tr(b)$ with the name c. If we look at the definition of evaluability, c is evaluable, as it does not exist in a circular reference loop. This fact is crucial, as it means that the standard T-Schema holds for c, which means that if we can prove that $\neg Tr(b)$ then we can prove that $Tr(c)$. In other words, it will be possible to prove the assertion that "It is true that "b is not true"" within the formal language. It is this assertion that was not provable within $\mathcal{SLT}_1$, since in $\mathcal{SLT}_1$ this assertion could only be represented by $Tr(b)$, which cannot be consistently proven. Thus, taking truth bearers as name/sentence pairs, which allows the same sentence to have different names with different truth values, allows us to address the semantic closure problems with $\mathcal{SLT}_1$.

Moreover, this solution mirrors the intuitive solution to the Xodarap example above. In that case, it is very plausible to say that there is something different between the Doctor's assertion and the President's assertion, and therefore the two different assertions can have different truth values. That would allow that different assertions of the same sentence type in the same context can have different truth values, in this case because of different speakers.

Allowing the same sentence type to have different truth values when identified by different names has therefore the potential to achieve semantic closure in the sense that, for any sentence P in the relevant language, for some name p of P, one of $Tr(p)$ or $\neg Tr(p)$ will be provable. This is possible because, at least for the cases relevant to the Liar Paradox, for any name/sentence pair that is not evaluable, it is always possible that there is another name for the same sentence type that is evaluable.

In Chapter 4, two types of situations were identified which meant that a name/sentence pair could be not evaluable. The first case, which is the case relevant to the Liar Paradox, is when the name/sentence pair is within a circular reference loop. Two tokens of the same type must refer to the same sentences and therefore have identical reference sets. Thus, if we have name/sentence pair that is not evaluable, so long as there is another name for the same sentence which is not within the reference loop, then this second name/sentence pair will be evaluable.

Moreover, whenever there is a name/sentence pair that is *not* evaluable for this reason, it will be possible to choose another name for the same sentence that *is* evaluable provided the language is sufficiently liberal with naming. Suppose a name/sentence pair is not evaluable because it belongs to at least one circular reference loop. Since we are not considering cases where quantification over sentences is allowed, the name/sentence pair must belong to finitely many circular reference loops. Moreover, a circular reference loop can only include finitely many name/sentence pairs within the loop. Therefore there can be only finitely many name/sentence pairs within the circular reference loops that the relevant sentence belongs to. It will therefore be possible to identify the original sentence by means of a name that is outside the finite number of names within the circular reference loops, and the original sentence paired with this name will be by definition evaluable. It does not belong to any of the relevant circular reference loops.

This means that whenever there is name/sentence pair that is paradoxical, it will be possible to assert that the sentence type involved is true or not true using an alternative name for the same sentence type. If this works, it will be possible to non-paradoxically assert the truth status of every sentence in the language.

The second situation where a name/sentence pair is not evaluable is of less interest as it does not lead to paradox. However, it gives rise to different problems with semantic closure, which are not resolved in the same way by the change to name/sentence pairs. Suppose we have an infinite series of sentences such as the following:

S_1 S_2 is true.

S_2 S_3 is true.

S_3 S_4 is true.

$\vdots$ $\vdots$

All of these sentences are not evaluable as the semantic process that governs the truth predicate does not ever identify a sentence with a definite truth value. It follows, if we adopt the Strengthened T-Schema, that each of these sentence is not true. However, say we have a further sentence A which says that "S_1 is not true". A combined with the series S_n also makes an infinite series which will face exactly the same problem as the infinite series that only included the S_n. Thus although we would conclude that S_1 is not true, we cannot conclude within the language that A is true, even though that is what A asserts. Moreover, there is nothing special about the sentence A. Any sentence which refers to any $S_n$will have exactly the same problem. Thus the approach outlined here that works for the paradoxical cases with circular reference cannot provide semantic closure in these cases.

This will be simply accepted here as a limitation of this method, which has no real consequences when the Liar Paradox is considered. These types of cases involving infinite series are only paradoxical if we allow quantification over sentences, which has not been considered. So consistency is not threatened if this problem is left out. In $\mathcal{SLT}_1$, these types of sentences were not given any truth value, since the computation of their evaluability requires quantification or infinite sets, neither of which were available in the sentential logic. This limitation in achieving semantic closure will therefore not affect an extension of $\mathcal{SLT}_1$ which does not change the underlying logic.

The important point is that semantic closure is possible for all sentences that relate to the Liar Paradox if we shift to taking name / sentence pairs as truth bearers. Semantic closure in this context would have to mean that any semantic (i.e. to do with the truth definition) fact that is true for the language can be asserted as true within the language using an evaluable name/sentence pair which expresses this fact. Thus every semantic fact is assertible within the language, just not every instance of a sentence type which expresses that fact is true.

This limitation is firstly fairly minor, and secondly reflects what seems to actually occur in natural languages. For example, consider the following liar sentence:

2) Sentence 2) is not true.

Now suppose that a detailed discussion and argument follows this sentence, and that this argument leads to a definite conclusion about the truth status of this sentence. It is quite possible that the conclusion of the argument could be expressed in the following sentence: Sentence 2) is not true. In terms of ordinary use of language, there is nothing problematic about this form of asserting the conclusion. If, however, we accept that the conclusion can be stated in this form, then there is nothing strange about making distinctions between the truth values of different instances of a sentence type. For this conclusion is stated using the same sentence type as the paradoxical sentence.

The approach to be adopted is therefore to alter $\mathcal{SLT}_1$ to give a new formal language $\mathcal{SLT}_2$ that allows sentence types to be associated with more than one name. The predicates in $\mathcal{SLT}_2$ which take names as arguments will therefore refer to name/sentence pairs, rather than sentence types as in the case with $\mathcal{SLT}_1$ in which sentences had canonical names. This means that whenever a name appears within a predicate in $\mathcal{SLT}_2$, it should be read as referring to a name/sentence pair, rather than simply a sentence. This will allow $\mathcal{SLT}_2$ to be semantically closed in the way described above.

Before this system is presented, however, it is necessary to consider the problem posed previously with respect to adopting sentence tokens as truth bearers in formal systems. Formal derivations normally work on the basis that a sentence type is proven, and it follows that any repeated assertion of the same sentence type will also be proven. The problem was that if sentence tokens are the truth bearers there is no guarantee that, when one token of a sentence type is proven and therefore true, that another token of the same type will be true. This same point also applies to name/sentence pairs, as two different pairs with the same sentence may have different truth values. This could mean that a radical change in argument structure is required.

However, there is a natural way of structuring the formal system so that this is not required, and this can be philosophically and intuitively justified. Firstly, we should note that we are moving to name/sentence pairs as truth bearers because of particular problems that arise when predicating over sentences. This means that the possibility of having two name/sentence pairs with the same sentence with differing properties is only relevant in cases where the sentence includes a predicate that predicates over sentences. This means that the majority of sentences are not affected.

However, due to the nature of the truth definition being implemented, a careful treatment of the remaining sentences is not required. The Strengthened T-Schema, for a name a; sentence A and name/sentence pair a/A; when fully spelt out would read "a/A is true iff A and a/A is evaluable". This means to prove that "a/A is true", we need to prove that A and that "a/A is evaluable". The first of these works purely on sentence types, as it always has in formal logic, and at least in $\mathcal{SLT}_1$ the second has a series of rules which govern its derivation. Only Truth and Evaluability need to take into account name/sentence pairs, and these are catered for already in the definitions. There does not need to be any change to the proof structure as the relevant distinctions are already tracked within the language.

While it is completely coherent to treat provability as a property of sentence types and truth and evaluability as properties of name/sentence pairs, this may seem like a sleight of hand or conceptually unappealing. If this is the case, the practice to be adopted below of proving sentence types but predicating over name/sentence pairs can be seen as a notational convenience. As argued above, every name/sentence pair with the same sentence type that is evaluable will have the same truth status. Moreover, for every sentence type, it is possible to ensure that there is a name for that type which is evaluable. It follows, therefore, that proofs of sentence types can be considered to be proofs of any name/sentence pairs including the same sentence type where the name/sentence pair is evaluable. This fact could be included notationally, but it would be necessary to identify a name for every line of proof and then invoke rules about when one can use the same sentence with a different name. This would add notation and a vast amount of work without actually changing any of the proofs or the conclusions. Given that the equivalent proofs hold and that it is far simpler, the system will be defined so that proofs work on sentence types.

6.3.1 Adequate definition of truth

The system $\mathcal{SLT}_2$ will be judged against the same criteria as $\mathcal{SLT}_1$, namely as to whether it provides adequate truth definitions for $\mathcal{SL}$ and itself, and the extent to which it offers a semantically closed solution. The following repeats what was defined at the start of Chapter 5.

Definition. A language $\mathcal{LT}$ which extends a language $\mathcal{L}$ provides an *adequate truth definition* for $\mathcal{L}$ if the following three conditions hold for any sentence P expressible in $\mathcal{L}$:

1. If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash P$.

2. If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash Tr\ulcorner P \urcorner$. (where $Tr\ulcorner\urcorner$ is the truth predicate)

3. $\mathcal{LT}$ is as consistent as $\mathcal{L}$.

It should be noted that we need to be careful as to how we understand the second condition in this context, where name/sentence pairs are acting as truth bearers. In this context we need to consider the situation where one sentence can have different names with different truth values and properties. Given this, we can either allow that the second condition holds for all name/sentence pairs which include the relevant sentence, or that it holds for at least one name/sentence pair which includes the relevant sentence. If the second condition held for all name/sentence pairs in a particular language, then for any provable sentence in $\mathcal{L}$ all assertions of its truth with all names would also be provable. However, this would not allow for sentences to have different truth values under different names, which is a key to the approach being adopted here. Admittedly, if $\mathcal{L}$ does not include a truth predicate, then the distinction will be irrelevant, but the aim of this definition is to be as general as possible. We will therefore consider the second condition to be satisfied, if there is some name/sentence pair that satisfies it. The definition will therefore be the following:

Definition. A language $\mathcal{LT}$ which extends a language $\mathcal{L}$ provides an *adequate truth definition* for $\mathcal{L}$ if the following three conditions hold for any sentence P expressible in $\mathcal{L}$:

1. If $\mathcal{L} \vdash P$, then $\mathcal{LT} \vdash P$.

2. If $\mathcal{L} \vdash P$, then for some name p, p is a valid name of P and $\mathcal{LT} \vdash Tr(p)$.

3. $\mathcal{LT}$ is as consistent as $\mathcal{L}$.

The same considerations hold for, and are in general more relevant to, the definition of an adequate truth definition for itself:

Definition. A language $\mathcal{LT}$ provides an *adequate truth definition for itself* if the following two conditions hold for any sentence P expressible in $\mathcal{LT}$:

1. If $\mathcal{LT} \vdash P$, then for some name p, p is a valid name of P and $\mathcal{LT} \vdash Tr(p)$.

2. $\mathcal{LT}$ is as consistent as the T-free part of $\mathcal{LT}$.

In the previous chapter it was shown that $\mathcal{SLT}_1$ failed to provide an adequate truth definition for itself, and for that reason it was not semantically closed. The problem was that there was a particular sentence that was provably true, but the assertion that it was true was not true in $\mathcal{SLT}_1$. The aim in the construction of $\mathcal{SLT}_2$ is to avoid this problem, so a defining test of $\mathcal{SLT}_2$ will be whether it succeeds in this.

6.4 Classical Sentential Logic with Truth (2)

6.4.1 Definition

As in the previous chapter, we let $\mathcal{SL}$ be a Classical Sentential Logic, with $P, Q, R, ...$ as atomic propositions; $\{\neg, \wedge, \vee, \supset, \equiv\}$ the set of connectives defined in the normal way; and $A, B, C, ...$ as metatheoretic propositional variables. Furthermore, we assume that SL is defined as a standard Fitch style natural deduction system. It contains the following introduction rules:

$\neg$I		$\supset$I		$\vee$ I	$\wedge$ I	
1	P	1	P	P	j	P
⋮	⋮	⋮	⋮	$P \vee Q$	⋮	⋮
m	$\bot$	m	Q		k	Q
n	$\neg P$	n	$P \supset Q$		m	$P \wedge Q$

And the following Elimination Rules:

$\neg$E	$\supset$E / MP	$\vee$ E	$\wedge$ E
1 $\neg P$	j P	1 $P \vee Q$	$P \wedge Q$
$\vdots$ $\vdots$	$\vdots$ $\vdots$	i P	P, Q
m $\bot$	k $P \supset Q$	$\vdots$	
n P	m Q	j R	
		k Q	
		$\vdots$	
		l R	
		m R	

We extend $\mathcal{SL}$ to a language $\mathcal{SLT}_2$ which includes a truth definition. We first add a class of individual constants $p, q, r, ...$ (and corresponding metatheoretic variables $a, b, c, ...$) to $\mathcal{SLT}_2$ which will function as names for sentences within $\mathcal{SLT}_2$. We will assume no particular method of attaching names to sentences. Furthermore, in order to be able to identify within the system which sentences have which name, we will introduce a two place predicate, $N(a, A)$, which takes Names as its first argument and Sentences as the second argument. $N(a, A)$ will simply be read as "a is a name of the sentence: A".[1] One can also understand $N(a, A)$ as meaning that a/A is a valid name/sentence pair.

Extending $\mathcal{SL}$ also requires adding certain new vocabulary, in particular some predicates ($Eval()$, $Tr()$, $Ref_n()$, $REF()$) and a calculus of names that allows reason about the reference relationships between sentences (or to be more precise, between name/sentence pairs) within $\mathcal{SLT}_2$. This will include a new class of individual constants $(p, q, r, ...)$ which are names; finite sets of names, $\{\}$; and rules governing these.

6.4.2 Grammatical Sentences

Firstly, we need to define what a grammatical sentence in $\mathcal{SLT}_2$ is. We let $P, Q, R, ...$ stand for the same atomic propositions in $\mathcal{SLT}_2$ as in $\mathcal{SL}$, and the set of (propositional) connectives are naturally defined by the same derivation rules. We will use the letters $a, b, c, ...$ as metatheoretic variables for names.

[1] This could be left to the metatheory, or rules of application of the system as in the previous chapter, but it is more informative to be able to express this within the language, $\mathcal{SLT}_2$.

Definition. A *basic proposition* in $\mathcal{SLT}_2$ is any string which takes one of the following forms:

- P for P an atomic proposition
- $N(a, A)$ for some name a and some grammatical sentence A.
- $Eval(a)$ for some name a
- $Tr(a)$ for some name a
- $Ref_i(a) = \{b, c, d, ..., f\}$ for a name a, a finite number of names $b, c, d, ..., f$ and a number i (or replacing Ref_i with REF)
- $Ref_i(a) = Ref_1(b)$ for some names a and b and a number i (or replacing Ref_i with REF)

It should be noted that the names within the scope of the various predicates denote name/sentence pairs, rather than only sentences. We can use basic propositions to define the grammatical sentences of $\mathcal{SLT}_2$ recursively. We will let $A, B, C, ...$ function as metatheoretic sentential variables within $\mathcal{SLT}_2$.

Definition. A *grammatical sentence* in $\mathcal{SLT}_2$ is either

- a basic proposition;
- $\neg A$ for some grammatical sentence A.
- $A \vee B$; $A \wedge B$; $A \supset B$; or $A \equiv B$ for some grammatical sentences A and B.

We will assume that the naming conventions of $\mathcal{SLT}_2$ are fixed by an explicit list of axioms of the form $N(a, A)$. This list of axioms is subject to the unambiguity condition that no name can be attached to two sentences. As previously stated, we will generally assume that there is at least one axiom of the form $N(a, A)$ for every sentence A. This list of naming axioms will be taken as primitive in the system, however, it could be generated, for example, by a Gödel Numbering system, as long as these conditions are satisfied.

6.4.3 Name Calculus

The Name Calculus is a set of rules and axioms that allow us to reason about the concept of Reference introduced in Chapter 4, and therefore to be able to define Evaluability. The basic elements of the Name Calculus are the class of names, p, q, r, ..., finite sets of names $\{a, b\}$ including the empty set $\{\}$, a series of predicates $Ref_i()$, and a further predicate REF.

In order to reason about sets of names, our system includes set union $\cup$, equality between sets $=$ and set inclusion $\in$. Both are defined in the standard way, and equal sets are always substitutable.

The series of predicates Ref_i is not a single predicate but an infinite series of predicates, which primarily take names as an argument and define sets. We will however extend its definition to take sets of names in an obvious way. The Ref_i predicates are define recursively over the construction of sentences as follows:

1. $Ref_1(a) = \{\}$ if the A, such that $N(a, A)$, is a basic proposition not of the form $Tr(b)$.
2. $N(a, Tr(b)) \supset Ref_1(a) = \{b\}$
3. $(N(a, \neg B) \wedge N(b, B)) \supset Ref_1(a) = Ref_1(b)$
4. $(N(a, B \vee C) \wedge (N(b, B) \wedge N(c, C))) \supset (Ref_1(a) = (Ref_1(b) \cup Ref_1(c))$
5. $(N(a, B \wedge C) \wedge (N(b, B) \wedge N(c, C))) \supset (Ref_1(a) = Ref_1(b) \cup Ref_1(c))$
6. $(N(a, B \supset C) \wedge (N(b, B) \wedge N(c, C))) \supset (Ref_1(a) = Ref_1(b) \cup Ref_1(c))$
7. $Ref_1(\{a, b, ..., c, d\}) = Ref_1(a) \cup Ref_1(b) \cup ... \cup Ref_1(c) \cup Ref_1(d)$
8. $Ref_{i+1}(a) = Ref_i(a) \cup Ref_1(Ref_i(a))$

These Ref_i predicates again define the concept of n-Reference Sets. Importantly, the $n + 1$ set includes the n set, which means that in many cases the set will stabilise after some i. This allows us to define a Complete Reference Set, where it is finite:

REF Axiom $[Ref_i(a) = Ref_{i+1}(a)] \supset [REF(a) = Ref_i(a)]$

That is, once the set of sentences in an n-Reference set stabilise, no more new sentences can be added. Therefore we have the complete set of sentences that the original sentence refers to. The predicates can be used to define introduction rules for the *Eval* predicate:

Eval 1 $\quad a \in Ref_i(a) \vdash \neg Eval(a)$ $\quad$ for some i

Eval 2 $\quad \neg(a \in REF(a)) \vdash Eval(a)$

That is, if a is in any n-Reference set then a is not evaluable. If a has a complete reference set and is not a member of its own reference set, then a is evaluable. In this way, the *Eval* predicate identifies the correct sentences as evaluable and not evaluable. We therefore use it to define the Tr predicate by the axiom schema:

N - Strengthened T-Schema: $N(a, A) \supset [Tr(a) \equiv A \wedge Eval(a)]$

The key difference between the Strengthened T-Schema in $\mathcal{SLT}_1$ is that this version explicitly checks in the language that the name is a name of the correct sentence. This is necessary as sentences can have more than one name in $\mathcal{SLT}_2$. It should also be remembered that $Tr(a)$, given $N(a, A)$ holds, is to be read as saying that the name/sentence pair a/A is true. We have defined a formal language $\mathcal{SLT}_2$ which includes a truth definition and allows there to be multiple names for the same sentence. This was introduced in order to allow a semantically closed formal truth definition. Obviously we now need to check whether that is achievable.

