Truth & Paradox

Toby Meadows

READ ME:

These notes are still drafts:

- there will be typos;
- there may be errors; and
- I plan to augment them.

That said, they should be complete enough to be useful and I hope you find them so.

I plan to update this document on my website:

https://sites.google.com/site/tobymeadows/.

Unless you've already come from there, it could be worth looking there for a more recent version of this document.

Also, if you do spot any problems or there's something you don't like or understand, I'd like to hear about it. Please drop me an email at:

toby.meadows@gmail.com.

Contents

CHAPTER 1

om liar sentences to Kripke's construction

This week, we're going to build up some of the technical basics involved in understanding the liar paradox and some of the contemporary approaches to it on the market. We'll run through the following sections:

(1) The liar paradox - we'll develop the basic ingredients of the paradox, investigate some problematic solutions and in so doing make things more precise.
(2) Getting formal - we'll formulate the liar paradox in a more precise manner, which will make the problem more pressing.
(3) What do we do? - we'll characterise solutions into three broad classes: revising the rules of truth; revising language; and revising logic.
(4) Tarksi's solution - we outline Tarksi's levelled approach to the paradox.

1.1. The Liar Paradox

1.1.1. An informal version of the liar paradox. Consider the following sentence:

> *This sentence is not true.*

Suppose it is true. Then it *says* that it is not true, so it must not be true. But this contradicts our assumption. So it cannot be true after all. But then this *means* that it is not true that the sentence is not true; i.e., it is true. This is a contradiction: we have shown that it is both true and not true.

So our first question is:

> Why would this be a problem?

The *standard answer* to this is that that we have argued to an inconsistency. Thus, we have two choices, either:

(1) we accept the inconsistency and go on with our lives; or
(2) we take the inconsistency as indicating that something is wrong in our understanding of the liar sentence.

Now if we go with (1) and assume that our ordinary reasoning is, roughly, like classical logic - *a controversial assumption* - then the fact that we hold inconsistent beliefs would give us reason to believe anything at all. This is known as *trivality.*

Let's give a quick demonstration of this. First recall the definition of consequence.

DEFINITION 1. Let $\Gamma \cup \{\varphi\} \subseteq Sent_{\mathcal{L}}$. Then φ *is a consequence of* Γ, abbreviated $\Gamma \models \varphi$ if for every model $\mathcal{M}$ of $\mathcal{L}$, it is the case that

$$\text{if } \mathcal{M} \models \gamma \text{ for all } \gamma \in \Gamma, \text{ then } \mathcal{M} \models \varphi.$$

REMARK 2. The definition is also often given such that $\Gamma \models \varphi$ if every semantic evaluation function v for $\mathcal{L}$ is such that:

$$\text{if } v(\gamma) = 1 \text{ for all } \gamma \in \Gamma, \text{ then } v(\varphi) = 1.$$

For example, Priest [2008] tends to use this notation. There is no significant different here. Models $\mathcal{M}$ and semantic evaluation functions v are really just the same thing. For any $\mathcal{M}$ there is a v such that for all $\varphi \in Sent_{\mathcal{L}}$

$$\mathcal{M} \models \varphi \iff v(\varphi) = 1$$

and vice versa. However, in non-classical contexts with more than two truth values, it is less convenient to use the $\mathcal{M}$ notation.

It is also worth noting that the $\models$ symbol has two different meanings here. One for consequence and one for truth in a model. This is known as *overloading*, but it doesn't cause problems since the truth in a model relation requires that model $\mathcal{M}$ be on the right hand side of the $\models$, while the consequence relation may use a set of sentences Γ in that place.

Using this it is easy to show how triviality arises.

THEOREM 3. *$\psi, \neg\psi \models \varphi$ for arbitrary $\varphi, \psi \in Sent_{\mathcal{L}}$.*

PROOF. Let $\mathcal{M}$ be an arbitrary model of $\mathcal{L}$. Then it is not the case that:

- $\mathcal{M} \models \psi$; and
- $\mathcal{M} \models \neg\psi$.