6.4.4 Basic Results

The basic properties of the system are identical to those of $\mathcal{SLT}_1$, and the basic proofs are almost identical. We will therefore in general only give detailed proofs for results where the proof differs significantly because of the different naming convention.

Lemma 15. *Every theorem of $\mathcal{SL}$ is also a theorem of $\mathcal{SLT}_2$.*

Proof. If A is a theorem of $\mathcal{SL}$, then there is a proof of A within $\mathcal{SL}$. However, all the derivation rules of $\mathcal{SL}$ are valid in $\mathcal{SLT}_1$. Therefore there will be a proof of A within $\mathcal{SLT}_1$. □

Lemma 16. *Every T-free sentence is evaluable.*

Proof. Let A be a T-free sentence. That is, A does not contain any instances of the Tr predicate. From the definition of grammatical sentences, this means that either A is a basic proposition which is not of the form $Tr(b)$ or it is composed of basic propositions of this form. For any basic proposition P in A, $Ref_1(\ulcorner P \urcorner) = \{\}$. It follows from the recursive definition of Ref_i that $Ref_1(\ulcorner A \urcorner) = \{\}$. From this it quickly follows that A is evaluable, as its reference sets are all empty. The full proof is identical to that of the equivalent proof in $\mathcal{SLT}_1$. It follows that every T-free sentence A is evaluable. □

The next proof makes explicit use of the N Predicate, and therefore will be given, although it is essentially identical to the equivalent proof in the previous chapter. This Lemma shows that whenever a name/sentence pair is evaluable, the standard T-Schema holds.

Lemma 17. $N(a, A), Eval(a) \vdash Tr(a) \equiv A$.

Proof. The proof is the following formal derivation:

1	$N(a, A)$	Hypothesis
2	$Eval(a)$	Hypothesis
3	$N(a, A) \supset [Tr(a) \equiv (A \wedge Eval(a))]$	Strengthened T-Schema
4	$Tr(a) \equiv (A \wedge Eval(a))$	MP ln.1,3
5	$\mid \underline{Tr(a)}$	Assume
6	$\mid A \wedge Eval(a)$	$\supset$Elim 4,5
7	$\mid A$	$\wedge$Elim 6
8	$Tr(a) \supset A$	$\supset$Intro 5-7
9	$\mid \underline{A}$	Assume
10	$\mid A \wedge Eval(a)$	$\wedge$Intro 2,9
11	$\mid Tr(a)$	$\supset$Elim 4,10
12	$A \supset Tr(a)$	$\supset$Intro 9-11
13	$Tr(a) \equiv A$	$\equiv$ Intro 9,13

□

The next Lemma shows that if a name/sentence pair is not evaluable, then it is necessarily not true. Again the proof makes explicit use of the N predicate.

Lemma 18. $N(a, A), \neg Eval(a) \vdash \neg Tr(a)$

Proof. The proof is the following formal derivation:

1	$N(a, A)$	Hypothesis
2	$\neg Eval(a)$	Hypothesis
3	$N(a, A) \supset [Tr(a) \equiv (A \wedge Eval(a))]$	Strengthened T-Schema
4	$Tr(a) \equiv (A \wedge Eval(a))$	MP ln.1,3
5	$\mid \underline{A \wedge Eval(a)}$	Assume
6	$\mid Eval(a)$	$\wedge$Elim 5
7	$\mid Eval(a) \wedge \neg Eval(a)$	$\wedge$Intro 2,6
8	$\neg(A \wedge Eval(a))$	$\neg$Intro 5-7
9	$\neg Tr(a)$	Modus Tollens 4,8

□

The following Theorem demonstrates one of the key results to show that $\mathcal{SLT}_2$ provides an adequate truth definition of $\mathcal{SL}$.

Theorem 19. *For any sentence A s.t. $N(a, A)$, if A is a theorem of $\mathcal{SL}$, then $Tr(a)$ is a theorem of $\mathcal{SLT}_2$.*

Proof. If A is a theorem of $\mathcal{SL}$, then A is T-free, since there are is no Truth Predicate within $\mathcal{SL}$. Therefore, by Lemma 16, A is evaluable, i.e. $Eval(a)$ is a theorem of $\mathcal{SLT}_2$. Since A is a theorem of $\mathcal{SL}$, by Lemma 15, A is a theorem of $\mathcal{SLT}_2$. Thus both A and $Eval(a)$ are theorems.

Furthermore, since we are also given that $N(a, A)$ is a theorem, then it follows from the N-Str T-Schema that $Tr(a)$ is a theorem. □

Thus $\mathcal{SLT}_2$ thus far satisfies all of the properties that were satisfied by $\mathcal{SLT}_1$. We now need to check how it works when faced by the Liar Paradox.

6.4.5 Paradoxes

In $\mathcal{SLT}_2$, the archetypal, self-referential Liar Sentence is defined by the existence of a naming axiom of the form: $N(l, \neg Tr(l))$. We will therefore show that adding this axiom to $\mathcal{SLT}_2$ does not have any undesirable consequences.

Theorem 20. *Accepting $N(l, \neg Tr(l))$ as a theorem/axiom in $\mathcal{SLT}_2$ does not immediately produce a contradiction.*

Proof. We first note that the N-Str T-Schema, in the relevant context, has the following form: $N(l, \neg Tr(l)) \supset [Tr(l) \equiv \neg Tr(l) \wedge Eval(l)]$. Given the assumption that $N(l, \neg Tr(l))$ is a theorem of $\mathcal{SLT}_2$, it follows by Modus Ponens that $\vdash Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$.

Thus the question of whether $N(l, \neg Tr(l))$ is consistent in $\mathcal{SLT}_2$ is equivalent to the question of whether the sentence: $Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$ is consistent in $\mathcal{SLT}_2$. The proof is identical to the proof in the case of $\mathcal{SLT}_1$.

Firstly, we can note that a sentence of the form $A \equiv \neg A \wedge B$ is consistent in Sentential Logic. If we consider it model theoretically, if the truth values of A and B are both false, the truth value of $A \equiv \neg A \wedge B$ is true, and hence this sentence is consistent with Sentential Logic.

Within $\mathcal{SLT}_2$ we prove both $\neg Tr(l)$ and $\neg Eval(l)$ from the Strengthened T-Schema.

Firstly, we prove that $\neg Tr(l)$:

1	$N(l, \neg Tr(l))$	Hypothesis
2	$N(l, \neg Tr(l)) \supset [Tr(l) \equiv \neg Tr(l) \wedge Eval(l)]$	Str. T-Schema
3	$Tr(l) \equiv (\neg Tr(l) \wedge Eval(l))$	MP 1,2
4	$\quad Tr(l)$	assume
5	$\quad \neg Tr(l) \wedge Eval(l)$	$\supset$E 3,4
6	$\quad \neg Tr(l)$	$\wedge$Elim 5
7	$\quad Tr(l) \wedge \neg Tr(l)$	$\wedge$Intro 4,6
8	$\neg Tr(l)$	$\neg$ Intro 4-7

We can also prove $\neg Eval(l)$:

1	$N(l, \neg Tr(l))$	Hypothesis
2	$N(l, \neg Tr(l)) \supset [Tr(l) \equiv \neg Tr(l) \wedge Eval(l)]$	Str. T-Schema
3	$Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$	Str. T-Schema
4	$\neg Tr(l)$	See Previous Proof
5	$\neg(\neg Tr(l) \wedge Eval(l))$	Modus Tollens 3,4
6	$\neg\neg Tr(l) \vee \neg Eval(l)$	DeM 5
7	$Tr(l) \vee \neg Eval(l)$	Double Negation 6
8	$\neg Eval(l)$	Disj. Syll 4,7

These proofs are simple applications of the rules of Sentential logic, and hence if Sentential Logic is consistent, then no contradiction can be derived purely from the Strengthened T-Schema (for the Liar Sentence in question).

Furthermore, there are no other rules or axioms which govern the behaviour of the Truth Predicate, and therefore the conclusion that $\neg Tr(l)$ cannot be directly contradicted in $\mathcal{SLT}_2$. However, there are other rules governing the behaviour of the *Eval* predicate. If these rules contradict the conclusion, then the Liar Sentence will be inconsistent, otherwise it will be consistent. That means if $Eval(l)$ is a theorem, it is inconsistent, but if $\neg Eval(l)$ is a theorem it is consistent.

We therefore compute the value of $Eval(l)$:

1	$Ref_1(l) = Ref_1(\ulcorner Tr(l) \urcorner)$	Defn of Ref (n.3)
2	$Ref_1(\ulcorner Tr(l) \urcorner) = \{l\}$	Defn of Ref (n.2)
3	$Ref_1(l) = \{l\}$	Substitutivity 1,2
4	$l \in Ref_1(l)$	Defn of $\in$
5	$\neg Eval(l)$	Eval 1

Therefore no contradiction can be derived from the sentence $\neg Tr(l)$ with the name l. □

This shows that the system $\mathcal{SLT}_2$ shares the most of same properties as $\mathcal{SLT}_1$ in terms of how it deals with paradoxes. However, crucially for the success of this project, it avoids the problem of semantic closure that was identified for $\mathcal{SLT}_1$ in the way it was designed to. The idea behind moving to name/sentence pairs as truth bearers is that it is possible for the same sentence to have different truth values when it is paired with different names. With regards to the Liar Paradox, the idea was that while the pair $l/\neg Tr(l)$ would not be true as it is not evaluable, it ought to be the case that any other pair $p/\neg Tr(l)$ would be true, and therefore the relevant truth could be expressed within $\mathcal{SLT}_2$.

The following proves that this is in fact the case:

Fact 21. *If $N(p, \neg Tr(l))$ and $N(l, \neg Tr(l))$ are theorems, then $\vdash Tr(p)$ and $\vdash \neg Tr(l)$*

Proof. The first stage is to compute whether p is evaluable. To do this it is necessary to assume a name for the sentence $Tr(l)$, and we choose q:

1	$N(p, \neg Tr(l))$	
2	$N(q, Tr(l))$	Define
3	$(N(p, \neg Tr(l)) \wedge N(q, Tr(l))) \supset Ref_1(p) = Ref_1(q)$	Ref Axiom 3
4	$N(q, Tr(l)) \supset Ref_1(q) = \{l\}$	Ref Axiom 2
5	$N(p, \neg Tr(l)) \wedge N(q, Tr(l))$	$\wedge$ Intro ln.1,5
6	$Ref_1(p) = Ref_1(q)$	MP 6,8
7	$Ref_1(q) = \{l\}$	MP 5,7
8	$Ref_1(p) = \{l\}$	Substitution, 9,10
9	$Ref_2(p) = Ref_1(p) \cup Ref_1(Ref_1(p))$	Ref Axiom 8
10	$Ref_2(p) = \{l\} \cup Ref_1(l)$	Substitution 11,12
11	$Ref_1(l) = \{l\}$	See Previous Proof
12	$Ref_2(p) = \{l\} \cup \{l\}$	Substitution 13,14
13	$Ref_2(p) = \{l\}$	Defn of $\cup$
14	$Ref_2(p) = Ref_1(p)$	Subst. 11,16
15	$(Ref_2(p) = Ref_1(p)) \supset (REF(p) = Ref_1(p))$	Defn of REF
16	$REF(p) = Ref_1(p)$	MP 17,18
17	$REF(p) = \{l\}$	Subst. 14,19
18	$\neg(p \in REF(p))$	Defn of $\in$
19	$Eval(p)$	Eval 1

Then we use the Strengthened T-Schema and the fact that $N(l, \neg Tr(l)) \vdash \neg Tr(l)$ was proved in the proof for Theorem 20 to prove that $Tr(p)$:

1	$\neg Tr(l)$	Previous Proof
2	$Eval(p)$	Previous Proof
3	$N(p, \neg Tr(l)) \supset (Tr(p) \equiv (\neg Tr(l) \wedge Eval(p)))$	N-Str T-Schema
4	$Tr(p) \equiv (\neg Tr(l) \wedge Eval(p))$	MP ln 1,3
5	$\neg Tr(l) \wedge Eval(p)$	$\wedge$ Intro
6	$Tr(p)$	MP 2, 22

□

Thus, we can prove that $Tr\ulcorner\neg Tr(l)\urcorner$, so long as $\ulcorner\neg Tr(l)\urcorner$ is not l. That is, we can prove that "l is not true", so long as the name of "l is not true" is not l. This fact will allow the system to be semantically closed in the way required. All of the other results proven in $\mathcal{SLT}_1$ hold for $\mathcal{SLT}_2$, such as the fact that Curry's Paradox also does not bite. However, the key question is whether $\mathcal{SLT}_2$ is still consistent.

6.5 Consistency

In Chapter 5, the consistency of $\mathcal{SLT}_1$ was proven by constructing a model of $\mathcal{SLT}_1$. This model was more complicated than is normally the case as it included information about what sentences refer to what. To be more precise, the model contained information about what names referred to what names, but as each sentence has a single name in $\mathcal{SLT}_1$, this is equivalent. As $\mathcal{SLT}_2$ builds on $\mathcal{SLT}_1$, it is unsurprising that we will adopt the same approach to proving consistency. Building a model for $\mathcal{SLT}_2$ however involves including even more information in the model.

The key difference between $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ is that the truth bearers in $\mathcal{SLT}_2$ are name/sentence pairs, rather than sentences. It is worth pointing out though that we could interpret the truth bearers in $\mathcal{SLT}_1$ as being name/sentence pairs. As $\mathcal{SLT}_1$ has the restriction that every sentence has one unique name, the name/sentence pairs uniquely correlate with the sentences, and so there is no real formal difference between treating sentences and name/sentence pairs as truth bearers in $\mathcal{SLT}_1$. In $\mathcal{SLT}_2$ however there is a significant difference as we allow there to be non-identical name/sentence pairs with the same sentence. That is, one sentence can have multiple names, and the difference names have consequences for the semantic properties of the name/sentence pairs.

This means that in the model of $\mathcal{SLT}_2$, it is necessary to assign truth values to name/sentence pairs rather than sentences. This is a significant conceptual change, and requires some care in the definition of the model, but the basic structure of the definition is identical to the definition of the model for $\mathcal{SLT}_1$.

6.5.1 Building a Model for $\mathcal{SLT}_2$

As in the case of the model for $\mathcal{SLT}_1$, we will construct the model for $\mathcal{SLT}_2$ out of two smaller models for fragments of the language. The first sub-model of $\mathcal{SLT}_2$ would obviously be the standard model for $\mathcal{SL}$, since all of the theorems of $\mathcal{SL}$ are theorems of $\mathcal{SLT}_2$, except that the standard model of $\mathcal{SL}$ has sentences as truth bearers rather than name/sentence pairs. The second sub-model will be a model that deals with the definition of Evaluability within the model. To do this, we will construct a model of the Name Calculus part of $\mathcal{SLT}_2$. Once we have a model for the Name Calculus, the model for $\mathcal{SLT}_2$ can be built from the union of these two sub-models, in the same way as the model for $\mathcal{SLT}_1$ is built.

The key fact about $\mathcal{SLT}_2$ is that it is name/sentence pairs, rather than sentences, which have truth values. However, in the definition of $\mathcal{SLT}_2$, this distinction is only relevant in the scope of the relevant predicates. We explicitly prove sentences, rather than name/sentence pairs in $\mathcal{SLT}_2$. The role of a semantic model is to demonstrate the consistency of the system of proof in the language, which means that we only need to be concerned with true sentences in the model, rather than name/sentence pairs. This means that we can use the standard model for $\mathcal{SL}$ as a sub-model for $\mathcal{SLT}_2$ and that we only need to be concerned about name/sentence pairs in the model of the Name Calculus.

6.5.1.1 A Model for the Name Calculus

Nevertheless, a model for the Name Calculus is not any more difficult to construct than the case of $\mathcal{SLT}_1$. We will think about it is as a collection of points and arrows which connect points. The points are names/sentence pairs, and an arrow from one point to another means that the sentence at the first point refers to the sentence at the second point by means of the name at the second point. In the context of $\mathcal{SLT}_2$, this means that the name at the second point occurs within the scope of the Tr predicate within the sentence at the first point.

Points may have have none, one or finitely many arrows leading away from them, and every point will have many arrows pointing towards it. For any p, there are many sentences $Tr(p)$; $\neg Tr(p)$; and $Tr(p) \wedge Tr(q)$ that refer to it. As every name in $\mathcal{SLT}_2$ identifies a sentence these arrows are well defined in the model. The set $Ref_i(p)$ is the set of points which one can get to by following up to i arrows from p. The set $REF(p)$ is the complete set of points that one can get to by following arrows from p. If we follow the definitions through, a point (i.e. name/sentence pair) in this model will be evaluable if it has a finite REF set, and p itself is not in the REF set. That is, one cannot follow an arrow away from p and end up back at p via other arrows. We will refer to a series of arrows which follow from each other at points as a path. This definition is almost identical to the definition for $\mathcal{SLT}_1$, and so has the same properties. In particular, every name/sentence pair is either evaluable or not evaluable; and the evaluable name/sentence pairs are split between those that are grounded and those which are not grounded.

In order to make the differences between this definition and the one for $\mathcal{SLT}_1$ clear, it is useful to consider the situation where there are two name/sentence pairs which include the same sentence. The first point to note is that the arrows from this point depend purely on the syntax of the sentence. Therefore the two points which represent these two name/sentence pairs will have arrows to exactly the same points. This means that the reference structure of these two points is identical, after the first arrow. This means in particular that if one of these points is grounded, then the other must also be grounded.

However, it does not mean that if one of these two points is evaluable, then both of them are evaluable. We will take the sentence $Tr(a)$ and pair it with the names a and c to see how this works. In this case, both of the points ($< a, Tr(a) >$ and $< c, Tr(a) >$) have a single arrow that goes to the point $< a, Tr(a) >$. This means that the Ref_i sets of both of the points are identical to $\{< a, Tr(a) >\}$, which will equal the REF set for each of the points. However, $< a, Tr(a) >$ is in its REF set, and so is not evaluable. On the other hand, $< c, Tr(a) >$ is not in its REF set, and it is therefore evaluable.

Despite this change, it is still easy to see that this model is consistent. There cannot be a situation where a point is both evaluable and not, since one can either follow arrows away from a point and back to it, or one cannot. Therefore the model of the name calculus is consistent.

6.5.1.2 A Model for $\mathcal{SLT}_2$

We will construct the model for $\mathcal{SLT}_2$ in the same way as for $\mathcal{SLT}_1$, that is by using the two sub-models to define a base and then use a recursive definition on top of the base. To repeat, we generate the model for $\mathcal{SLT}_2$ by the following steps:

The Base

1. Include all of the true sentences from the models of $\mathcal{SL}$ and the Name Calculus in the set $\mathcal{T}_0$ in the base; and include all of the false sentences from the models of $\mathcal{SL}$ and the Name Calculus in the set $\mathcal{F}_0$ in the base.

2. For any sentence for which $\neg Eval(p)$ is true (i.e. in $\mathcal{T}_0$) in the base, add $Tr(p)$ to $\mathcal{F}_0$.

3. Close both $\mathcal{T}_0$ and $\mathcal{F}_0$ by using the following recursive definitions for the connectives in $\mathcal{SL}$:

   (a) If P is in $\mathcal{F}_0$ (or $\mathcal{T}_0$), add $\neg P$ to $\mathcal{T}_0$ (or $\mathcal{F}_0$).

   (b) For all other connectives use the standard recursive definitions.

The Recursive Definition

1. We begin with $\mathcal{T}_n$ and $\mathcal{F}_n$, and add elements to form $\mathcal{T}_{n+1}$ and $\mathcal{F}_{n+1}$ according to the following.

2. We add $Tr(p)$ to $\mathcal{T}_{n+1}$ (or $\mathcal{F}_{n+1}$) for every p that is in $\mathcal{T}_n$ (or $\mathcal{F}_n$), unless p is not evaluable (i.e. unless $\neg Eval(p)$ is true in the base).[2]

3. $\mathcal{T}_{n+1}$ and $\mathcal{F}_{n+1}$ are closed by using the same recursive definitions of the connectives as above.

If we take the union of all levels on this definition, we will get a model for $\mathcal{SLT}_2$and every sentence in $\mathcal{SLT}_2$ will have a truth value. The T-free sentences in $\mathcal{SLT}_2$ will all have a truth value in the base, so they are not a problem. The truth value of any sentence which contains the truth predicate will depend on whether the name/sentence pair within the scope of the truth predicate is evaluable or not. However, every name/sentence pair is either evaluable or not in the model, and so every sentence containing a truth predicate will have a truth value in the model of $\mathcal{SLT}_2$. If this model is consistent, and it is genuinely a model, i.e. everything that is provable in $\mathcal{SLT}_2$ is true in this model, then $\mathcal{SLT}_2$ is consistent.

[2] For the definition of $\mathcal{T}_1$ and $\mathcal{F}_1$, $\mathcal{T}_{n-1}$ and $\mathcal{F}_{n-1}$ are not defined and so are taken to be empty sets.

6.5.2 Is the model consistent?

The argument that the model is consistent is the same as the argument for $\mathcal{SLT}_1$, however it will be repeated here for completeness. We will follow the two stage construction of the model and use mathematical induction in order to show that the model is consistent. The key thing in showing that the model is consistent is showing that the sets $\mathcal{T}_i$ are consistent, that is there is no sentence such that both P and $\neg P$ are in $\mathcal{T}_i$. If the $\mathcal{T}_i$ are consistent then the $\mathcal{F}_i$ will also be consistent, as the negation rules guarantee that any inconsistency in one is transferred to the other. To do this, we will that the base ($\mathcal{T}_0$) is consistent, and then show that if at any level n, $\mathcal{T}_n$ is consistent, then at level $n+1$, $\mathcal{T}_{n+1}$ is also consistent.

The two sub-models are obviously consistent, so we must first check that adding $\neg Tr(p)$ to $\mathcal{T}_0$ for every sentence which is not evaluable preserves consistency. All of the sentences in $\mathcal{SL}$ and the Name Calculus are T-free, hence they are evaluable. This is because a T-free sentence does not refer to another sentence and hence is grounded. This means the set of sentences that are true in the sub-models and the set of not evaluable sentences are mutually exclusive sets. Therefore, there will be no sentence q that is both true in one of the two sub-models, and the sentence $\neg Tr(q)$ is added to $\mathcal{T}_0$ by Step 2 of the definition. This means that $\mathcal{T}_0$ is consistent before the closure over the connectives.

The only connectives that could cause trouble is negation, since the recursive rules for the other connectives cannot introduce a contradiction. The key point about the negation rule, is that it ensures that the negation of every false sentence is true and that the negation of every true sentence is in $\mathcal{F}_0$. The issue therefore is whether there is some sentence which is in $\mathcal{F}_0$, but whose negation is inconsistent in $\mathcal{T}_0$; or some sentence in $\mathcal{T}_0$, whose negation is inconsistent in $\mathcal{F}_0$. We know that the submodels are consistent and do not share vocabulary, therefore the only possibility must involve the sentences of the form $\neg Tr(p)$ which were introduced into $\mathcal{T}_0$ for p that are not evaluable. However, as argued, sentences that are not evaluable and sentences from the sub-models are mutually exclusive. Therefore the negation rules cannot introduce any inconsistency. This means that $\mathcal{T}_0$ is consistent.

We will now assume that T_n is consistent for some n. The second step of the recursive definition introduces sentences of the form $Tr(p)$ into $\mathcal{T}_{n+1}$. If T_n are consistent, the introduced sentences will be consistent, which means that the introduced sentences will also be consistent when closed under negation. Closing under the other connectives in classical logic cannot introduce an inconsistency if the starting set of sentences is consistent.

This means that if $\mathcal{T}_n$ is consistent, $\mathcal{T}_{n+1}$ can only be inconsistent if one of the introduced sentences contradicts a sentence that already has a truth value in level n (i.e. is in either $\mathcal{T}_n$ or $\mathcal{F}_n$). We will go through the different cases identified with regards to Evaluability to check that none of these can introduce an inconsistency.

The different cases all depend on the a key insight. In the model for the name calculus, the reference structure for each sentence is represented by arrows leading from that sentence, which sets up a complicated network when all sentences are considered. The important aspect to the network when considering evaluability is the tree leading from any particular sentence. If the tree and all paths in the tree are finite, the sentence is grounded. If the tree includes finitely many sentences, but has infinite paths (i.e. loops), it is evaluable if the original sentence is not in any loop and not evaluable if the original sentence is included in a loop. If the tree includes infinitely many sentences, it is not evaluable.

Not evaluable sentences are the easiest case to deal with. $Tr(p)$ for a p that is not evaluable is not introduced at any level apart from the base, due to the condition on the second step. Hence these sentences cannot introduce an inconsistency in the recursive definition.

We will consider the situation with grounded sentences next. For every grounded sentence, there will be a longest path in the tree leading from it. For a T-free sentences, that path length will be 0, and every sentence whose longest path length is 0 will be T-free, for any sentence that includes the truth predicate will refer to another sentence and so will have a non-zero maximum path length. This means that any sentence with maximum path length 0 will be in the Base.

Any sentence with maximum path length 1 will have at least one part of the form $Tr(p)$ where p is T-free, and no parts of the form $Tr(q)$ where q is not T-free. This means that every sentence with maximum path length 1 will be introduced into the model by the recursive definition in level 1, i.e. $\mathcal{T}_1$ and $\mathcal{F}_1$, and not in any earlier level. One can easily see that this pattern will continue, and that all sentence with maximum path length n will be introduced at level n, and not at any earlier level.

Now since $\mathcal{T}_{n-1} \subset \mathcal{T}_n$, the introduction rule ($Tr(p)$ for every p in $\mathcal{T}_n$), every sentence of the form $Tr(p)$ in $\mathcal{T}_n$ will be 'reintroduced' into $\mathcal{T}_{n+1}$. Since these are sets this does not affect any properties of the sets. The key thing to focus on are the sentences that are introduced by the recursive definition at level n, which are not introduced at any previous level. We know that every sentence with maximum path length n in the name calculus model are introduced at level n. However, we also know that every sentence with maximum path length less than n are introduced at a smaller level and those with maximum path length greater than n are introduced at a higher level. This means that the 'new' introductions at any level n are exactly those with maximum path length n (for grounded sentences). This means that no grounded sentence introduced at level n can be in contradiction with a sentence introduced at a lower level, since the sentences with smaller maximum path length are a distinct set of sentences. This demonstrates that grounded sentences cannot introduce an inconsistency.

The final category to consider are sentences that are evaluable, but are not grounded. Similar considerations apply to grounded sentences, except that we measure the maximum length to a not evaluable sentence in which none of the sentences in the path are themselves not evaluable. In this case, the Base will contain all sentences with maximum path length 1, and level 1 will introduce all sentences with maximum path length 2 and so on. However, as in the case with grounded sentences, the set of sentences introduced at each level is distinct, and so no sentence can be introduced at a level which contradicts a sentence at a lower level. Therefore no sentence that is evaluable but not grounded can introduce an inconsistency in the recursive hierarchy.

This exhausts the cases, and hence it follows that $\mathcal{T}_{n+1}$ is consistent, if $\mathcal{T}_n$ is consistent. By induction, every level of the hierarchy is consistent and the model is therefore consistent.

6.5.3 Is every provable sentence true in the model?

We must again follow the two stage construction of the model to show that every provable sentence is true in the model.

The first step is to note that as the models for $\mathcal{SL}$ and the Name Calculus are part of the base, all provable sentences in these two parts of $\mathcal{SLT}_2$ are automatically in the model. The only part of $\mathcal{SLT}_2$ that is not in one of these is the Strengthened T-Schema. If we can show that the Strengthened T-Schema holds for every sentence in $\mathcal{SLT}_2$, the fact that we close the model under the standard recursive definition for the connectives at each level means that all of the classical consequences from the Strengthened T-Schema will also hold in the model. This will mean that everything provable in $\mathcal{SLT}_2$ is true in the model.

Every name/sentence pair in $\mathcal{SLT}_2$ is either evaluable or it is not evaluable. For every non-evaluable b, both $\neg Tr(b)$ and $\neg Eval(b)$ are true in the base, so $Tr(b) \equiv (b \wedge Eval(b))$ is true by the recursive definition of the connectives since both sides are false. This means that the Strengthened T-Schema is true in the base for every sentence that is not evaluable.

If c is evaluable, $Eval(c)$ is true in the base. Also it means that either c is grounded, in which case every path from it ends with a T-free sentence, or that at least one path ends in a circular loop which does not contain c. If c is grounded, then every path ends with a T-free sentence, but every T-free sentence in $\mathcal{SLT}_2$ is a member either of $\mathcal{SL}$ or the Name Calculus. All the T-free sentences will have a truth value fixed in the base, and the recursive process will only transmit this up to c and $Tr(c)$. The recursive definition will mean that either c and $Tr(c)$ are both true or are both not true. In both cases the Strengthened T-Schema will hold as $Eval(c)$ is true. If c is not grounded, and has a path that ends in a circular reference loop, all of the name/sentence pairs in that circular loop will be not evaluable and hence defined as not true in the base. The relevant values will transmit up the levels and recursive definition will mean that either c and $Tr(c)$ are both true or are both not true. Again in both cases the Strengthened T-Schema will hold as $Eval(c)$ is true.

This means that the Strengthened T-Schema is true in the model for every sentence in $\mathcal{SLT}_2$

and hence every provable sentence in $\mathcal{SLT}_2$ is true in the model.

6.6 Summary

The formal truth definition in this chapter is an improvement on the definition in the previous Chapter, since we have provided a consistent truth definition for Classical Sentential Logic that can be plausibly semantically closed. To confirm this, we need to firstly check that the truth definition is adequate in the senses defined above.

Fact 22. *$\mathcal{SLT}_2$ provides an adequate truth definition for $\mathcal{SL}$ in the sense defined above.*

Proof. The first condition is satisfied by Lemma 15. The second condition is satisfied by Theorem 19 and the third condition is satisfied by the consistency proof. □

Thus the truth definition is $\mathcal{SLT}_2$ is adequate for $\mathcal{SLT}$, and it can also be shown to be adequate for itself, assuming that $\mathcal{SLT}_2$ is sufficiently liberal with naming. As argued above, for any sentence, it is always in principle possible to find a name/sentence pair for that sentence which is evaluable. The reason is that a name/sentence pair is not evaluable if it is in at least one circular reference loop. Given that a name/sentence pair can only exist within finitely many circular reference loops within $\mathcal{SLT}_2$, and each loop can have at most finitely many names, then assuming there are infinitely many names, there will always be some name outside the reference loops. This means that, for any sentence, it is always possible to find a name which would be evaluable in a pair with that sentence. We will assume in what follows that the definition of the naming axioms in $\mathcal{SLT}_2$ ensures that for every sentence there is a name/sentence pair which is evaluable.[3]

Fact 23. *$\mathcal{SLT}_2$ provides an adequate truth definition for itself in the sense defined above.*

Proof. The second condition, that $\mathcal{SLT}_2$ is as consistent as the T-free part of $\mathcal{SLT}_2$ is satisfied by the consistency proof, since both the T-free part and $\mathcal{SLT}_2$ are consistent.

The first condition is that if $\mathcal{SLT}_2 \vdash P$, then for some name p, p is a valid name of P and $\mathcal{SLT}_2 \vdash Tr(p)$. We know that it is possible to find some name q, such that $\mathcal{SLT}_2 \vdash N(q, P)$, and that q is evaluable. For this name/sentence pair, q/P, the standard T-Schema holds. This means that if $\mathcal{SLT}_2 \vdash P$, then it follows that $\mathcal{SLT}_2 \vdash Tr(q)$, as required. Hence the first condition is satisfied. □

The major motivation for developing $\mathcal{SLT}_2$ was that $\mathcal{SLT}_1$ was not semantically closed, since there was a sentence which was intuitively true and provable, but which could not be asserted as true within $\mathcal{SLT}_1$. The fact that $\mathcal{SLT}_2$ provides an adequate truth-definition for itself means that this precise problem cannot occur. For every provable sentence in $\mathcal{SLT}_2$, it is possible to assert

[3] We could alternatively amend the definition of $\mathcal{SLT}_2$ to allow the introduction of Naming Axioms within proofs in such a way that the name/sentence pair created is evaluable.

that that sentence is true. However, this does not guarantee semantic closure. We need to show that it is possible to assert that untrue sentences are not true in $\mathcal{SLT}_2$. To see this, we will prove the following Lemma:

Lemma 24. *If $\mathcal{SLT}_2 \vdash \neg P$ then there is some sentence p such that $\mathcal{SLT}_2 \vdash N(p, P)$ and $\mathcal{SLT}_2 \vdash \neg Tr(p)$.*

Proof. For any P in $\mathcal{SLT}_2$, we know that there is some p such that $\mathcal{SLT}_2 \vdash N(p, P)$ and p is evaluable. It follows, by Lemma 8, that $\mathcal{SLT}_2 \vdash Tr(p) \equiv P$. However, since $\neg P$ is provable, it follows that $\mathcal{SLT}_2 \vdash \neg Tr(p)$. □

Fact 25. *$\mathcal{SLT}_2$ is semantically closed for the truth predicate.*

Proof. From the previous two proofs, we have shown both that if P is provable then $Tr(p)$ is also provable, for some p which is a name of P; and that if $\neg P$ is provable then $\neg Tr(p)$ is also provable, for some p which is a name of P. Hence $\mathcal{SLT}_2$ is semantically closed. □

Thus $\mathcal{SLT}_2$ satisfies the full list of desiderata for a formal truth definition. It both offers a formal truth definition for $\mathcal{SL}$ and itself, it is consistent and is semantically closed for the truth predicate. Something obviously had to be given up to reach this. What has been given up in comparison to other approaches to the formal definition of truth are the T-Schema (for particular sentences), and the idea that sentence types are truth bearers, at least for the same particular sentences. These two changes have nevertheless been well-motivated, fit with our intuitions and only apply to certain sentences.

Importantly, the machinery that allows this to occur does not depend on classical logic, but can be defined in any sufficiently expressive logic. This has a number of consequences, with the most significant being that this approach to the definition of truth should apply equally well to natural languages. As noted, the techniques adopted are naturally expressed in natural languages, and the idea of taking the name/sentence pair as a truth bearer (at least in certain circumstances) is a natural response to certain natural language situations.

This also means that while $\mathcal{SLT}_2$ plausibly embodies the correct formal approach to the definition of truth in natural languages, it does not follow that $\mathcal{SLT}_2$ itself is the correct logical analysis of truth in natural languages. The application of this definition to natural languages will be discussed further in the next chapter. What is most significant about this Chapter is that the formal goal articulated in the opening chapters has been achieved, we have a formal truth definition which satisfies all of conditions identified previously that hold for a natural language, and which is consistent and semantically closed.

Chapter 7

Application to Natural Languages

The truth definitions in the previous two chapters have provided consistent formal truth definitions within classical logic including one that, in the case of $\mathcal{SLT}_2$, is semantically closed. By doing this, the approach to the definition of truth that has been adopted appears to meet all of the requirements to be a philosophically satisfactory truth definition. That is, it allows a consistent formal truth definition that does not rely on changing the grammar of the language or restricting the grammatically acceptable sentences. This means that it has the potential to be applied to our understanding of truth in natural languages. However, before we discuss this potential and how far it can be applied, it is worth backtracking a little and going over aspects of the definitions to gain a better understanding of how they work, particularly in the light of the analysis in the first three chapters.

7.1 Understanding the Formal Definition

In Chapter 2, various solutions to the Liar Paradox were analysed against the nine conditions that characterise a logical language which is affected by the Liar Paradox. It is useful to measure the definitions in $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ against these conditions, to see how they fit in with the other approaches.

Both of the languages $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ have a method of expressing untruth (Condition 1) since they include a truth predicate and a negation (Condition 6), untruth can be predicated of sentences or sentence/name pairs (Condition 2) via names which allow sentences to refer to other sentences or sentence/name pairs (Conditions 3). There are, moreover, no restrictions on the way that sentences can be constructed using these (Condition 4). As per Chapter 2, this means that $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$satisfy all of the conditions that natural languages satisfy.

Out of the remaining conditions, a contradiction can be derived from the paradoxical sentences (Condition 9) and the *ex contradictio quodlibet* holds (Condition 5), as the base logic is a classical logic. Obviously, the T-Schema does not hold in either $\mathcal{SLT}_1$ or $\mathcal{SLT}_2$, so the eighth condition does not hold. As the T-Schema, and the assumptions about truth that it reflects, were identified as a key contributor to the Liar Paradox, the T- Schema is replaced by the Strengthened T-Schema in both systems.

However, $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ differ as to whether they satisfy the seventh condition. All sentences in $\mathcal{SLT}_1$ and all sentence/name pairs in $\mathcal{SLT}_2$ are either true or not true, however it is not possible to express this fact for every sentence within $\mathcal{SLT}_1$. This is the problem of semantic closure for $\mathcal{SLT}_1$ which was discussed in Chapters 5 and 6 and which lead to the development of $\mathcal{SLT}_2$. $\mathcal{SLT}_1$ is therefore similar to most of the solutions examined in Chapter 2 which also do not satisfy condition 7, although this is not philosophically ideal for reasons already discussed. $\mathcal{SLT}_2$ does satisfy Condition 7, since it is semantically closed.