Thus letting $\Gamma = \{\psi, \neg\psi\}$, we have

$$\text{if } \mathcal{M} \models \gamma \text{ for all } \gamma \in \Gamma, \text{ then } \mathcal{M} \models \varphi$$

since the conditional is *vacuously satisifed*. Moreover, since $\mathcal{M}$ was arbitrary, we have shown that $\Gamma \models \varphi$ as required. □

It's probably worth noting that the proof of Theorem 3 has a *sneaky* feel about it. We get validity because there is no model $\mathcal{M}$ in which the antecedent of the key condition is true; it has nothing to do with the conclusion at all. While this is just how the classical "if ..., then ..." works, this has been a source of concern to logicians in the *relevant logic tradition*.

The upshot of this is that if we have already concluded that ψ and $\neg\psi$ are true, then we are warranted in concluding that φ is true for any φ whatsoever.[1] So if we have concluded that the liar sentence and its negation are both true, then we should be able to conclude that $2 + 2 = 5$ and indeed that

[1] Strictly, since I'm using quite epistemic language, I should take a detour through the completenss theorem and note that we have $\varphi, \neg\varphi \vdash \psi$ where $\vdash$ says that there is a *derivation* of ψ from assumptions φ and $\neg\varphi$. This is because a derivation (or proof) has a more plausible link with giving us warrants to draw conclusions. This point could be important in the case of second order logic, for example, where no completeness proof is available. However, it's probably not too important for our present discussion.
It is also worth noting that I'm assuming transitivity of the consequence relation in my reasoning above.

$2 + 2 \neq 5$. Thus if we are connecting our consequence relation to our beliefs, then we'd be warranted in believing anything. This seems problematic.

For this reason, we usually disregard option (1) and take up (2) accepting that there is something wrong with our understanding of what is involved in the liar paradox.[2]

This suprising conclusion might lead us to the following line of inquiry:

(A) We have a demonstration that triviality ensues if we accept that the liar sentence is both true and false, so we accept that something is awry in our understanding of the liar paradox.

(B) But it's also really clear that I don't believe or infer just any old sentence.

(C) So what kind of problem is this?

There are two aspects to a good answer here and we start to tread into more contentious territory in Philosophy of Logic.

First we might think of this as a *descriptive problem*. In giving my informal sketch of the liar paradox, I have something wrong in my description of what is that we *do* in the face of the liar sentence. (B) appears to be evidence for this claim. This buys into thinking of philosophical logic as providing a description of what we actually do when we reason. The underlying methodology is close in spirit to linguistics or parts of cognitive science. It is not a very popular view. However, Dave Ripley and Paul Egre have done some interesting work in this area.

On the other hand, we might see this as a *normative problem*. We wouldn't want to say that, presented with the liar sentence, you *should* use that to infer anything you like. Thus, our problem is to set out a better way of dealing with that situation with the eventual goal of telling us what we *ought* to do with the liar. This view buys into the idea that logic provides us with a way of regimenting our reasoning in a uniform manner and in such a way as to tell us what we should infer in new situations.

I'm not sure that we should see these as alternative views engaged in philosophical debate, so much as different programmes in philosophical logic with quite different sets of ambitions.

[2] See Azzouni [2007], Azzouni, Eklund [2002], Patterson [2006] for recent trends which appear to accept (1) in some fashion.

1.1.2. Possible causes of our problem. Accepting that something is awry, brings us to the question of what. In this section, we'll look at a few options:

(1) Can we do without the truth predicate?
(2) Is self reference problematic?
(3) Are inferences based on what a sentence *say* or *means* problematic?

We'll make a few remarks about them here, but the following section will demonstrate that each of them are non-starters.

1.1.2.1. *Can we do without the truth predicate?* It's obvious that truth is playing a starring role here. Perhaps we can do without it. This isn't a common response, but let's make a couple of quick remarks before we move on.

Really the important point here is that this gives us the opportunity to say why we want a truth predicate. There could be a very long and contentious list here, but we'll satisfy ourselves with just two reasons.

First, the truth predicate allows us to make *indefinite generalisations*. If I want to endorse everything Crispin just said, I have a couple of options:

(1) Suppose Crispin just said the sentences φ, ψ, and χ. Then I could say $\varphi \wedge \psi \wedge \chi$; or
(2) I could just say, "All the sentences Crispin just said are *true*."

So with a truth predicate I can describe a set of sentences and then say that they're all true. In the example above, the truth predicate seems useful, but hardly necessary. However, we may want to articulate more difficult sets of sentences. For example, we might want to say.