Nevertheless, the fact that both $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ satisfy the first four conditions and therefore the conditions on a Grammar-Only language being affected by the Liar Paradox mean that they are both candidates for a plausible account of the truth predicate in natural languages. $\mathcal{SLT}_2$ has the significant advantage that it is semantically closed for the truth predicate, but requires the philosophical commitment to sentence/name pairs being primary truth bearers, rather than simply sentences or sentence types. It is not in the scope of this thesis to decide between these alternatives, as any choice depends on broader philosophical concerns which have not been considered.

7.1.1 Evaluability

While there is this difference between $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$, the key to these highly successful definitions is the Strengthened T-Schema and the definition of Evaluability. However, while the consistency of these languages depend crucially on the Strengthened T-Schema, they do not depend crucially on the particular definition of Evaluability that was adopted. There are potentially a range of definitions of Evaluability that will formally suffice, and any decision on the best definition of Evaluability must depend on non-formal considerations.

The clearest way of seeing that other definitions of Evaluability may suffice is to look at the fragment of $\mathcal{SLT}_1$ which includes the Strengthened T-Schema but does not include the Name Calculus. In this fragment, the *Eval* predicate is not defined outside of the Strengthened T-Schema. What is interesting about this fragment is that we can still prove that paradoxical sentences are not evaluable, even though *Eval* is not defined. The following Lemma is an example.

Lemma. *In the fragment of $\mathcal{SLT}_1$ without the Name Calculus, for the sentence $\neg Tr(l)$ (which has name l), it is provable that $\neg Eval(l)$.*

Proof. The proof simply consists of a simple derivation which we will break into two. Notice that no rules from the Name Calculus are required.

Firstly, we prove that $\neg Tr(l)$:

1	$Tr(l) \equiv (\neg Tr(l) \wedge Eval(l))$	Str. T-Schema
2	$Tr(l)$	assume
3	$\neg Tr(l) \wedge Eval(l)$	$\supset$E 1,2
4	$\neg Tr(l)$	$\wedge$Elim 3
5	$Tr(l) \wedge \neg Tr(l)$	$\wedge$Intro 2,4
6	$\neg Tr(l)$	$\neg$ Intro 2-5

We can also prove $\neg Eval(l)$:

1	$Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$	Str. T-Schema
2	$\neg Tr(l)$	See Previous Proof
3	$\neg(\neg Tr(l) \wedge Eval(l))$	Modus Tollens 1,2
4	$\neg\neg Tr(l) \vee \neg Eval(l)$	DeM 3
5	$Tr(l) \vee \neg Eval(l)$	Double Negation 4
6	$\neg Eval(l)$	Disj. Syll 2,5

□

Equivalent derivations obviously hold for other paradoxical examples, and this demonstrates the power of the Strengthened T-Schema. Paradoxical sentences can be judged to be not true simply as a result of the Strengthened T-Schema's logical structure. In fact, in this derivation, the *Eval* predicate could represent any concept, not necessarily Evaluability. So long as there is a second term on the right hand side of the Strengthened T-Schema, paradoxical sentences will turn out to be not true. This means that even if the idea of evaluability that has been adopted here is not correct, the basic structure of the truth definition can still hold and it can still be consistent.

The most important consequence of this fact from a formal perspective is that the Strengthened T-Schema shifts the responsibility for the consistency of the system from the Truth Predicate to the Evaluability Predicate (or whatever the second term on the right hand side is). If we extend the derivation above to all the different paradoxical cases, we can see that all of the paradoxical sentences will be provably not evaluable in these systems. This means that, so long as on the formal definition of evaluability all paradoxical sentences are not evaluable, the system will be consistent. The key to defining a consistent truth definition is therefore to define evaluability so that all of the paradoxical sentences are not evaluable. It is this fact that allows $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ to be consistent, since they both include a syntactically defined evaluability predicate which satisfies this.

We can formulate this a little more precisely as a criterion on a satisfactory definition of the Evaluability Predicate for any formal language ($\mathcal{L}$) which is being extended to include a truth definition ($\mathcal{LT}$):

Satisfactory *Eval*: The definition of *Eval* in $\mathcal{LT}$ is satisfactory iff whenever a is paradoxical it is the case that $\vdash \neg Eval(a)$ (in $\mathcal{LT}$).[1]

The exact definition of Evaluability used in $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ is simple to define, possible to intuitively motivate and Satisfactory on this definition. However, it is arguably more coarse grained than would be ideal. It errs on the side of caution and potentially identifies too many sentences as not evaluable. For example, we will consider (in $\mathcal{SLT}_1$) the sentence $\neg Tr(q) \vee (P \supset P)$ whose unique name is q. According to our definition of Ref, $Ref_1(q) = \{q\}$. This means that q refers to itself, and therefore that $\neg Eval(q)$. Thus, although the sentence $\neg Tr(q) \vee (P \supset P)$ is provable within the system, and it is intuitively evaluable as we can evaluate one of the disjuncts, it can be proven to be not evaluable and hence not true within $\mathcal{SLT}_1$.

This suggests that we need a more finely-grained formal definition of evaluability in order to capture our intuitions in these cases. This may be achievable, however, there is a significant obstacle in the way of an improved definition. Consider instead the sentence $\neg Tr(r) \vee (P \wedge \neg P)$ with name r. Again on the definition of $Eval$ in any of the defined systems, $\vdash \neg Eval(r)$ and hence $\vdash \neg Tr(r)$. In this case, however, the conclusion seems justified, since although one disjunct is evaluable, the truth value of the sentence depends on the unevaluable disjunct. Thus although Q and R share a very similar form, the truth values of the disjuncts appear to introduce differences in evaluability.

It is obviously not possible to deal with this in a purely syntactic way as was provided previously, as there would need to be some sort of recursive definition of sentences which takes into account both evaluability and truth value. There is also the potential for problems with circular definitions in this case, since Truth (via the Strengthened T-Schema) depends on Evaluability which for some sentences depends on the Truth Values of sentences. It is probably possible to devise a definition of evaluability along these lines, and so long as it is Satisfactory on the definition above, it will work in a formal truth definition. Even if successful, it would make the conceptual task of articulating the relevant concept of evaluability far more difficult.

There is an alternative approach to dealing with this problem implicit in $\mathcal{SLT}_2$, which offers a way around this issue. If we consider the sentence $\neg Tr(q) \vee (P \supset P)$ with a name other than q, in $\mathcal{SLT}_2$ this sentence/name pair is evaluable and true, as is intuitively correct. With q as the name, the sentence/name pair is not true due to the circular reference, but we can still say that the relevant sentence is true using a different name. Similarly, the sentence $\neg Tr(r) \vee (P \wedge \neg P)$ with a different name is evaluable and true, since it is the case that $\neg Tr(r)$. Allowing multiple names for one sentence allows us to preserve the intuitions without having to provide a far more complicated definition of evaluability.

[1] There is a formal definition of paradoxical as part of the proof in the Appendix.

7.1.2 Bivalence

One of the most remarkable features of this truth definition is that it can be carried out within a classical bivalent logic. However, one of the lessons of the Liar Paradox is very often taken to be that classical bivalence is untenable and either we must accept truth value gaps or gluts. This raises the question, which has been studiously avoided so far, of whether the truth definition offered here is bivalent. The key issue is what is meant by bivalence.

The most trivial point is that the underlying logic is bivalent, and that the Law of the Excluded Middle holds. This means that for any sentence P, $Tr\ulcorner P\urcorner \vee \neg Tr\ulcorner P\urcorner$ is true. Furthermore, for every sentence that $Eval\ulcorner P\urcorner$ is defined, and if P is either provable or disprovable, then $Tr\ulcorner P\urcorner$ is either provable or disprovable. Thus all of the relevant sentences are either true or not true, and this truth definition is obviously bivalent in this sense.

However, bivalence is normally taken to be the position that every sentence is either true or false. The question of whether $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ are bivalent in this sense depends on what we understand falsity to be. There are two standard ways of understanding what it means for a sentence to be false: either the sentence is not true, or its negation is true. These two conceptions of falsity are however not identical in these systems. To see this, we will consider two possible falsity definitions within $\mathcal{SLT}_1$:

1. $F_1\ulcorner P\urcorner \equiv \neg Tr\ulcorner P\urcorner$
2. $F_2\ulcorner P\urcorner \equiv Tr\ulcorner \neg P\urcorner$

On the first definition of falsity, every sentence is either true or false. Within the language, we know that $Tr\ulcorner P\urcorner \vee \neg Tr\ulcorner P\urcorner$ is true as it is a tautology of Classical Logic (and the language is consistent). Moreover, within the model every sentence will be either true or false. However, there are some unintuitive consequences if we accept this definition as a definition of falsity. We consider whether the sentence with name l in $\mathcal{SLT}_1$ is false on the first definition. It should be noted that $\vdash P$ is understood here as meaning that P is provable.

It was shown in Chapter 5 that $\vdash \neg Tr(l)$ in $\mathcal{SLT}_1$, so by definition, $F_1(l)$ is provable. In other words, $F_1\ulcorner \neg Tr(l)\urcorner$ is provable. However, we now consider the sentence $Tr(l)$. By the Str. T-Schema, $Tr\ulcorner Tr(l)\urcorner \equiv Tr(l) \wedge Eval\ulcorner Tr(l)\urcorner$. $Tr(l)$ is not self-referring, and so $Eval\ulcorner Tr(l)\urcorner$ is true and the Str. T-Schema reduces to the normal T-Schema: $Tr\ulcorner Tr(l)\urcorner \equiv Tr(l)$.

However, by the Str. T-Schema for l, we have $Tr(l) \equiv \neg Tr(l) \wedge Eval(l)$. Hence, by equating the two equivalences, we get $Tr\ulcorner Tr(l)\urcorner \equiv \neg Tr(l) \wedge Eval(l)$. We know that $Eval(l)$ is false, and hence we can conclude that $\neg Tr\ulcorner Tr(l)\urcorner$. By the definition of F_1, this means that we have proven $F_1\ulcorner Tr(l)\urcorner$. Thus we have proven that there is a sentence in $\mathcal{SLT}_1$ such that both it and its negation are false. This seems to undermine the concept of negation.

The situation in $\mathcal{SLT}_2$ is more intricate but not so counter-intuitive. $F_1(l)$ is also provable within $\mathcal{SLT}_2$, however, if s is also a name for the sentence $\neg Tr(l)$, then we can prove that $Tr(s)$and therefore $\neg F_1(s)$. To put it differently, in $\mathcal{SLT}_2$ the pair $l/\neg Tr(l)$ is false (on the first definition) but the pair $s/\neg Tr(l)$ is true. It is also the case in $\mathcal{SLT}_2$ that $Tr(l)$ is false (on the first definition), but it does not follow that its negation is false. It depends on which name/sentence pair we are looking at. Moreover, where truth values depend on the name of a sentence, it is plausible that we should take an evaluable name/sentence pair as providing the correct truth value for the sentence. If we adopt this principle, then $Tr(l)$ is false and $\neg Tr(l)$ is true, as expected.

Thus if we take the first definition of falsity, every sentence is both systems are either true or false, however we need to be careful about how we understand falsity in certain cases as there are some counter-intuitive consequences. In particular, in $\mathcal{SLT}_1$ there are cases where both the sentence and its negation are false.

If we take the second definition of falsity, it turns out that the language is not bivalent as there are sentences which are neither true nor false. Unsurprisingly, one such sentence is the Liar Sentence. We have already proven that it is not true, and the proof that it is not false (on the second definition) follows the example just outlined for the first definition.

We have proven $\neg Tr(l)$, and we want to show that $\neg F_2(l)$. Since l is the sentence $\neg Tr(l)$, that means that we need to show $\neg Tr\ulcorner\neg(\neg Tr(l))\urcorner$, by the definition of F_2. We can easily see that $\neg\neg Tr(l)$ is not self-referring, and so $Eval\ulcorner\neg\neg Tr(l)\urcorner$ is provable. This means that the strengthened T-Schema reduces to the normal T-Schema: $Tr\ulcorner\neg\neg Tr(l)\urcorner \equiv \neg\neg Tr(l)$. In both $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$, $\neg Tr(l)$ is provable, and hence we can prove from this biconditional that $\neg Tr\ulcorner\neg\neg Tr(l)\urcorner$. Thus we have proven that $\neg F_2(l)$, and the Liar Sentence is neither true nor false on this definition. That is, there is a sentence that is not true and its negation is also not true.

It is important to note is that it is necessary to be precise about how we understand this in $\mathcal{SLT}_2$. In $\mathcal{SLT}_2$ the name/sentence pair $l/\neg Tr(l)$ is not true, and any name/sentence pair $a/\neg\neg Tr(l)$ is also not true. Thus the name/sentence pair $l/\neg Tr(l)$ is plausibly neither true nor false, on this second definition of falsity. However, this does not mean that any other sentence pair $s/\neg Tr(l)$ is neither true nor false (they are true), and nor does it mean that any sentence $t/Tr(l)$ is also neither true nor false. Any such pair $t/Tr(l)$ is not true in $\mathcal{SLT}_2$, however only one of the possible pairs of the form $s/\neg Tr(l)$ is also not true. Given this, it is plausible to argue that all pairs of the form $t/Tr(l)$ are simply false on this definition, as the negation (pairs of the form $s/\neg Tr(l)$) is true.

If we accept the second definition of falsity, then the systems are no longer bivalent in the sense that every sentence (or name/sentence pair) is true or false. Unsurprisingly, Liar Sentences come out as neither true nor false.

The aim of this discussion is not to decide between these two definitions of falsity. Each of the definitions has intuitive support and each is equally tenable in this context. The choice does not impact the validity of the truth definition. However, what is clear is that the two definitions are not equivalent in either of the two developed systems. There are sentences which are false on one definition but not false (and not true) on the other definition. This is a result of adopting the Strengthened T-Schema, and the concept of evaluability. Furthermore, we cannot accept that these two definitions are equivalent without accepting inconsistency, as has been shown elsewhere.[2]

7.2 Application to Natural Language

This thesis began with the observation that, if the Liar Paradox is a serious problem, it means that it is not possible to consistently use natural languages to assert truth of sentences within natural languages. This arose because certain properties of natural languages, combined with intuitive principles about truth and the relation between truth and language, lead to a necessary inconsistency. The nine conditions identified in Chapter 1 articulated the relevant properties and principles.

The challenge in responding to the Liar Paradox was to find a solution that does not require us to change the grammar of natural languages, or to restrict what can be said truthfully in natural languages. Changing the grammar is an implausible solution and imposing restrictions on what can be said truthfully undermines any attempt to discuss the Liar Paradox and hence negates any solution. Nevertheless, in Chapter 2, it was argued that existing solutions have not accomplished this in a satisfactory way. Chapter 3 identified two key assumptions in standard modern formal semantics, the definition of all semantic properties directly with respect to the model, and that sentence types are truth bearers, which are incompatible with a satisfactory solution. The systems $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ were attempts to get around these limitations by following a new approach to the definition of truth, which was mainly articulated in Chapter 4.

This new approach is built on the observation that the pattern of what sentences refer to other sentences within a language has semantic consequences, when that reference occurs within semantic predicates. That is, the truth value of a sentence is affected both by what sentences it refers to, but also by what sentences refer to it. This fact requires an extra condition to be included in any truth definition, to take into account when this occurs. We adopted the concept of Evaluability to represent this idea, and defined the Strengthened T-Schema to capture the concept of truth.

This idea was implemented in $\mathcal{SLT}_1$, while in $\mathcal{SLT}_2$ this observation was taken further. If the reference pattern in a language affects truth values, it is to be expected that the method we use to identify sentences within the scope of predicates, whether that is names or definite descriptions or by speaker, may also affect the truth value. In $\mathcal{SLT}_2$ we therefore took name/sentence pairs as the primary truth bearers, rather than sentences alone. This allowed the system to be semantically

[2] See J. C. Beall and Octavio Bueno. “The simple liar without bivalence?” In: *Analysis* 62.1 (2002), pp. 22–26; and Adam Rieger. “The Liar, The Strengthened Liar, and Bivalence”. In: *Erkenntnis* 54 (2001), pp. 195–203.

closed in the way we intended, which demonstrates the value of implementing this observation.

Both of these systems are highly successful with respect to responding to the philosophical challenge, although only $\mathcal{SLT}_2$ is truly semantically closed and does not place any restrictions on what can be said truthfully in the language. Thus, we can say that the philosophical problem posed by the Liar Paradox has been solved, and we have an understanding of truth and reasoning in natural languages which allows us to consistently use natural languages to assert the truth of sentences.

It is important however to be careful about what exactly has been shown in this thesis. The approach to the definition of truth that was articulated in Chapter 4 and the start of Chapter 6 offers an understanding of truth and reasoning which is consistent with the way we use natural languages. This fact was demonstrated by the successful definitions of $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$. However, this does not mean that either $\mathcal{SLT}_1$ or $\mathcal{SLT}_2$ offer a complete or even satisfactory account of reasoning for natural languages. Each system has limitations with respect to its application to natural languages.

The most obvious limitation is that neither of these systems allow quantification, which obviously exists in natural languages. However extending either $\mathcal{SLT}_1$ or $\mathcal{SLT}_2$ to predicate logic is not necessarily a straightforward task. There are questions about domains of quantification that need to be addressed, and if complete applicability to natural languages is sort, then quantification over names and sentences needs to be introduced. This raises further questions about difference paradoxes which have not been addressed here. The flexibility of this approach adopted, particularly in the definition of Evaluability, suggests that a successful definition is very possible. However, without this, $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ have only limited application to natural languages.

A second limitation is that both $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ have more regimented and limited methods of referring to other sentences than natural languages do. In both of these, the method of reference is by means of explicit syntactic names. As this formal method of naming is syntactic, it has the strong advantage that it allows explicit definitions of properties of the reference relations (such as Evaluability) within the language. Natural languages however do not have a natural class of names for sentences and we generally do not use explicit names for sentences. The most common devices in natural languages are demonstratives (This sentence ....), definite descriptions (The first sentence on the page....) and name forming devices such as quotation marks. None of these devices are directly found within $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$, which is further evidence that these systems are limited in their application to natural languages.

It should be noted that it is possible to translate these devices into the systems $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$. It is easier to translate these devices into $\mathcal{SLT}_2$, since $\mathcal{SLT}_2$ mirrors natural languages in allowing more than one way to refer to a single sentence. For example, each definite description which denotes a sentence can be translated into $\mathcal{SLT}_2$ as a unique name. Also, the quotation name of a sentence can be translated into any evaluable name of that sentence.[3] Demonstratives are more difficult to translate, as they depend on their context for meaning. However, it is generally possible to translate demonstratives into $\mathcal{SLT}_2$, at least in the sense of translating the same reference pattern between the relevant sentences. To do this, however, it is necessary to assign names to the relevant sentences (or relevant sentence tokens) and then use those names within $\mathcal{SLT}_2$ in the place of the demonstratives.

Translating natural language sentences into $\mathcal{SLT}_1$ is more difficult, as each sentence is only allowed one name. This means, for example, that in the case of definite descriptions, one must identify the sentence being referred to and then translate the definite description by the name of that sentence. Quotation names are obviously easy to translate into $\mathcal{SLT}_1$, as each quote name is replaced by the canonical name of the sentence. These examples show that $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ have a place in the analysis of natural languages, but cannot be considered to be complete formal accounts of natural languages.

The limitations with $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ however do not call into question the success of the key principles used in the definition of the systems. The Strengthened T-Schema, the concept of Evaluability, and taking name/sentence pairs to be primary truth bearers, are concepts that are broader than these systems. The arguments for these principles did not depend on any features of $\mathcal{SLT}_1$ or $\mathcal{SLT}_2$, and they therefore can be used very broadly. The method by which these concepts prevent the Liar Paradox and allow semantic closure in particular do not depend on anything in particular in $\mathcal{SLT}_1$ or $\mathcal{SLT}_2$. These concepts therefore embody an approach to the definition of a truth predicate which is successful in solving the Liar Paradox in a satisfactory way.

There is however one final limitation to $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$, which was touched on in the discussion of Tarski's definition of truth. This is a significant limitation as it arguably applies to all modern languages, and raises questions about the typical formulation of the T-Schema and the Strengthened T-Schema. As noted in Chapter 2, the only sentences in formal languages that can be legitimately stated or asserted are provable sentences. The rules that govern assertion in formal languages are rules of derivation. In natural languages, on the other hand, sentences can be stated regardless of their provability. We expect only justifiable sentences to be asserted in many contexts, but there are no linguistic rules which prevent unjustifiable sentences from being asserted.

The formal language approach to assertability has a significant advantage with regard to formal truth definitions. In a formal language, all assertable sentences are provable, and therefore must be true. This means that we can use the assertability of a sentence as a guide to its truth, and assertability can be determined within the formal language. As a result of this, a truth definition does not need to refer to the semantic of the language, since provability is a guarantee of truth (and disprovability is a guarantee of untruth). In this formal context, schema such as the T-Schema or

[3] Quotation names can produce the Liar Paradox if we allow quotation names of predicates, as in the Quinean Liar. These are not allowed in $\mathcal{SLT}_1$ or $\mathcal{SLT}_2$, so there are no problems in these systems.

the Strengthened T-Schema can define a truth predicate. They tie $Tr\ulcorner P\urcorner$ to the provability of P, which works as provability guarantees truth.

Natural languages do not in general possess an uncontroversial concept of provability in the same way that formal languages do. While certain fragments of natural languages, say the fragment that deals with mathematics, plausibly have a clear concept of provability, in general we often do not have an explicit grasp on what it means to prove that, say “Snow is white”. There are various ways that one can go about justifying natural language statements, and different methods of justification are considered to be proof in different contexts. While this does not undermine the proposed approach to the Liar Paradox, it means that we need to be careful when using schema such as the T-Schema or the Strengthened T-Schema in natural language contexts.

If we are using either version of the T-Schema as a schema for generating truths, we need to think about what it means to show or prove that P. The derivation that $Tr\ulcorner P\urcorner$ depends on proving that P (and $Eval\ulcorner P\urcorner$ for the Strengthened T-Schema). Proving that P however does not mean the same thing in natural languages as it does in formal languages, and this has potential issues for our understanding of truth in natural languages. For example, at times we accept a level of proof in natural languages that does not imply complete certainty. If provability implies truth, this could in turn mean that we are accepting something as true which is not completely certain. Intuitions as to whether this is legitimate can go either way, which demonstrates that some caution and further consideration is required at this point. The fact that a T-Schema can be used to generate truths in formal languages does not necessarily mean that it explains truth in natural languages. More should be said and there is more work to do on this issue as it goes to the heart of how we understand the concept of truth and the relationship between natural and formal languages.

It should be noted however that this does not meant that the Strengthened T-Schema does not apply in natural languages. It very plausibly and consistently describes the logical relationship between $Tr\ulcorner P\urcorner$ and P.

In any case, these considerations do not undermine key achievements of this thesis. By changing the focus of analysis of the Liar Paradox onto languages, structural conditions were identified which ensure languages which satisfy these are trivialised by the paradox. A further structural analysis of formal semantics and truth definitions identified some key ideas that allowed consistent truth definitions within classical sentential logic with a very high degree of semantic closure. These key ideas, the Strengthened T-Schema, Evaluability and taking name/sentence pairs as truth values, are applicable well beyond the formal languages of $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$. They provide a structural understanding of truth and truth predicates which allows a clear, consistent and appealing resolution of the Liar Paradox.

7.3 Philosophical Consequences

It was noted in Chapter 4 that the focus of this thesis was on the formal definition of the concept of truth, not on any of the debates about the nature or correct theory of truth, or truth-bearers. Nevertheless, it is worth noting that, if correct, the approach to the formal definition of truth developed in this thesis has some significant consequences for these broader philosophical debates. These consequences can only be sketched here, and they raise many questions and areas for future work, but are well worth noting.

Firstly, with regards to theories of truth, the success of our approach undermines the popular theory of Deflationism. Roughly speaking, Deflationism about truth is that the truth predicate does not strictly mean anything, and the $Tr\ulcorner P \urcorner$ is simply another way of saying that P, and that these are always intersubstitutable. This theory obviously relies on the universal applicability of the Tarskian T-Schema, which has been rejected in this thesis with impressive results.

Adopting the Strengthened T-Schema requires that we accept there are two conditions on a sentence such as $Tr\ulcorner P \urcorner$ being true, rather that the single condition of P. It is necessary to identify both whether P and whether P is evaluable. If this is correct, then $Tr\ulcorner P \urcorner$ and P are obviously not always substitutable and it cannot be the case that the truth predicate is simply a device, as J. C. Be all as expressed it, "to enable generalizations that, given our finite constraints, we couldn't otherwise express."[4]

Secondly, the approach adopted for $\mathcal{SLT}_2$ which was the most successful definition in terms of semantic closure, has consequences for our understanding of truth bearers. The formal device adopted in Chapter 6 was to treat name/sentence pairs as truth bearers. This exact approach may not be the most appropriate for natural languages, particularly as sentences are not automatically assigned names in natural languages, but if this approach is adopted, it means that primary truth bearers must be sensitive to the way we refer to a sentence. There are various ways that this might be realised, perhaps through a token approach, perhaps through a particular definition of context and accommodates names or method of reference, or perhaps an account of truth bearers such that a sentence *qua* one method of reference has a different truth value to a sentence *qua* another. However, we cannot simply treat sentence tokens or sentence types as truth bearers, as examples above showed cases where the same sentence token or type could vary in truth value depending on who we referred to it.[5]

There is a further consequence of adopting the approach here that affects another debate in the philosophy of language. The definition of truth in both $\mathcal{SLT}_1$ and $\mathcal{SLT}_2$ were shown to be non-recursive for truth values. The truth value of sentence of the form $Tr\ulcorner P \urcorner$ did not depend purely on P. This means, in the philosophy of language, that the successful definition of truth

[4] J. C. Beall. *Spandrels of Truth.* Oxford University Press, 2009, p. 1

[5] See Chapter 6.

adopted here is not compositional for truth values. This means in turn, either that meaning is not compositional, or that meaning is compositional and that truth values do not always purely depend on meaning. Either of these approaches can be accommodated within the logical framework here, but the traditional view that meaning is compositional, meaning imposes truth conditions and therefore truth values are compositional cannot survive unaltered.

While the approach to the formal definition of truth developed in this thesis does not solve any of these broader philosophical issues, it rules out certain approaches to these issues and provides parameters that other approaches must fit within. There is significant work to be done following through the philosophical consequences of this approach more exactly, since, as noted previously, a complete solution to the Liar Paradox involves a complete philosophy of language, truth and reasoning.

7.4 Conclusion

It has been argued that the Liar Paradox arises as a side effect of the patterns of reference necessary to the definition of Truth as a predicate. The patterns of reference allow situations where it is not possible to follow the reference through to an evaluation of the truth value of the sentences involved. Once we take into account this possibility as a further condition on the definition of the truth predicate through the Strengthened T-Schema, a consistent understanding of truth is possible, and formally definable. The key problem identified with the Liar Paradox is that it brings into question our ability to use natural languages consistently in any discussion which involves truth. The analysis and formal definition in this thesis demonstrates that there is a consistent understanding of the truth predicate which is compatible with the way we normally want to use natural languages. To go back to the terminology from the Introduction, the approach to the definition of the truth predicate is a plausible candidate for the building $English^+$, the correct linguistic system which includes English and the correct assumptions about reasoning and truth. The fact that there is a consistent candidate for $English^+$ that resolves the Liar Paradox and allows us to assert the truth or falsity of any sentence in the language, ensures that the Liar Paradox no longer poses a problem for our use of natural languages.

Appendix

In Chapter 5, it was pointed out that it is possible to prove the consistency of $\mathcal{SLT}_1$ without using a standard soundness proof with respect to a model. This proof was not given in Chapter 5 as it is more technically and mathematically involved that the consistency proof offered, and tends to obscure the structure of the solution. However, for completeness, and as few proofs of this sort exist, the alternative consistency proof is presented here. It can be adapted to a proof of $\mathcal{SLT}_2$ with some minor modifications.

In order to make it more readable, the following proof uses a notational convention that was mentioned in Chapter 5. The name in $\mathcal{SLT}_1$ of a sentence denoted by a capital letter will be the lower case letter. For example, p is the name of P and a is the name of A.

Alternative Consistency Proof

The structure of this proof of the consistency of $\mathcal{SLT}_1$ starts from the observation that if we replace the Strengthened T-Schema with the ordinary T-Schema, then the language will be inconsistent. Given that the Strengthened T-Schema licenses certain instances of the ordinary T-Schema, namely those instances where the sentence is Evaluable, $\mathcal{SLT}_1$ can only be consistent if the definition of Evaluability succeeds in preventing the instances of the ordinary T-Schema which generate the paradox. This in turn requires some method of identifying the relevant sentences and instances of the T-Schema.

We will therefore begin by considering the language $\mathcal{SLT}_1^*$ which is identical to $\mathcal{SLT}_1$ except that it does not contain the Strengthened T-Schema. This means that $\mathcal{SLT}_1^*$ contains all of the same sentences as $\mathcal{SLT}_1$, but not all of the same theorems. The general method will be to consider the consequences of introducing various instances of the ordinary T-Schema ($Tr(a) \equiv A$) into $\mathcal{SLT}_1^*$. We will begin with a couple of Lemmas about $\mathcal{SLT}_1^*$:

Lemma 26. *If $\mathcal{SL}$ is consistent, then so is $\mathcal{SLT}_1^*$.*

Proof. $\mathcal{SLT}_1^*$ includes all of $\mathcal{SL}$, plus the rules governing the Name Calculus, i.e. the definition of Ref_n and REF, and the Introduction Rules the $Eval$ predicate. The definitions of Ref_n and REF simply define sets of sentences, and hence these, by themselves cannot introduce a contradiction.

The consistency of $\mathcal{SLT}_1^*$ therefore depends on the consistency of the introduction rules for the *Eval* predicate. That is, whether the following two rules are consistent:

Eval 1 $\quad a \in Ref_i(a) \vdash \neg Eval(a)$ $\quad$ for some i

Eval 2 $\quad \neg(a \in REF(a)) \vdash Eval(a)$

From these rules, we can see that the definition of the *Eval* predicate can only be inconsistent, if it is possible that there is some sentence such that both $a \in Ref_i(a)$ and $\neg(a \in REF(a))$ hold. However, the definition of $REF(a)$ is such that $Ref_i(a)$ is a subset of $REF(a)$ for all i. This means that it is not possible that both of these hold, and therefore the definition of the *Eval* predicate is consistent. Therefore $\mathcal{SLT}_1^*$ is as consistent as $\mathcal{SL}$ is. □

Lemma 27. *The only theorems in* $\mathcal{SLT}_1^*$ *which contain the Truth Predicate are classical tautologies.*

Proof. Since $\mathcal{SLT}_1^*$ does not contain any axioms or rules governing the Truth Predicate, any instance of $Tr(a)$ (for some a) can only be treated as if it is an atomic sentence. Therefore, the only theorems which contain Tr are classical tautologics (e.g. $Tr(a) \supset Tr(a)$) □

The first of these two Lemmas demonstrates that in order to prove that $\mathcal{SLT}_1$ is consistent, we only need to focus on the effects of introducing the Strengthened T-Schema into $\mathcal{SLT}_1^*$. The second plays a key technical role in the proof, since it establishes that $\mathcal{SLT}_1^*$ does not include any theorems of the form $Tr(a)$. This means that it is possible to isolate the effects of introducing the Strengthened T-Schema, and therefore demonstrate that $\mathcal{SLT}_1$ is consistent.

To do this, we will begin by comparing the Strengthened T-Schema with the Ordinary T-Schema, since introducing the Ordinary T-Schema into $\mathcal{SLT}_1^*$ will introduce a contradiction. It is important to note that not every instance of the Ordinary T-Schema is inconsistent. The point of the Strengthened T-Schema is that it is equivalent to the Ordinary T-Schema for evaluable sentences, and otherwise it implies that the relevant sentence is not true. If we can show that all of the 'problematic' instances of the T-Schema are not evaluable, then $\mathcal{SLT}_1$ can be shown to be consistent.

The first thing to note is that it is not necessary to introduce the ordinary T-Schema as a general axiom schema in order to generate a contradiction. Introducing particular subsets of instances of it as axioms will suffice. For example, if we introduce the single instance $Tr(l) \equiv L$ (for L as defined above), then the language is immediately inconsistent. This means that for any set of instances of the ordinary T-Schema that is inconsistent when assumed in $\mathcal{SLT}_1^*$, there will be a subset of this, whose instances are necessary for the derivation of a contradiction. That is, for a set Λ of instances of the ordinary T-Schema such that $\mathcal{SLT}_1^*, \Lambda \vdash \bot$, there will be some set $\Delta \subseteq \Lambda$, such that $\mathcal{SLT}_1^*, \Lambda \setminus \Delta \nvdash \bot$. A couple of points are important to note here. Firstly, the set Δ may be identical to Λ, for example in the case that $\Lambda = \{Tr(l) \equiv L\}$. However, since $\mathcal{SLT}_1^*$ is consistent, we know that for any Λ which satisfies this property, there is a corresponding non-empty Δ.

The first step in proving the consistency of $\mathcal{SLT}_1$ is identifying the particular subsets of instances of the T-Schema will generate a contradiction. As $\mathcal{SLT}_1^*$ is consistent, and In particular, since $\mathcal{SLT}_1$ only allows finite proofs, we only need to consider finite subsets of instances of the normal T-Schema. In turn, once we have identified which subsets generate a contradiction, it is possible in turn to identify which instances of the T-Schema (and hence which sentences) are essential to generating the contradiction, and hence are paradoxical.

Definition 28. A sentence, A (with name a) in $\mathcal{SLT}_1$ is *Paradoxical* iff, there is some set Λ of instances of the ordinary T-Schema, such that $Tr(a) \equiv A \notin \Lambda$; and $\mathcal{SLT}_1^*, Tr(a) \equiv A, \Lambda \vdash \bot$ but $\mathcal{SLT}_1^*, \Lambda \nvdash \bot$.

That is, a sentence is Paradoxical iff it is essential to the derivation of a contradiction that arises from the T-Schema. An immediate consequence of this is that L is Paradoxical, since $\mathcal{SLT}_1^*, Tr(l) \equiv L \vdash \bot$, but $\mathcal{SLT}_1^* \nvdash \bot$ by Lemma 26 (assuming Sentential Logic is consistent). The point of introducing this concept is that it picks out the sentences, or more precisely the sentences whose instances of the T-schema, that are crucial to the derivation of a contradiction in some case. Removing the instances of the T-schema for any particular paradoxical sentence need not guarantee the consistency of a system, as there may be other paradoxical sentences in the system. In fact, there are systems where every sentence is paradoxical. However, the point is that if the instances of the T-schema for all of the paradoxical sentences are removed, then the system will be consistent by definition. This is proven for the case at hand in the following Lemma:

Lemma 29. *For* Π={all paradoxical sentences in $\mathcal{SLT}_1^*$} *and* $\Sigma = \{Tr(a) \equiv A : \mathit{all}\, a\, \mathit{in}\, \mathcal{SLT}_1^*\}$, the following holds: $\mathcal{SLT}_1^*, \Sigma \setminus \{Tr(a) \equiv A : A \in \Pi\} \nvdash \bot$.

Proof. We will prove this using a Reductio argument. We will therefore begin by assuming the conclusion is false, namely that $\mathcal{SLT}_1^*, \Sigma \setminus \{Tr(a) \equiv A : A \in \Pi\} \vdash \bot$. That is we are assuming that $\mathcal{SLT}_1^*$ plus all instances of the T-schema except those of paradoxical sentences is not consistent. Given that $\mathcal{SLT}_1^*$ is consistent (and that $\mathcal{SLT}_1$ is a finitary system), by Compactness there must be some finite subset $\Psi \subseteq \Sigma \setminus \{Tr(a) \equiv A : A \in \Pi\}$ that allows the derivation of a contradiction (possibly the whole set). Now take Ψ and remove each element in turn. At every stage, either a contradiction will be provable from the remaining elements of Ψ or it will be not. At some stage, however, since the empty set is consistent, the removal of some element of Ψ will produce consistency. That element of Ψ is therefore a Paradoxical sentence by Defn 10. However, that means this element is in Π and therefore cannot be a member of $\Psi \subseteq \Sigma \setminus \{Tr(a) \equiv A : A \in \Pi\}$. Thus we have a contradiction, and it follows that $\mathcal{SLT}_1^*, \Sigma \setminus \{Tr(a) \equiv A : A \in \Pi\} \nvdash \bot$. □

It remains to be now shown that the criterion of Evaluability, as defined above, suffices to remove all of the Paradoxical instances of the ordinary T-Schema.

Lemma 30. *For any sentence A in $\mathcal{SLT}^*_1$, if A is paradoxical, then $a \in Ref_n(a)$, for some n.*

Proof. A being paradoxical means that there is some set Λ, such that $\mathcal{SLT}^*_1, Tr(a) \equiv A, \Lambda \vdash \bot$ but $\mathcal{SLT}^*_1, \Lambda \nvdash \bot$. The proof will proceed by cases. Firstly, we will consider the case where $\Lambda = \emptyset$; secondly, where $|\Lambda| = 1$, then when $|\Lambda| \geq 2$. Furthermore, we will assume the minimality condition that for a given$A$, if $|\Lambda| = n$, then there is no $m < n$ such that there is a Λ' with the same property such that $|\Lambda'| = m$.

In the first case, if $\Lambda = \emptyset$ then $Tr(a) \equiv A$ is itself a contradiction. (For example, A might be the sentence $\neg Tr(a)$). Now, from the perspective of Sentential Logic, the string $Tr(a)$ is treated exactly as though it is an atomic sentence. Therefore, since there are no rules governing $Tr(a)$ in $\mathcal{SLT}^*_1$, any sentence of the form $Tr(a)$ behaves like an atomic sentence in $\mathcal{SLT}^*_1$. Given that $Tr(a)$ is effectively an atomic sentence, the most minimal condition on $Tr(a) \equiv A$ being a contradiction is that $Tr(a)$ is repeated somewhere as part of A. Otherwise A could not contain any conditions which contradict the Left Hand Side of the biconditional. However, if $Tr(a)$ is part of A, then by the definition of Ref_n, $a \in Ref_1(a)$, as required.