> Everything that will every be said by a philosopher is true.

Since we're now describing an indefinite collection, it doesn't seem so obvious that we could take the other option; i.e., there doesn't seem to be a collection of sentences that we can just conjoin together.

More pointedly, I might want to say that all of the axioms of ZFC are true. There are infinitely many axioms of ZFC, so there is no way to articulate this using a conjunction of sentences.

So the upshot here is that:

- the truth predicate is *useful* for providing efficient ways of asserting large collections of sentences; and
- the truth predicate is *required* for asserting infinite collections of sentences.

The second point is that the truth predicate allows us to do semantics. This is most obvious in the case of Davidson's truth conditional semantics. Here we take for granted that we know how the truth predicate works and we use the truth predicate to break complicated sentences down into the component conditions required for the complex sentence to be true. In this way an account of meaning is provided, for which truth is the foundation.

It may then seem that model-theoretic accounts like those of Kaplan [Kaplan, 1989, 1978] are off the hook as they don't make a substantive use of the truth predicate *per se*. However, this is a little misleading. While this style of account does not use a truth predicate, they do make use of satisfaction relations and intepretation functions (like $\models$ and v described above). The prototype for these tools is the truth predicate; indeed Tarski's work in model theory grew directly out of his work on truth. Thus, while these accounts are not exposed to the liar paradox, they are exposed to paradoxes which are directly analogous.

I should note that paradox doesn't really beset Davidsonian or model-theoretic accounts of semantics. This is because both of these accounts avail themselves of a version of Tarksi's solution, which we'll visit properly in Section 1.4. However, we might for the moment note that the crucial element of the solution is that these accounts do not provide a semantics for the truth predicate (satisfaction relation, evaluation function) themselves. Leaving truth out of the account still leaves plenty of interesting work for philosophers of language and linguists, but we might think that the project to give our language a semantics is not complete until truth too is also accommodated.

1.1.2.2. *Is self reference problematic?* An obvious idiosyncracy of the liar sentence is that it refers to itself via the indexical "*this*". Self-reference is a difficult matter to get a good grip on. The fact that the liar sentence says something about itself has a certain vertiginous quality. Moreover, we're much more accustomed to sentence which talk about things other than themselves. For a particularly exciting example:

> All dogs are mammals.

Perhaps there's something fishy about self-reference and perhaps we'd do better to avoid it. This was Wittgenstein's approach in the *Tractatus* [Wittgenstein, 2001].[3]

In the following section, we'll show that the liar paradox can be formulated without genuine self-reference and thus head off this line of investigation. However, it's also worth making a more positive remark about self-reference. Consider the sentence:

> This sentence contains 37 characters.

It doesn't. But the very fact that we can so confidently say this demonstrates that self-reference may not be so problematic after all.

1.1.2.3. *Are inferences based on what a sentence* say *or* means *problematic?* A final worry is that the reasoning we used above made use of inferences involving what a sentence *means* or *says*. These are murky concepts to characterise and reasoning with them can lead to problems and fallacies. This, of course, doesn't licence us to conclude that the argument taking us from the liar sentence to inconsistency is unsound. However, it should tell us that a more thoroughgoing analysis is required. There are two options we might take here:

(1) We could provide a sufficiently satisfying analysis of meaning that shows that the argument goes through and could not be reasonably revised to a weaker form which would block the argument.
(2) We could find a way of getting a version of the paradox that avoids meaning altogether.

In the next section, we shall investigate option (2). This seems like a better way to go in that any controversy that could have emerged in the account of meaning developed in option (1) is avoided. However, in Section 1.2.4.1, we'll see an account which does exploit option (1).

1.2. Getting formal

In this section we are going to formulate a version of the liar paradox which avoids (strict) self-reference and semantic vocabularly other than the truth predicate. Our strategy has three steps:

[3] Of course, he did do it by saying that no proposition can refer to itself and in so doing referred to that very proposition. Another pesky ladder to be pushed away.

(1) We find a canonical way of representing, or better, *coding* the sentences of our language. Our goal is to find an interpreted language with sufficient expressive resources that it is, in some sense, able to talk about its own syntax.

(2) We find a way of *simulating self-reference* and use this to produce a sentence that says of itself that it is not true.