The rest of the cases involves the situation where $Tr(a) \equiv A$ is essential to the derivation of a contradiction, however it itself is not contradictory (since $\Lambda \neq \emptyset$). We note first that since there are no derivation rules or axioms governing the Truth Predicate within $\mathcal{SLT}^*_1$, $Tr(a)$ has the status of an atomic proposition. That means that the only provable sentences which contain $Tr(a)$ in $\mathcal{SLT}^*_1$ are tautologies.

In order to investigate the following cases, we assume therefore that we are given a set Λ_1 of instances of the T-Schema for the sentences $B_1, B_2, ..., B_n$, such that $\mathcal{SLT}^*_1, Tr(a) \equiv A, \Lambda_1 \vdash \bot$, but $\mathcal{SLT}^*_1, \Lambda_1 \nvdash \bot$. This means that we have a list of biconditionals from which we can derive a contradiction as follows:

$$\begin{array}{l} Tr(a) \equiv A \\ Tr(b_1) \equiv B_1 \\ Tr(b_2) \equiv B_2 \\ \vdots \\ Tr(b_n) \equiv B_n \end{array}$$

This list of biconditionals has some useful properties. Firstly, every term on the Left Hand Side of a biconditional only appears once on the LHS. Secondly, since Λ_1 is not itself inconsistent, each of the biconditionals is individually consistent with $\mathcal{SLT}^*_1$. Thirdly, independently of these biconditionals, the LHS of the biconditionals cannot be proven or disproven within $\mathcal{SLT}^*_1$. This is because there are no rules governing the Tr predicate in $\mathcal{SLT}^*_1$ and hence the only theorems which contain them must be tautologies in $\mathcal{SLT}^*_1$. A sentence of the form $Tr(p)$ cannot be a tautology in $\mathcal{SLT}^*_1$.

We will now consider the first case, where $|\Lambda_1| = 1$. That is, $\Lambda_1 = \{Tr(b_1) \equiv B_1\}$ for some B_1. Thus, by definition,$\mathcal{SLT}^*_1, Tr(a) \equiv A, Tr(b_1) \equiv B_1 \vdash \bot$, but $\mathcal{SLT}^*_1, Tr(b_1) \equiv B_1 \nvdash \bot$.[6] Importantly, it follows from our minimality condition that B_1 is also paradoxical. The minimality condition means that a contradiction is not derivable from $Tr(a) \equiv A$ independently of $Tr(b_1) \equiv B_1$. It follows that B_1 is critical to the derivation of a contradiction in the same wat that A is, and hence is paradoxical.

As noted, the only theorems within $\mathcal{SLT}^*_1$ that contain $Tr(a)$ or $Tr(b_1)$ are tautologies. Also, neither $Tr(a) \equiv A$ nor $Tr(b_1) \equiv B_1$ are tautologies since it is impossible for the T-schema for paradoxical sentence, that is one that is crucial to the derivation of a contradiction, to be a tautology.It must be remembered that $\mathcal{SLT}^*_1$ does not contain any rules governing the Tr predicate, so both $Tr(a)$ and $Tr(b_1)$ behave like atomic propositions with respect to the rules of $\mathcal{SLT}^*_1$.

Now, we have assumed that $\mathcal{SLT}^*_1, Tr(b_1) \equiv B_1 \nvdash \bot$ but $\mathcal{SLT}^*_1, Tr(a) \equiv A, Tr(b_1) \equiv B_1 \vdash \bot$. This means that from the assumption of $Tr(b_1) \equiv B_1$ within $\mathcal{SLT}^*_1$, it must be possible to derive something that is contradictory to $Tr(a) \equiv A$. This something must include $Tr(a)$ as a part, and cannot be a theorem of $\mathcal{SLT}^*_1$. However, as $Tr(a)$ acts like a basic proposition within $\mathcal{SLT}^*_1$, this is only possible if $Tr(a)$ is included as a part of $Tr(b_1) \equiv B_1$. That is, $Tr(a)$ must be a part of B_1, from which it follows that $a \in Ref_1(b_1)$.

This can be seen in a different way if we consider the possible truth values of the sentences, which given we are working with classical sentential logic is valid. The fact that we have assumed that $\mathcal{SLT}^*_1, Tr(a) \equiv A, Tr(b_1) \equiv B_1 \vdash \bot$ means that there is no possible consistent assignment of truth values such that both $Tr(a) \equiv A$ and $Tr(b_1) \equiv B_1$ come out true. We know that there is a consistent assignment of truth values such that $Tr(b_1) \equiv B_1$ is true (since $\mathcal{SLT}^*_1, Tr(b_1) \equiv B_1 \nvdash \bot$). This means that for every assignment of truth values such that $Tr(b_1) \equiv B_1$ is true, $Tr(a) \equiv A$ must come out as false. Given that $\mathcal{SLT}^*_1$ does not contrain the possibles truth value for $Tr(a)$ (as the only theorems involving $Tr(a)$ are tautologies), this means that the assumption of $Tr(b_1) \equiv B_1$ must constrain the possible truth values of $Tr(a)$ - at least in relation to the truth values of A. Otherwise $Tr(a) \equiv A$ would not come out as false. However, $Tr(b_1) \equiv B_1$ can only constrain the possible truth value of $Tr(a)$ if $Tr(a)$ is a part of B_1, as required.

As noted, B_1 is also paradoxical in the same way that A is. This means that we can use exactly the same argument for $Tr(b_1)$ as just applied to $Tr(a)$. In particular, this means that $Tr(b_1)$ must be a part of A. But that means that $b_1 \in Ref_1(a)$.

By the definition of the Ref_n predicates,it follows that $a \in Ref_2(a)$, as required.

The proof for the cases where $|\Lambda_1| = n > 1$ is based on the same basic observation, but the

[6]

extra possibilities in reference structure mean that the argument is a little more involved.

The first thing to note is that, as in the case where $|\Lambda_1| = 1$, the minimality condition means that every sentence whose instance of the T-schema appears within Λ_1 is paradoxical. If some B_i from $B_1, ..., B_n$ is not paradoxical, then it is not crucial for a paradox and the list $A, B_1, ..., B_n$ minus B_i will give rise to paradox, which contradicts the minimality condition. This means that anything we conclude about A will hold equally for every B_i.

The minimality condition also has another important consequence in this case. We have assumed that a series of equivalences of the form $Tr(b_i) \equiv B_i$ are inconsistent with another equivalence of the form $Tr(a) \equiv A$. Moreover, the minimality condition means that there is no subset of the B_i equivalences with the same property. This means that each equivalence must provide information that is relevant to the derivation of the contradiction. More precisely, each equivalence must provide constraints on the possible truth values assigned to the basic propositions involved. Given the form of the equivalences, the relevant basic propositions are $Tr(a)$ and the $Tr(b_i)$. It is only possible that every equivalence is necessary for the contradiction if each equivalence contrains the possible truth values for at least one of these basic propositions. It follows that every B_i (and A) must contain $Tr(b_j)$ (or $Tr(a)$) for some j.

In the case above, the fact that A is paradoxical means that $Tr(a)$ must be contained in B. The same argument applies in the more general case here, except that we can only conclude that $Tr(a)$ is included in at least one B_i. To put this in terms of references, it means that there must be some B_i that refers to A. We will assume without loss of generality that $Tr(a)$ is part of B_1, and may be part of others.

Our basic strategy will be to follow this pattern of reference and show that it must be the case that $a \in Ref_j(a)$ for some j. There are numerous cases to deal with, and we will structure it as a reductio argument. That is, we will assume that $a \notin Ref_j(a)$.

The same argument means that $Tr(b_1)$ must be included in some B_i or A. It is not possible that it be in A, since then we get $a \in Ref_2(a)$, which contradicts our assumption. Thus $Tr(b_1)$ must be part of some B_i.

If $Tr(b_1)$ is part of B_1, and is part of no other B_i, we get the situation where the only constraint on the truth values of $Tr(b_1)$ is in the equivalence $Tr(b_1) \equiv B_1$. As B_1 is paradoxical, it follow that this equivalence must be contradictory. This however contradicts the minimality conditions. Thus $Tr(b_1)$ must be part of some B_i where $i \neq 1$. We will therefore assume without loss of generality that $Tr(b_1)$ is part of B_2.

Our basic argument applies to B_2 and therefore $Tr(b_2)$ must be included in some B_i. The same arguments as for B_1 mean that it cannot be part of A or only part of B_2. We will consider two cases, firstly that it is only part of B_1, and secondly that it is part of some other B_i, say B_3.

We will consider the case where $Tr(b_2)$ is only part of B_1, and $Tr(b_1)$ is only part of B_2. In this case the equivalences containing B_1 and B_2 are the only equivalences that contrain the truth values of $Tr(b_1)$ and $Tr(b_2)$. Given each of these is paradoxical, it follows that these two equivalences are the only ones necessary for the contradiction, which contradicts the minimality condition. If $Tr(b_2)$ is only part of B_1, and $Tr(b_1)$ is part of B_2 and some other B_i, we will reorder our series so that this other B_i becomes B_2 and we go through this step again.

We will now consider the case where $Tr(b_2)$ is part of some B_3. The familiar argument means that $Tr(b_3)$ must be part of some B_i. If this sets up some self-contained reference loop, the argument considered for B_1 and B_2 above shows that this contradicts the minimality condition. So either $Tr(b_3)$ is part of some B_4, or we must be able to reorder the series so that there is a B_3 that is a part of some B_4.

The argument will repeat for each following B_i, so that $Tr(b_i)$ must be a part of some B_{i+1}. This, however, will be impossible to continue once $i = n$, and hence our reductio assumption cannot hold. Hence $a \in Ref_j(a)$ for some j.

We have now dealt with all possible cases for $|\Lambda_1|$, so it follows that, for any paradoxical sentence a, $a \in Ref_n(a)$ for some n, as required. □

From this, the obvious result is as follows:

Lemma 31. *If a sentence A is Paradoxical, then it is not evaluable.*

Proof. If a sentence A is Paradoxical, by Lemma 30, it follows that $a \in Ref_n(a)$ for some n. However, if $a \in Ref_n(a)$ for some n, by the Eval rules, it is provable that $\neg Eval(a)$. That is, A is not evaluable. □

Theorem 32. *$\mathcal{SLT}_1$ is consistent, if $\mathcal{SL}$ is consistent.*

Proof. Firstly, if $\mathcal{SL}$ is consistent, then $\mathcal{SLT}_1^*$ is consistent (Lemma 26). Now $\mathcal{SLT}_1$ is identical to $\mathcal{SLT}_1^*$ with the Strengthened T-Schema as an axiom, that is $\mathcal{SLT}_1 \vdash P$ if, and only if, $\mathcal{SLT}_1^*, \{Tr(a) \equiv (A \wedge Eval(a)) : all\, a\} \vdash P$. This means that $\mathcal{SLT}_1$ is consistent exactly when $\mathcal{SLT}_1^* \cup \{Tr(a) \equiv (A \wedge Eval(a)) : all\, a\}$ is consistent.

Now the *Eval* rules are in $\mathcal{SLT}_1^*$, so for all sentences, either $\mathcal{SLT}_1^* \vdash Eval(a)$ or $\mathcal{SLT}_1^* \vdash \neg Eval(a)$.

If $\mathcal{SLT}_1^* \vdash Eval(a)$, then $\mathcal{SLT}_1^*, Tr(a) \equiv (A \wedge Eval(a)) \vdash Tr(a) \equiv A$ (Lemma 8) and $\mathcal{SLT}_1^*, Tr(a) \equiv A \vdash Tr(a) \equiv (A \wedge Eval(a))$.[7] This means that $\mathcal{SLT}_1^*, \{Tr(a) \equiv (A \wedge Eval(a)) : all\, a\} \vdash P$ if, and only if, $\mathcal{SLT}_1^*, \{Tr(a) \equiv A : all\, A\, s.t. \vdash Eval(a)\}, \{Tr(a) \equiv (A \wedge Eval(a)) : all\, a\, s.t. \vdash \neg Eval(a)\} \vdash P$.

If $\mathcal{SLT}_1^* \vdash \neg Eval(a)$, then it follows that $\mathcal{SLT}_1^*, Tr(a) \equiv (A \wedge Eval(a)) \vdash \neg Tr(a)$ and $\mathcal{SLT}_1^*, \neg Tr(a) \vdash Tr(a) \equiv (A \wedge Eval(a))$. The first of these is Lemma 9. The second can be shown as follows:

[7] The proof of this is trivial.

1	$\neg Eval(a)$	Hypothesis
2	$\neg Tr(a)$	Hypothesis
3	$\mid\ Tr(a)$	Assume
4	$\mid\ Tr(a) \vee (A \wedge Eval(a))$	$\vee$ Introduction, ln 3
5	$\mid\ A \wedge Eval(a)$	Disj. Syll, ln 2,4
6	$Tr(a) \supset (A \wedge Eval(a))$	$\supset$ Introduction, ln 3-5
7	$\mid\ A \wedge Eval(a)$	Assume
8	$\mid\ Eval(a)$	$\wedge$ Elim ln 7
9	$\mid\ Eval(a) \vee Tr(a)$	$\vee$ Intro ln 8
10	$\mid\ Tr(a)$	Disj. Syll ln1,9
11	$(A \wedge Eval(a)) \supset Tr(a)$	$\supset$ Intro, ln7-10
12	$Tr(a) \equiv (A \wedge Eval(a)]$	$\equiv$ Itro, ln 6,11

This means that for all a that are not evaluable in $\mathcal{SLT}^*_1$, $\neg Eval(a)$, $\neg Tr(a)$ and the Strengthened T-Schema are equivalent in $\mathcal{SLT}^*_1$.

It follows that assuming the Strengthened T-Schema is equivalent to assuming the ordinary T-Schema for evaluable sentences and assuming that $\neg Tr(a)$ for unevaluable sentences. That is, $\mathcal{SLT}^*_1, \{Tr(a) \equiv (A \wedge Eval(a)) \ : \ all\, a\} \vdash P$ if, and only if, $\mathcal{SLT}^*_1, \{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\}, \{\neg Tr(a) \ : \ all\, a\, s.t. \vdash \neg Eval(a)\} \vdash P$.

However, $\mathcal{SLT}^*_1 \cup \{Tr(a) \equiv (A \wedge Eval(a)) \ : \ all\, a\}$ is equivalent to $\mathcal{SLT}_1$. This means that everything that is provable in $\mathcal{SLT}_1$ is provable from $\mathcal{SLT}^*_1 \cup \{Tr(a) \equiv (A \wedge Eval(a)) \ : \ all\, a\}$. However, as just shown, everything provable in this is provable in $\mathcal{SLT}^*_1 \cup \{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\} \cup \{\neg Tr(a) \ : \ all\, a\, s.t. \vdash \neg Eval(a)\}$. It follows that $\mathcal{SLT}_1$ is consistent, if, and only if, $\mathcal{SLT}^*_1 \cup \{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\} \cup \{\neg Tr(a) \ : \ all\, a\, s.t. \vdash \neg Eval(a)\}$ is consistent.

Now, by Lemma 29, $\mathcal{SLT}^*_1 \cup \{Tr(a) \equiv A \ : \ all\, A\, not\, paradoxical\}$ is consistent. Since, by Lemma 31, all Paradoxical Sentences are not evaluable, so $\mathcal{SLT}^*_1 \cup \{Tr(a) \equiv A \ : \ all\, A\, that\, are\, evaluable\}$ is consistent. It therefore, is only necessary to demonstrate that adding $\neg Tr(a)$ for all a such that $\vdash \neg Eval(a)$ does not introduce an inconsistency.

The addition of the set of sentences of the form $\neg Tr(a)$ can only introduce a consistency if $\mathcal{SLT}^*_1, \{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\} \vdash Tr(b)$ for at least one b s.t. $\vdash \neg Eval(b)$. By Lemma 27, $\mathcal{SLT}^*_1 \nvdash Tr(b)$ for all b. Furthermore, for any relevant b such that $\mathcal{SLT}^*_1, \{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\} \vdash Tr(b)$, $Tr(b) \equiv B$ is not a member of $\{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\}$. This means that the question of consistency reduces to the question of whether it is possible to derive a $Tr(b)$ such that $\mathcal{SLT}^*_1 \vdash \neg Eval(b)$ from the set $\{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\}$ within $\mathcal{SLT}^*_1$.

Firstly, the set $\{Tr(a) \equiv A \ : \ all\, A\, s.t. \vdash Eval(a)\}$ is consistent in $\mathcal{SLT}^*_1$, which means that $Tr(b)$ cannot be proven by means of a contradiction. Secondly, if $Tr(b)$ is provable, it must at the very least appear in some Right Hand Side of an equivalence, since all other instances of $Tr(b)$ in

$\mathcal{SLT}_1^*$ are tautological, i.e. $Tr(b)$ must be part of some A. However, essentially the same situation as in the case of the proof of Lemma 30 arises here:

Let $Tr(b)$ be a part of A_1, and $Tr(a_1) \equiv A_1$ is a member of the set. If $Tr(b)$ is provable, then either A_1 or $\neg A_1$ must be provable. However, given that $Tr(b)$ is not provable independently of $Tr(a_1) \equiv A_1$, it follows that neither A_1 nor $\neg A_1$ are provable independently of $Tr(a_1) \equiv A_1$ - which means that A_1 is only provable or disprovable if $Tr(a_1)$ is provable or disprovable independently of $Tr(a_1) \equiv A_1$. This in turn is only possible if $Tr(a_1)$ is a part of some A_2 and $Tr(a_2) \equiv A_2$ is part of the relevant set. Exactly the same argument applies again, with $Tr(a_2)$ depending on some further A_3 and so on. It follows that $Tr(b)$ will not be provable, unless this chain eventually refers back to some previous element in the chain, that is $Tr(a_n)$ is a part of A_m for some $m < n$. However if that happens, it follows that $Ref_{n-m}(a_m) \hookrightarrow a_m$ and hence A_m is not evaluable. But this contradicts the definition of the set. Therefore $Tr(b)$ is not provable.

Thus $\mathcal{SLT}_1^* \cup \{Tr(a) \equiv A : all\, A\, s.t. \vdash Eval(a)\} \cup \{\neg Tr(a) : all\, a\, s.t. \vdash \neg Eval(a)\}$ is consistent, which means that $\mathcal{SLT}_1$ is consistent. □

Discussion

While this proof does not bring out the structure of $\mathcal{SLT}_1$ and the Strengthened T-Schema as clearly as the soundness proof, it highlights the key reason why the Strengthened T-Schema succeeds in solving the Liar Paradox. The key to this proof is showing that all of the paradoxical sentences are not evaluable. Once this is established, the Strengthened T-Schema is shown to ensure that $\mathcal{SLT}_1$ is consistent. This is, as noted in Chapter 7, the reason that the approach works: the new concept of Evaluability isolates the paradoxical sentence by virtue of its place in the Strengthened T-Schema.