(3) We show that such a sentence leads to inconsistency using a canonical natural deduction system of classical first order logic.

1.2.1. Coding.

1.2.1.1. *Coding syntax.* To get things started we need a language to work with. We are going to use the language of arithmetic $\mathcal{L}_{Ar} = \{0, 1, +, \times\}$. We do this because there is an easy way to use natural numbers to represent the formula of $\mathcal{L}_{Ar}$. Moreover, we'll assume that the language of arithmetic is interpreted in the standard way. Thus, we'll work in the standard model of arithmetic, $\mathbb{N} = \{\omega, 0^{\mathbb{N}}, 1^{\mathbb{N}}, +^{\mathbb{N}}, \times^{\mathbb{N}}\}$.

However, I want to stress that this choice is arbitrary. We could have also chosed a language that is able to talk about the syntactical objects themselves. For example, Quine explored theories of this kind in [Quine, 1996]. However, the benefit of using arithmetic is that it is very well understood by logicians and mathematicians.

Rather than providing a formal definition, I'm going to illustrate how the coding works in stages.

First of all we consider the syntax we need to be able to form sentence. This involves both the logical and non-logical vocabulary.

We may do this as described in the following table.

$=$	0	1	$+$	$\times$	$\forall$	$\neg$	$\wedge$	$($	$)$	v
0	1	2	3	4	5	6	7	8	9	10

You'll notice that I need 10 letters to represent all the symbols and unfortunately, I only have 9 arabic numerals to play with. To get around this, I'll just use the letter, a, to play the role of 10 in representing v. Thus, we have

$=$	0	1	$+$	$\times$	$\forall$	$\neg$	$\wedge$	$($	$)$	v
0	1	2	3	4	5	6	7	8	9	a

Then when we come to represent a formula of $\mathcal{L}_{Ar}$, we simply write out the numerals representing each piece of syntax in the same order they appear in the formula.

EXAMPLE 4. Consider the formula $1 + 0 = 1$; we represent this by 23101.

If we want to reprenent numbers other than 0 and 1, say the number 3, then we don't appear to have anything available to represent this. However, this can simply be done by remember that $3 = 1 + 1 + 1$.

EXAMPLE 5. Consider the formula $2 + 2 = 5$. This is reprensented by the number

$$82329382329082323232329.$$

I've used the parenthesis symbols to distinguish the terms representing the 2's. Otherwise, we'd have the same sequence of symbols on either side of the 0, which would make it impossible to recover our formula from the code number.

Our next problem is representing formula using quantification. For this we are going to use the symbol v. However, we may want formulae which use more than one variable. To do this we usually use a list of variables v_0, v_1, v_2. Of course, we have no way of representing subscripts with our meagre vocabularly, but we can just use the parentheses for this. Thus when every we write v_n, we can just write $v(n)$ instead.

EXAMPLE 6. Consider the formula $\forall v_0 \exists v_1 (v_0 = v_1)$. We first re-write this as

$$\forall v(0) \neg \forall \neg v(1)(v(0) = v(1))$$

and then we give it the following number:

$$5a819656a8298a8190a8299.$$

Now given that we have a new numeral a, we might ask what number this represents. The answer to this is that we move from a base 10 system to a base 11 system. Thus,

$$\begin{aligned} 1a7 &= 7 \times 11^0 + \\ &\quad 10 \times 11^1 + \\ &\quad 1 \times 11^2 \\ &= 7 + 11 + 121 \\ &= 139. \end{aligned}$$

So now we know how to assign formulae of $\mathcal{L}_{Ar}$ natural numbers, but this still doesn't gives us a way of representing then in the language $\mathcal{L}_{Ar}$. For this we use the trick of re-writing n as $1 + ... + 1$ (n many times). This will then be the final code representing a formula. We write $\ulcorner \varphi \urcorner$ for the appropriately long string of $1 + ... + 1$.

EXAMPLE 7. Consider the formula $0 = 0$. This is represented by the number 101, which in base 10 is

$$1 \times 11^2 + 1 \times 11^0 = 122.$$

So $\ulcorner 0 = 0 \urcorner = 1 + 1 + ... + 1$ with 122, 1s.

As you can see, codes get very big very quickly. But this isn't a problem, there are plenty of natural numbers.