Bibliography

Allais, Lucy. “Kant’s Transcendental Idealism and Contemporary Anti-Realism”. In: *International Journal of Philosophical Studies* 11.4 (2003), pp. 369–392.

Archangelsky, Dmitri A. and Mikhail A. Taitslin. “A Logic for Information Systems”. In: *Studia Logica* 58 (1997), pp. 3–16.

Armour-Garb, Bradley. “Minimalism, The Generalization Problem and The Liar”. In: *Synthese* 139 (2004), pp. 491–512.

Armour-Garb, Bradley and Graham Priest. “Analetheism: a Pyrrhic Victory”. In: *Analysis* 65.2 (2005), pp. 167–73.

Armour-Garb Bradley & Beall, J. C. “Further Remarks on Truth and Contradiction”. In: *The Philosophical Quarterly* 52.207 (2002), pp. 217–225.

Ashworth, E. J. “Strict and Material Implication in the Early Sixteenth Century”. In: *Notre Dame Journal of Formal Logic* XIII.4 (1972), pp. 556–560.

— “The Treatment of Semantic Paradoxes from 1400 to 1700”. In: *Notre Dame Journal of Formal Logic* XIII.1 (1972), pp. 34–52.

Atlas, Jay David. “On Presupposing”. In: *Mind* 87.347 (1978), pp. 396–411.

Barba, Juan. “Construction of Truth Predicates: Approximation Versus Revision”. In: *The Bulletin of Symbolic Logic* 4.4 (1998), pp. 399–417.

Bartlett, Stephen J. “Varieties of Self-Reference”. In: *Self-Reference : Reflections on Reflexivity.* Ed. by Stephen J. Bartlett and Peter Suber. Dordrecht: Martinus Nijhoff Publishers, 1974.

Bartlett, Stephen J. and Peter Suber, eds. *Self-Reference : Reflections on Reflexivity.* Dordrecht: Martinus Nijhoff Publishers, 1987.

Barwise, Jon and John Etchemendy. *The Liar: An Essay on Truth and Circularity.* New York: Oxford University Press, 1987.

Beall, J. C. “Is the Observable World Consistent?” In: *Australasian Journal of Philosophy* 78.1 (2000), pp. 113–118.

— “Is Yablo’s Paradox Non-Circular?” In: *Analysis* 61.3 (2001), pp. 176–187.

— *Spandrels of Truth.* Oxford University Press, 2009.

Beall, J. C. “The Singularity Theory of Denotation”. In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 253–261.

Beall, J. C. and Octavio Bueno. “The simple liar without bivalence?” In: *Analysis* 62.1 (2002), pp. 22–26.

Beall, J.C. “True, false and paranormal”. In: *Analysis* 66.2 (2006), pp. 102–14.

Belnap, Nuel D. “Gupta’s Rule of Revision Theory of Truth”. In: *Journal of Philosophical Logic* 11 (1982), pp. 103–16.

Berg, Jan. “What is a Proposition?” In: *Logique et Analyse* 10 (1967), pp. 293–306.

Betti, Arianna. “Leśniewski’s early Liar, Tarski and Natural Language”. In: *Annals of Pure and Applied Logic* 127 (2004), pp. 267–287.

Bhave, S. V. “The Liar Paradox and Many-Valued Logic”. In: *The Philosophical Quarterly* 42.169 (1992), pp. 465–79.

Bigelow, John C. “Believing in Semantics”. In: *Linguistics and Philosophy* 2 (1978), pp. 101–144.

Boolos, George. “Constructing Cantorian Counterexamples”. In: *Journal of Philosophical Logic* 26.3 (1997), pp. 237–239.

Boos, William. “Consistency and Konsistanz”. In: *Erkenntnis* 26 (1987), pp. 1–43.

Brendel, Elke. “Bemerkung zu R. Schüblers ‘Super-Lügner’.” In: *Erkenntnis* 24 (1986), pp. 397–398.

— “Partial Worlds and Paradox”. In: *Erkenntnis* 39 (1993), pp. 191–208.

Bromand, Joachim. “Why Paraconsistent Logic Can Only Tell Half the Truth”. In: *Mind* 111 (2002), pp. 741–749.

Burge, Tyler. “Epistemic Paradox”. In: *The Journal of Philosophy* 81.1 (1984), pp. 5–29.

— “Semantical Paradox”. In: *The Journal of Philosophy* 76 (1979), pp. 169–98.

Buridan, John. *John Buridan on Self-Reference.* Ed. by G. E. Hughes. Trans. G. E. Hughes. Cambridge: Cambridge University Press, 1982.

Caton, Charles E. “Strawson On Referring”. In: *Mind* 68 (1959), pp. 539–544.

Chihara, Charles S. “Priest, the Liar, and Gödel”. In: *Journal of Philosophical Logic* 13 (1984), pp. 117–124.

Clark, Michael. “Recalcitrant variants of the liar paradox”. In: *Analysis* 59.2 (1999), pp. 117–126.

Cohen, L. Jonathon. “Can the Logic of Indirect Discourse be Formalised?” In: *The Journal of Symbolic Logic* 22.3 (1957), pp. 225–232.

— “Professor Goodstein’s Formalisation of the Policeman”. In: *The Journal of Symbolic Logic* 23.4 (1958), p. 420.

Cohen, Yael. *Semantic Truth Theories.* Jerusalem: The Magnes Press, 1994.

Cook, Roy T. “Counterintuitive consequences of the Revision Theory of Truth”. In: *Analysis* 62.1 (2002), pp. 16–22.

Culina, Boris. “The Concept of Truth”. In: *Synthese* 126 (2001), pp. 339–360.

Dahl, Östen. “In Defense of a Stawsonian Approach to Presupposition”. In: *Crossing the Boundaries in Linguistics: studies presented to Manfred Bierwisch.* Ed. by Wolfgang Klein and William Levelt. Dordrecht: D. Reidel Pub. Co., 1981, pp. 191–200.

Dever, Josh. “Compositionality as Methodology”. In: *Linguistics and Philosophy* 22 (1999), pp. 311–326.

DeVidi, David and Graham Solomon. “Tarski on “Essentially Richer” Metalanguages”. In: *Journal of Philosophical Logic* 28 (1999), pp. 1–28.

Donnellan, Keith S. “A Note on the Liar Paradox”. In: *The Philosophical Review* 66.3 (1957), pp. 394–7.

Dowden, Bradley H. “Accepting Inconsistencies from the Paradoxes”. In: *Journal of Philosophical Logic* 13 (1984), pp. 125–30.

Eklund, Matti. “Deep Inconsistency”. In: *Australasian Journal of Philosophy* 80.3 (2002), pp. 321–331.

Encarnacion, Jose. “On Ushenko’s Version of the Liar-Paradox”. In: *Mind* 64.253 (1955), pp. 99–100.

Epstein, Richard L. “A Theory of Truth Based on a Medieval Solution to the Liar Paradox”. In: *History and Philosophy of Logic* 13 (1992), pp. 149–177.

Evans, Ellis. “Notes on the Symbolic Process”. In: *Mind* 60.237 (1951), pp. 62–79.

Feferman, Solomon. “Tarski’s Conception of Logic”. In: *Annals of Pure and Applied Logic* 126 (2004), pp. 5–13.

Fenstad, Jens Erik. “Tarski, truth and natural languages”. In: *Annals of Pure and Applied Logic* 126 (2004), pp. 15–26.

Field, Hartry. *Saving Truth from Paradox.* Oxford University Press, 2008.

— “Semantic Paradoxes and Vagueness Paradoxes”. In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 262–310.

Floridi, Luciano. “Open Problems in the Philosophy of Information”. In: *Metaphilosophy* 35.4 (2004), pp. 554–582.

Fodor, Jerry A. “Having Concepts: A Brief Refutation of the Twentieth Century”. In: *Mind and Language* 19.1 (2004), pp. 29–47.

Fodor, Jerry A. and Ernest Lepore. *The Compositionality Papers.* Oxford: Clarendon Press, 2002.

— “What Can’t Be Valued Can’t Be Valued, and it Can’t be Supervalued Either”. In: *Journal of Philosophy* 93 (1996), pp. 516–536.

— “Why Compositionality Won’t Go Away: Reflections on Horwich’s ‘Deflationary’ Theory”. In: *Ratio* 14 (2001), pp. 350–368.

Fodor, Jerry A. and Ernest Lepore. “Why Meaning (Probably) Isn’t Conceptual Role”. In: *Mind and Language* 6.4 (1991), pp. 329–343.

Garner, Richard. “On Saying what is True”. In: *Noûs* 6.3 (1972), pp. 201–224.

Giaretta, Pierdaniele. “Liar, Reducibility and Language”. In: *Synthese* 117 (1999), pp. 355–374.

Glanzberg, Michael. “A Contextual-Hierarchical Approach to Truth and the Liar Paradox”. In: *Journal of Philosophical Logic* 33 (2004), pp. 27–88.

— “Against Truth-Value Gaps”. In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 151–194.

— “Minimalism and Paradoxes”. In: *Synthese* 135 (2003), pp. 13–36.

— “The Liar in Context”. In: *Philosophical Studies* 103 (2001), pp. 217–251.

Goldstein, Laurence. “A Unified Solution to Some Paradoxes”. In: *Proceedings of the Aristotelian Society* 100 (2000), pp. 53–74.

— “A Yabloesque Paradox in Set Theory”. In: *Analysis* 54.4 (1994), pp. 223–227.

— “’This Statement is Not True’ is Not True”. In: *Analysis* 52.1 (1992), pp. 1–5.

— “Truth-bearers and the Liar - a reply to Alan Weir”. In: *Analysis* 61.2 (2001), pp. 115–126.

Goldstein, Laurence and Leonard Goddard. “Stengthened Paradoxes”. In: *Australasian Journal of Philosophy* 58.3 (1980), pp. 211–221.

Gómez-Torrente, Mario. “The indefinability of truth in the “Wahrheitsbegriff””. In: *Annals of Pure and Applied Logic* 126 (2004), pp. 27–37.

Goodstein, R. L. *Essays in the Philosophy of Mathematics.* Leicester: Leicester University Press, 1965.

— “Logical Paradoxes”. In: *Essays in the Philosophy of Mathematics.* Leicester: Leicester University Press, 1965.

— “On the Formalisation of Indirect Discourse”. In: *The Journal of Symbolic Logic* 23.4 (1958), pp. 417–9.

— “Proof by Reductio ad Absurdum”. In: *Essays in the Philosophy of Mathematics.* Leicester: Leicester University Press, 1965, pp. 1–11.

Grattan-Guiness, I. “Structural Similarity or Structuralism? Comments on Priest’s Analysis of Paradoxes of Self-Reference”. In: *Mind* 107.428 (1998), pp. 823–834.

Graves, David C. “On Presenting Works of Art: An Analysis of Meaning in the Second Intention”. In: *Philosophia* 29.1-4 (2002), pp. 173–190.

Greenberg, Robert. “A Note on Strawson’s Theories of Presuppositions”. In: *Mind* 80 (1971), pp. 258–261.

Greenough, Patrick. “Free Assumptions and the Liar Paradox”. In: *Amercian Philosophical Quarterly* 38 (2001), pp. 115–134.

Groenendijk, Jeroen and Martin Stokhof. “Why Compositionality?” In: *Reference and Quantification: The Partee Effect*. Ed. by J. Carlson G. & Pelletier. Stanford: CSLI Press, 2005, pp. 83–106.

Grover, Dorothy. *A Prosentential Theory of Truth*. Princeton: Princeton University Press, 1992.

Gupta, Anil. “Partially Defined Predicates and Semantic Pathology”. In: *Philosophy and Phenomenological Research* 65.2 (2002), pp. 402–409.

— “Truth and Paradox”. In: *Journal of Philosophical Logic* 11 (1982), pp. 1–60.

Gupta, Anil and Nuel Belnap. *The Revision Theory of Truth*. The MIT Press, 1993.

Gupta, Anil and Robert L. Martin. “A Fixed Point Theorem for the Weak Kleene Valuation Scheme”. In: *Journal of Philosophical Logic* 13 (1984), pp. 131–35.

Hajek, Peter, Jeff Paris, and John Shepherdson. “The Liar Paradox and Fuzzy Logic”. In: *The Journal of Symbolic Logic* 65.1 (2000), pp. 339–346.

Halbach, Volker. “Conservative Theories of Truth”. In: *Studia Logica* 62 (1999), pp. 353–370.

— “How Innocent is Deflationism?” In: *Synthese* 126 (2001), pp. 167–194.

Halldén, Sören. *Truth Strategy Simplified*. Vol. No. 24. Library of Theoria. Stockholm: Thales, 1999.

Hendriks, Petra and Helen de Hoop. “Optimality Theoretic Semantics”. In: *Linguistics and Philosophy* 24.1 (2001). TY - JOUR, pp. 1–32.

Herzberger, Hans G. “Naive Semantics and the Liar Paradox”. In: *Journal of Philosophy* 79.9 (1982), pp. 479–497.

— “Notes on Naive Semantics”. In: *Journal of Philosophical Logic* 11 (1982), pp. 61–102.

— “Paradoxes of Grounding in Semantics”. In: *The Journal of Philosophy* 67.6 (1970), pp. 145–167.

Higginbotham, James. “Conditionals and Compositionality”. In: *Philosophical Perspectives*. Ed. by John Hawthorne and Dean Zimmerman. Vol. 17. Malden, MA: Blackwell, 2003, pp. 181–194.

Hinckfuss, Ian. “B, A Buridan-Style Propositional Logic”. In: *Logique et Analyse* 135-136 (1991), pp. 333–344.

— “Pro Buridano; Contra Hazenum”. In: *Canadian Journal of Philosophy* 21.3 (1991), pp. 389–398.

Hintikka, Jaakko. “Introduction and Postscript”. In: *Synthese* 126 (2001), pp. 1–16.

— “Post-Tarskian Truth”. In: *Synthese* 126.1-2 (2001), pp. 17–36.

Hintikka, Jaako. “Independence-friendly logic and axiomatic set theory”. In: *Annals of Pure and Applied Logic* 126 (2004), pp. 313–333.

— *The Principles of Mathematics Revisited*. Cambridge: Cambridge UP, 1996.

Hiż, Henryk. “Reexamination of Tarski’s Semantics”. In: *Annals of Pure and Applied Semantics* 126 (2004), pp. 39–48.

Hochberg, Herbert. “On Referring and Asserting”. In: *Philosophical Studies* 20.6 (1969), pp. 81–88.

Hochberg, Herbert. “Strawson, Russell and the King of France”. In: *Philosophy of Science* 37 (1970), pp. 363–384.

Hodges, Wilfrid. “An Editor Recalls Some Hopeless Papers”. In: *The Bulletin of Symbolic Logic* 4.1 (1998), pp. 1–16.

— “Compositional Semantics for a Language of Imperfect Information”. In: *Logic Journal of the IGPL* 5.4 (1997), pp. 539–563.

— “Formal Features of Compositionality”. In: *Journal of Logic, Language and Information* 10 (2001), pp. 7–28.

— “What languages have Tarski truth definitions?” In: *Annals of Pure and Applied Logic* 126 (2004), pp. 93–113.

Holbach, Volker. “How not to state T-sentences”. In: *Analysis* 66.4 (2006), pp. 276–80.

Hornstein, Norbert. “Putting Truth into Universal Grammar”. In: *Linguistics and Philosophy* 18.4 (1995), pp. 381–400.

Horwich, Paul. “Deflating Compositionality”. In: *Ratio* 14 (2001), pp. 369–385.

Houben, Jan E. M. “Bartrhari’s Solution to the Liar and Some Other Paradoxes”. In: *Journal of Indian Philosophy* 23 (1995), pp. 381–401.

Hudson, James L. “Schlesinger on the Newcomb Problem”. In: *Australasian Journal of Philosophy* 57.2 (1979), pp. 145–156.

Hugly, Philip and Charles Sayward. “The Lessons of the Liar”. In: *Theory and Decision* 11 (1979), pp. 55–70.

Humphries, Jill. “Gödel’s Proof and the Liar Paradox”. In: *Notre Dame Journal of Formal Logic* 20.3 (1979), pp. 535–544.

Hyttinen, Tapani and Gabriel Sandu. “Truth and definite truth”. In: *Annals of Pure and Applied Logic* 126 (2004), pp. 49–55.

Iacona, Andrea. “The Expressing Relation”. In: *Dialectica* 56.3 (2002), pp. 235–260.

Irvine, A. D. “Gaps, Gluts and Paradox”. In: *Canadian Journal of Philosophy Supplementary Volume* 18 (1992), pp. 273–299.

Janssen, Theo M. V. “Frege, Contextuality and Compositionality”. In: *Journal of Logic, Language and Information* 10 (2001), pp. 115–136.

Jenkins, C.S. “True, false, paranormal and designated: a reply to Beall”. In: *Analysis* 67.1 (2007), pp. 80–83.

Juhl, C. F. “A context-sensitive liar”. In: *Analysis* 57.3 (1997), pp. 202–204.

Kane, R. H. “Presupposition and Entailment”. In: *Mind* 81 (1972), pp. 401–404.

Kazmi, Ali and Francis Jeffry Pelletier. “Is Compositionality Formally Vacuous?” In: *Linguistics and Philosophy* 23 (1998), pp. 629–633.

Keene, G. B. “Self-referent Inference and the Liar Paradox”. In: *Mind* 92 (1983), pp. 430–433.

Kemp, Gary. "Salmon on Fregean Approaches to the Paradox of Analysis". In: *Philosophical Studies* 78 (1995), pp. 153–162.

Kempson, Ruth M. *Presupposition and the Delimitation of Semantics.* Cambridge: Cambridge University Press, 1975.

Kirkham, R. L. *Theories of Truth.* Cambridge, Mass.: MIT Press, 1992.

Klagge, James C. "Convention T Regained". In: *Philosophical Studies* 32 (1977), pp. 377–381.

Kossak, Roman. "Undefinability of truth and non-standard models". In: *Annals of Pure and Applied Logic* 126 (2004), pp. 115–123.

Kripke, Saul. "Outline of a Theory of Truth". In: *Journal of Philosophy* 72.6 (1975), pp. 690–716.

Kroon, Frederick. "Beyond Belief? A critical study of Graham Priest's *Beyond the Limits of Thought*". In: *Theoria* 67.2 (2001), pp. 140–153.

Laraudogoitia, Jon Pérez. "Priest on the paradox of the gods". In: *Analysis* 60.2 (2000), pp. 152–55.

Lee, Byeong D. "The Paradox of Belief Instability and a Revision Theory of Belief". In: *Pacific Philosophical Quarterly* 79 (1998), pp. 314–328.

Leitgeb, Hannes. "Truth as Translation - Part A". In: *Journal of Philosophical Logic* 30 (2001), pp. 281–307.