1.2.1.2. *Recursive functions.* So this gives us the tools to represent formulae in the language of arithmetic. However, we want and are able to do more than that: we can also use the language of arithmetic to say a lot of interesting things about those codes. To make this clearer I'm going to introduce the notion of a *recursive function.*

But first, we start with the notion of an *algorithm.* Essentially, an algorithm is a set of instructions for how to perform a pencil and paper calculation of some kind. It's idealised in the sense the we might need an infinite amount of paper and the calculation might never finish.

For example, Turing developed his Turing machines to make this idea more precise. He used an infinite piece of tape consisting of squares which are either blank or contain the symbol 1. We are then permitted to move to the left or right, read the square upon which we land and to either leave it as it, or change it. In this model an algorithm is a set of instructions which meet these constraints.

From here we may define a recursive function:

DEFINITION 8. A *(partial) recursive function* is a (partial) function which takes an algorithm and a Turing tape as input and returns as output the Turing tape that results one there are no more instructions left to complete - if such a stage occurs; otherwise the output is undefined.

REMARK. (i) Note there are cases where the instructions would never be completed. For example, if we instructed the machine to simply keep moving to the right. This is analogous to what happens when a computer programme hangs (or gets stuck in a loop). This is why the functions are partial.

(ii) Note that the notion of of a partical recursive function is *extensional* while an algorithm is *intensionsal*. There could be many algorithms which give the same partial recursive function. This is similar to the relationship between sets and properties.

It turns out that any computable function that anyone has every come up with can be modelled in this simple system: this is its power. However, there are many other models which have the same capacity. For example, Gödel provided a schematic way of representing algorithms, which yield exactly the same recursive functions where the model of computation is the natural numbers rather than Turing tapes. I won't write out a full definition here, but this leads us to an alternative definition.

DEFINITION 9. A *(partial) recursive function* is a (partial) function which uses an instance of Gödel schema as instructions, takes a natural number as input and gives a natural number as output - if it has one; and is undefined otherwise.

Given that we're working in the language of arithmetic, this definition is particularly convenient. But there are many more ways of producing models for computation. This leads to the Church-Turing thesis.

THESIS: The functions computable via a pen-and-paper algorithm are exactly the partial recursive function.

So the important thing here is the the left-hand-side of the thesis is informal. We are saying that any function performable by an idealised person can be modelled using a partial recursive function. Then using Definition 9, we can find an algorithm from the Gödel schema which gives the same partial function.

Now we just need one more thing: a way of connecting the Gödel schema to the language of arithmetic. The following theorem, which we won't prove, covers this:

THEOREM 10. *Let e be an algorithm satisfying the Gödel schema and let $f_e : \omega \rightharpoonup \omega$ be the partial recursive function derived from it. Then there is some formula $\psi_e(v_0, v_1, v_2)$ from $\mathcal{L}_{Ar}$ such that*

$$f_e(m) = n \;\Leftrightarrow\; \mathbb{N} \models \exists x \psi_e(m, n, x).$$

With this theorem, we can now take any pen-and-paper algorithm and represent it with a formula of arithmetic.

1.2.1.3. *Saying things about codes in arithmetic.* This puts us in a position to say things about syntax using formulae of arithmetic.

EXAMPLE 11. Consider the property of being a well-formed formula of the language of arithmetic. It should be clear that we could write a set of instructions that, if followed, will tell us whether or not some string of symbols is well-formed or not. Then:

(1) by the Church-Turing thesis, we can then see that there is partial recursive function corresponding to this set of instructions;
(2) so it can be represented by an algorithm from the Gödel schema; and
(3) thus by some formula $WFF(x)$ from the language of arithmetic such that

$$x \text{ is well-formed } \leftrightarrow \mathbb{N} \models WFF\ulcorner x \urcorner.$$

EXAMPLE 12. Let us consider the operation of substitution. So take a well-formed formula $\psi(v_0)$ and consider the process of substituting the symbol a for every v_0 occuring in $\psi(v_0)$. We write this as $\psi(a)$. A little reflection should convince us that we can write out a set of instructions (i.e., an algorithm) for this process. Thus, by the same argument as in the previous example, we can find a formula $Sub(v_0, v_1, v_2)$ in the language of arithmetic such that

$$\psi(a) \text{ is the result of substituting } a \text{ for } v_0 \text{ in } \psi(v_0)$$

$$\leftrightarrow$$

$$\mathbb{N} \models Sub(\ulcorner \psi(a) \urcorner, \ulcorner a \urcorner, \ulcorner \psi(v_0) \urcorner)$$

REMARK 13. So $Sub(k, m, n)$ is true in $\mathbb{N}$ if n is the code of a formulae $\psi(v)$ with one free variable and k is the code number of the formula resulting from substituting the numeral representing m into the free variable space in ψ; i.e., $z = \ulcorner\psi(\underline{\mathrm{m}})\urcorner$.

Indeed, these examples illustrate the the following more general principle holds:

> Any property of syntax that can be verified by a pen-and-paper algorithm can be represented by a formula of $\mathcal{L}_{Ar}$ in the standard model of arithmetic.

1.2.2. Something like self-reference. We are now, finally, in a position to simulate the effect of self-reference using arithmetic coding.

THEOREM 14. *(Semantic diagonal lemma) Let $\varphi(x)$ be a formula from $\mathcal{L}_{Ar}$ with at most one free variable. Then there is some sentence G in the language of arithmetic such that:*

$$\mathbb{N} \models G \leftrightarrow \varphi\ulcorner G\urcorner.$$

PROOF. Define $A(x)$ to be such that:

$$A(x) := \exists y(Sub(x, x, y) \wedge \varphi(y)).$$

Let $a = \ulcorner A(x)\urcorner$; $G := A(a)$ and $g = \ulcorner G\urcorner$. Then clearly,

$$\mathbb{N} \models \forall y(Sub(a, a, y) \leftrightarrow y = g).$$

Thus,

$$\begin{aligned} & \mathbb{N} \models A(a) \leftrightarrow \exists y(Sub(a, a, y) \wedge \varphi(y)) \\ \Leftrightarrow \quad & \mathbb{N} \models A(a) \leftrightarrow \exists y(y = g \wedge \varphi(y)) \\ \Leftrightarrow \quad & \mathbb{N} \models A(a) \leftrightarrow \varphi(g) \\ \Leftrightarrow \quad & \mathbb{N} \models G \leftrightarrow \varphi\ulcorner G\urcorner. \end{aligned}$$

□

The theorem still holds in language with are expansions of $\mathcal{L}_{Ar}$. Let us add a new predicate T, to $\mathcal{L}_{Ar}$ and call the result $\mathcal{L}_T$.

1.2.3. Inconsistency again. We are now ready to prove our formal version of the liar paradox.

First let us use Theorem 14, to get a sentence λ from $\mathcal{L}_T$ such that:

$$\mathbb{N} \models \lambda \leftrightarrow \neg T\ulcorner\lambda\urcorner.$$

Intuitively, this sentence says of itself that it is not true. However, note that no genuine self reference has occured here. In the standard model $\mathbb{N}$ we have that λ is the case iff λ's code is not in the extension of the truth predicate.

Next we need to say how the truth predicate works. For ths, we'll employ Tarski's T-schema.

(T-schema) For all $\varphi \in Sent_{\mathcal{L}_T}$, $\varphi \leftrightarrow T\ulcorner\varphi\urcorner$.

The underlying idea here is a kind of transparency. If φ is the case then so is the sentence saying that φ is true and *vice versa*.

We can now demonstrate the problem more precisely.

$$\dfrac{\dfrac{\dfrac{T\ulcorner\lambda\urcorner^{(1)} \quad \dfrac{}{T\ulcorner\lambda\urcorner \to \lambda}\,Thm}{\lambda} \quad \dfrac{T\ulcorner\lambda\urcorner^{(1)} \quad \dfrac{}{T\ulcorner\lambda\urcorner \to \neg\lambda}\,Ax}{\neg\lambda}}{\bot}\,(1)}{\neg T\ulcorner\lambda\urcorner}$$

Thus we see that $\vdash \neg T\ulcorner\lambda\urcorner$; and so

$$\dfrac{\dfrac{\dfrac{}{\neg T\ulcorner\lambda\urcorner}\,Thm \quad \dfrac{}{\neg T\ulcorner\lambda\urcorner \to \neg\lambda}\,Thm}{\neg\lambda} \quad \dfrac{\dfrac{}{\neg T\ulcorner\lambda\urcorner}\,Ax \quad \dfrac{}{\neg T\ulcorner\lambda\urcorner \to \lambda}\,Thm}{\lambda}}{\bot}$$

And so the system is is inconsistent.