— "Truth as Translation - Part B". In: *Journal of Philosophical Logic* 30 (2001), pp. 309–328.

— "What Truth Depends On". In: *Journal of Philosophical Logic* 34 (2005), pp. 155–92.

Löbner, Sebastian. "Polarity in Natural Language: Predication, Quantification and Negation in Particular and Characterizing Contexts". In: *Linguistics and Philosophy* 23 (2000), pp. 213–308.

Mar G. & St Denis, P. "What the Liar Taught Achilles". In: *Journal of Philosophical Logic* 28 (1999), pp. 29–46.

Martin, Robert L. "Relative Truth and Semantic Categories". In: *Journal of Philosophical Logic* 3 (1974), pp. 149–153.

— "Toward a Solution to the Liar Paradox". In: *The Philosophical Review* 76.3 (1967), pp. 279–311.

Martinich, A. P. "A Pragmatic Solution to the Liar Paradox". In: *Philosophical Studies* 43 (1983), pp. 63–67.

— "Conversational Maxims and Some Philosophical Problems". In: *The Philosophical Quarterly* 30.120 (1980), pp. 215–228.

Maudlin, Tim. *Truth and Paradox.* Oxford: Clarendon Press, 2004.

McDonough, Richard M. *The Argument of the"Tractatus": Its Relevance to Contemporary Theories of Logic, Language, Mind and Philosophical Truth.* Albany: SUNY Press, 1996.

McGee, Vann. "Maximal Consistent Sets of Instances of Tarski's Schema (T)". In: *Journal of Philosophical Logic* 21 (1992), pp. 235–241.

McGee, Vann. *Truth, Vagueness and Paradox: an essay on the logic of truth.* Indianapolis: Hackett Pub. Co, 1990.

McGrath, Matthew. "Scott Soames: Understanding Truth". In: *Philosophy and Phenomenological Research* 65.2 (2002), pp. 410–417.

Michael, Fred Seymour. "Entailment and Bivalence". In: *Journal of Philosophical Logic* 31 (2002), pp. 289–300.

Mills, E. "A Simple Solution to the Liar". In: *Philosophical Studies* 89 (1998), pp. 197–212.

Moffett, Marc A. "A Note on the Relationship Between Mates' Puzzle and Frege's Puzzle". In: *Journal of Semantics* 19 (2002), pp. 159–166.

Mortensen, Chris. *Inconsistent Mathematics.* Kluwer Academic Publishers, 1995.

Mou, Bo. "The Enumerative Character of Tarski's Definition of Truth and its General Character in a Tarskian System". In: *Synthese* 126 (2001), pp. 91–121.

Nerlich, G. "Presupposition and Entailment". In: *American Philosophical Quarterly* 2.1 (1965), pp. 33–42.

Niiniluoto, Ilkka. *Truthlikeness.* Dordrecht: D. Reidel Pub. Co., 1987.

Pagin P. & Westerståhl, D. "Compositionality: Current Issues". In: *Journal of Logic, Language and Information* 10 (2001), pp. 1–5.

Parsons, Charles. "Informal Axiomatisation, Formalisation and the Concept of Truth". In: *Synthese* 27 (1974), pp. 27–47.

— "The Liar Paradox". In: *Journal of Philosophical Logic* 3 (1974), pp. 381–412.

Parsons, Terence. "Assertion, Denial and the Liar Paradox". In: *Journal of Philosophical Logic* 13 (1984), pp. 137–52.

Pelletier, Frances Jeffrey. "On an Argument Against Semantic Compositionality". In: *Logic and Philosophy of Science in Uppsala.* Ed. by D. Prawitz and D. Westerstål. Dordrecht: Kluwer Academic Publishers, 1994, pp. 599–610.

Pelletier, Francis Jeffry. "Did Frege Delieve Frege's Principle?" In: *Journal of Logic, Language and Information* 10 (2001), pp. 87–114.

Peregrin, Jaroslav. "Is Compositionality an Empirical Matter?" In: *The Compositionality of Concepts and Meanings.* Ed. by E. Machery & G. Schurz M. Werning. Frankfurt: Ontos, 2005, pp. 135–150.

Post, John F. "Presupposition, Bivalence, and the Possible Liar". In: *Philosophia* 8 (1979), pp. 645–650.

— "Referential Presupposition". In: *Australasian Journal of Philosophy* 50.2 (1972), pp. 160–167.

— "Shades of the Liar". In: *Journal of Philosophical Logic* 2 (1973), pp. 370–386.

Powell, George. "Underdetermination and the Principles of Semantic Theory". In: *Proceedings of the Aristotelian Society* 102 (2002), pp. 321–328.

Priest, Graham. *Beyond the Limits of Thought.* Cambridge: Cambridge University Press, 1995.

— "Can Contradictions be True? II". In: *The Aristotelian Society: Supplementary Volume* 67 (1993), pp. 35–54.

— *Doubt Truth to be a Liar.* Oxford: Clarendon Press, 2006.

— *In Contradiction: A Study of the Transconsistent.* Dordrecht: Martinus Nijhoff Publishers, 1987.

— "Logic of Paradox Revisited". In: *Journal of Philosophical Logic* 13 (1984), pp. 153–79.

— "On the Principle of Uniform Solution: A Reply to Smith". In: *Mind* 109.433 (2000), pp. 123–126.

— "Rational Dilemmas". In: *Analysis* 62.1 (2002), pp. 11–16.

— "The Import of Closure: Some Comments on Grattan-Guiness". In: *Mind* 127.428 (1998), pp. 835–840.

— "The Logic of Paradox". In: *The Journal of Philosophical Logic* 8 (1979), pp. 219–241.

— "Truth and Contradiction". In: *The Philosophical Quarterly* 50.200 (2000), pp. 305–319.

— "What is So Bad about Contradictions?" In: *The Journal of Philosophy* 95.8 (1998), pp. 410–426.

— "Yablo's Paradox". In: *Analysis* 57.4 (1997), pp. 236–242.

Priest, Graham and Richard Routley. "The Philosophical Significance and Inevitability of Paraconsistency". In: *Paraconsistent Logic: Essays on the Inconsistent.* Ed. by Richard; Priest Graham; Routley and Jean Norman. Munich: Philosophia, 1989.

Puntel, Lorenz B. "Truth, Sentential Non-Compositionality, and Ontology". In: *Synthese* 126 (2001), pp. 221–259.

Quine, W. V. O. "The Ways of Paradox". In: *The Ways of Paradox and Other Essays.* Cambridge, Mass.: Harvard University Press, 1966, pp. 1–21.

Rauszer, Cecylia. "Semi-Boolean algebras and their applications to intuitionistic logic with dual operations". In: *Fundamenta Mathematicae* 83 (1974), pp. 219–249.

Ray, Greg. "Thinking in L". In: *Noûs* 29.3 (1995), pp. 378–396.

Read, Stephen. "Freeing Assumptions from the Liar Paradox". In: *Analysis* 63.2 (2003), pp. 162–166.

— "The Liar Paradox from John Buridan back to Thomas Bradwardine". In: *Vivarium* 40.2 (2002), pp. 189–218.

Reimer, Marga. "Do Adjectives Conform to Compositionality?" In: *Language and Mind.* Ed. by James E. Tomberlin. 16th ed. Philosophical Perspectives. Cambridge, Mass.: Blackwell Publishers, 2002.

Reuter, Mark. "Language, Lies, and Human Action in William of Ockham's Treatment of Insolubles". In: *Vivarium* 36.1 (1998), pp. 108–131.

Richards, Thomas J. "Self-Referential Paradoxes". In: *Mind* 76.303 (1967), pp. 387–403.

Rieger, Adam. “An Argument for Finsler-Aczel Set Theory”. In: *Mind* 109.434 (2000), pp. 241–253.

— “The Liar, The Strengthened Liar, and Bivalence”. In: *Erkenntnis* 54 (2001), pp. 195–203.

Robbins, Philip. “What Compositionality Still Can Do”. In: *The Philosophical Quarterly* 51.204 (2001), pp. 328–336.

Rumfitt, Ian. “Semantic Theory and Necessary Truth”. In: *Synthese* 126 (2001), pp. 283–324.

Rüstow, Alexander. *Der Lügner : Theorie/Geschichte und Auflösung.* Ed. by Leonardo Tarán. Vol. 35. Greek & Roman Philosophy. Reprint of 1910, B. G. Teubner, Leipzig. New York: Garland Publishing, 1987.

Sainsbury, R. M. “Two Ways to Smoke a Cigarette”. In: *Ratio* 14 (2001), pp. 386–406.

Sandu, Gabriel. “IF-Logic and Truth-Definition”. In: *Journal of Philosophical Logic* 27 (1998), pp. 143–164.

Sandu, Gabriel and Jaakko Hintikka. “Aspects of Compositionality”. In: *Journal of Logic, Language and Information* 10.1 (2001). TY - JOUR, pp. 49–61.

Sandu, Gabriel and Tapani Hyttinen. “IF Logic and the Foundations of Mathematics”. In: *Synthese* 126 (2001), pp. 37–47.

Sanford, David H. “What is a Truth Functional Component?” In: *Logique et Analyse* 13 (1970), pp. 483–486.

Sarkar, Hussain. “Anti-Realism Against Methodology”. In: *Synthese* 116 (1998), pp. 379–402.

Schantz, Richard. “Truth and Reference”. In: *Synthese* 126 (2001), pp. 261–281.

Schmidtz, David. “Charles Parsons on the Liar Paradox”. In: *Erkenntnis* 32 (1990), pp. 419–422.

Sellars, Wilfrid. “Presupposing”. In: *Philosophical Review* 63 (1954), pp. 197–215.

Serény, György. “Gödel, Tarski, Church, and the Liar”. In: *The Bulletin of Symbolic Logic* 9.1 (2003), pp. 3–25.

Sher, Gila. “Truth, Logical Structure, and Compositionality”. In: *Synthese* 126 (2001), pp. 195–219.

Shieh, Sanford. “On the Conceptual Foundations of Anti-Realism”. In: *Synthese* 115 (1998), pp. 33–70.

Simmons, Keith. “Reference and Paradox”. In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 230–252.

— “The Diagonal Argument and the Liar”. In: *Journal of Philosophical Logic* 19 (1990), pp. 277–303.

Sirridge, Mary. “Buridan: ‘Every Proposition is False’ is False”. In: *Notre Dame Journal of Formal Logic* XIX.3 (1978), pp. 397–404.

Skyrms, Brian. “Definitions of Semantical Reference and Self-Reference”. In: *Notre Dame Journal of Formal Logic* XVII.1 (1976), pp. 147–148.

Slater, B. H. “Paraconsistent Logics?” In: *Journal of Philosophical Logic* 24 (1995), pp. 451–454.

— “Syntactic Liars”. In: *Analysis* 62.2 (2002), pp. 107–09.

Slater, Hartley. “Liar Syllogisms and Related Paradoxes”. In: *Analysis* 51 (1991), pp. 146–153.

— “Tarski’s Hidden Assumption”. In: *Ratio* 17.1 (2004), pp. 84–89.

Sloman, Aaron. “Tarski, Frege and the Liar Paradox”. In: *Philosophy* 26 (1971), pp. 133–147.

Smiley, Timothy. “Can Contradictions be True? I”. In: *The Aristotelian Society: Supplementary Volume* 67 (1993), pp. 17–34.

Smith, Nicholas J.J. “The Principle of Uniform Solution (of the Paradoxes of Self-Reference)”. In: *Mind* 109.433 (2000), pp. 117–122.

Soames, Scott. “Précis of Understanding Truth”. In: *Philosophy and Phenomenological Research* 65.2 (2002), pp. 397–401.

— “Presupposition”. In: *Handbook of Philosophical Logic*. Ed. by Dov Gabbay and F. Guenther. Vol. IV. Dordrecht: D. Reidel Pub. Co., 1989, pp. 553–616.

— “Replies”. In: *Philosophy and Phenomenological Research* 65.2 (2002), pp. 429–452.

Sobel, Jordan Howard. “Lies, Lies, and More Lies: A Plea for Propositions”. In: *Philosophical Studies* 67 (1992), pp. 51–69.

Sorensen, Roy. “Faking Munchausen’s Syndrome”. In: *Analysis* 60.2 (2000), pp. 202–208.

Sorenson, Roy. *A Brief History of Paradox*. Oxford: Oxford University Press, 2003.

— “A Definite No-No”. In: *Liars and Heaps: New Essays on Paradox*. Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 225–229.

Spade, Paul Vincent. “John Buridan on the Liar: A Study and Reconstruction”. In: *Notre Dame Journal of Formal Logic* XIX.4 (1978), pp. 579–590.

— “On a Conservative Attitude Toward Some Naive Semantic Principles”. In: *Notre Dame Journal of Formal Logic* XVI.4 (1975), pp. 597–602.

— “Recent Research on Medieval Logic”. In: *Lies, Language and Logic in the Late Middle Ages*. Varirum Reprints, 1988, pp. 3–18.

— “The Origins of the Mediaeval *Insolubilia* - Literature”. In: *Lies, Language and Logic in the Late Middle Ages*. Varirum Reprints, 1988, pp. 3–18.

Spade, Paul Vincent and Gordon Antony Wilson. “Introduction”. In: *Johannes Wyclif: Summa Insolubilium*. Ed. by Paul Vincent Spade and Gordon Antony Wilson. Binghampton, New York: Medieval & Renaissance Texts & Studies, 1986, pp. ix–xxxviii.

Stirton, William R. “Anti-Realism, Truth-Conditions and Verificationism”. In: *Mind* 106.424 (1997), pp. 697–716.

Stiver, James L. “Presupposition and Entailment”. In: *Southern Journal of Philosophy* 13 (1975), pp. 485–497.

— “Presupposition, Implication and Necessitation”. In: *Southern Journal of Philosophy* 13 (1975), pp. 99–108.

Stjernberg, Fredrik. “The Contingent and the A Priori”. In: *Theoria* 66.1 (2000), pp. 83–85.

Strawson, Peter F. "A Reply to Mr. Sellars' "Presupposing"". In: *Philosophical Review* 63 (1954), pp. 216–231.

— "Identifying Reference and Truth-Values". In: *Logico-Linguistic Papers.* London: Methuen & Co. Ltd, 1971, pp. 75–95.

— *Introduction to Logical Theory.* London: Methuen & Co. Ltd., 1952.

— *Introduction to Logical Theory.* London: Methuen & Co Ltd, 1953.

— "On Referring". In: *Logico-Linguistic Papers.* London: Methuen & Co Ltd, 1971, pp. 1–27.

— "Particular and General". In: *Logico-Linguistic Papers.* London: Methuen & Co. Ltd, 1971, pp. 28–52.

Suter, Ronald. "Strawson's Analysis of Identity Statements". In: *Philosophy and Phenomenological Research* 31 (1971), pp. 597–9.

Szabó, Zoltán Gendler. "Compositionality as Supervenience". In: *Linguistics and Philosophy* 23.450 (2000), pp. 475–505.

— *Problems of Compositonality.* New York: Garland Publishing, Inc., 2000.

— "Review: The Compositionality Papers". In: *Mind* 113.450 (2004), pp. 340–344.

Tappenden, Jamie. "Comments on Soames' Understanding Truth". In: *Philosophy and Phenomenological Research* 65.2 (2002), pp. 418–421.

Tarski, Alfred. "Der Wahrheitsbegriff in den formalisierten Sprachen". In: *Alfred Tarski: Collected Papers.* Vol. 2. Birhäuser, 1986.

— "The Concept of Truth in Formalized Languages". In: *Logic, Semantics and Metamathematics.* Oxford University Press, 1956, pp. 152–278.

— "The Semantic Conception of Truth: and the Foundations of Semantics". In: *Philosophy and Phenomenological Research* 4.3 (1944), pp. 341–376.

Tennant, Neil. *Anti-Realism and Logic: Truth as Eternal.* Oxford: Clarendon Press, 1987.

Tomberlin, James E. "About the Problem of Truth". In: *Philosophy and Phenomenological Research* 27 (1966), pp. 85–89.

— ed. *Language and Mind.* Supplement tobr Noûs. Cambridge, MA: Blackwell Publishers, 2002.

Tzouvaras, Athanassios. "Logic of Knowledge and Utterance and the Liar". In: *Journal of Philosophical Logic* 27 (1998), pp. 85–108.

Van Heijenoort, Jean. "Logic a Calculus and Logic as Language". In: *Synthese* 17 (1966), pp. 324–330.

Visser, Albert. "Four Valued Semantics and the Liar". In: *Journal of Philosophical Logic* 13 (1984), pp. 181–212.

Weir, Alan. "Rejoinder to Laurence Goldstein on the Liar". In: *Analysis* 62.1 (2002), pp. 26–34.

— "Token Relativism and the Liar". In: *Analysis* 60.2 (2000), pp. 156–170.

— "Ultramaximalist minimalism!" In: *Analysis* 56.1 (1996), pp. 10–22.

Werning, Marcus. “Compositionality, Context, Categories and the Indeterminacy of Translation”. In: *Erkenntnis* 60 (2004), pp. 145–178.

Westerståhl, Dag. “On Mathematical Proofs of the Vacuity of Compositionality”. In: *Linguistics and Philosophy* 21 (1998), pp. 635–643.

Williamson, Timothy. “Soames on Vagueness”. In: *Philosophy and Phenomenological Research* 65.2 (2002), pp. 422–428.

Windt, Peter Y. “The Liar in the Prediction Paradox”. In: *American Philosophicl Quarterly* 10.1 (1973), 65–?

Wolenski, Jan. “In Defense of the Semantic Definition of Truth”. In: *Synthese* 126 (2001), pp. 67–90.

Woodruff, Peter W. “Paradox, Truth and Logic. Part I: Paradox and Truth”. In: *Journal of Philosophical Logic* 13 (1984), pp. 213–32.

Wright, Almroth E. *Prolegomena to the Logic Which Searches for Truth.* London: William Heinemann Ltd, 1941.

Wyclif, Johannis. *Summa Insolubilium.* Ed. by Paul Vincent Spade and Gordon Anthony Wilson. Binghampton, New York: Medieval & Renaissance Text & Studies, 1986.

Yablo, Stephen. “Circularity and Paradox”. In: *Self-Reference.* CSLI Publications, 2004. Chap. 8, pp. 139–157.

— “Definitions, Consistent and Inconsistent”. In: *Philosophical Studies* 72 (1993), pp. 147–175.

— “New Grounds for Naive Truth Theory”. In: *Liars and Heaps: New Essays on Paradox.* Ed. by J. C. Beall. Oxford: Clarendon Press, 2003, pp. 312–330.

Yablo, Steve. “Grounding, Dependence and Paradox”. In: *Journal of Philosophical Logic* 11 (1982), pp. 117–37.

Zadrozny, Wlodek. “From Compositional Semantics to Systematic Semantics”. In: *Linguistics and Philosophy* 17 (1994), pp. 329–342.

Zucker, J. I. and R. S. Tragresser. “The Adequacy Problem for Inferential Logic”. In: *The Journal of Philosophical Logic* 7 (1976), pp. 501–516.