REMARK. Notice that *self-reference* and problems about *meaning* have not played any part here.

So what have we shown here. Informally, we've shown the language of arithmetic cannot support a truth predicate. However, there are really two results contained here.

FACT 15. *(i) The truth predicate is* ***undefinable****: there is no formula $\psi(x) \in \mathcal{L}_{Ar}$ which is such that for all sentence $\varphi \in \mathcal{L}_{Ar}$:*

$$\mathbb{N} \models \psi\ulcorner\varphi\urcorner \leftrightarrow \varphi.$$

*(ii) The truth predicate is **inadmissible**: no model $\mathcal{M}$ containing a coding system may be expanded with a relation symbol T and interpreted in such a way that for all $\varphi \in \mathcal{L}_T$:*

$$\langle \mathcal{M}, T^{\mathcal{M}} \rangle \models T\ulcorner \varphi \urcorner \leftrightarrow \varphi.$$

The first result is more important to mathematical logic. The second is more important in philosophy.

REMARK 16. The upshot of this result is that the distinction between object language and metalanguage is not mere pragmatic convenience but and absolute necessity.

EXERCISE 17. There is a certain similarity between the liar sentence and Gödel's undecidable sentence. Does the Gödel sentence lead to inconsistency? Why not?

1.2.4. Other ways of getting liars.

1.2.4.1. *Contingent liar sentences.* Consider the following sentence:

> *The first sentence written in italics in Section 1.2.4.1 is not true.*

By Tarski's T-schema we have:

> p iff S is true,

where p is the proposition expressed by the sentence S.

Combining these we have:

> The first sentence written in italics in Section 1.2.4.1 is not true iff the first sentence written in italics in Section 1.2.4.1 is true.

This is clearly a contradiction.

More formally we might reconstrue this as follows:

- Let $I(x)$ mean that x is the first sentence written in italics in Section 1.2.4.1.

We might model the liar sentence above as: $\forall x(Ix \to \neg Tx)$.

Suppose $\forall x(Ix \to \neg Tx)$. We take it as a fact that $\exists! x Ix$. Fix such an object and call it a. Then we have $\neg Ta$. Since there is only on object a such that Ia, we have $a = \ulcorner\forall x(Ix \rightarrow \neg Tx)\urcorner$ and so we have $\neg\forall x(Ix \rightarrow \neg Tx)$. Let b be such that Ib and Tb. Then since there is a unique object b such that Ib we have $b = \ulcorner\forall x(Ix \rightarrow \neg Tx)\urcorner$; and so $\forall x(Ix \rightarrow \neg Tx)$: which is a contradiction. Thus $\neg\forall x(Ix \rightarrow \neg Tx)$.

Let b witness this; i.e., $Ib \wedge Tb$. Then this gives us $\forall x(Ix \rightarrow \neg Tx)$. But then we have $\neg Tb$ and so $\neg\forall x(Ix \rightarrow \neg Tx)$: contradiction.

1.2.4.2. *Liars via indexicals.* Consider the following sentence:

> *This sentence is not true.*

Using a semantics for indexicals as given by Kaplan in [1989, 1978], we could give this sentence a formal semantic representation.

1.2.4.3. *Liars via the metalanguage.* We can also make liar sentence by just assuming that our object language has names to represent all the sentences of $\mathcal{L}_T$. So rather than assuming that our object language possesses the expressive resources to define a coding system, we build it up, so to speak, from the outside.

For example, we might suppose we are working in a language $\mathcal{L}$ with an interpretation $\mathcal{M}$ in which every formula φ of $\mathcal{L}$ has a constant symbol c_φ representing it. Approaches along these lines are taken by Kremer in [1988] and Ripley in [Forthcoming].

However, they are not without there oddities. Suppose we start with a language $\mathcal{L}$, which contains a truth predicate T, and inductively enrich the language with a set of constant symbols C such that:

- every sentence of $\mathcal{L}$ has a constant symbol; and
- every new sentence which can be formed from from the new constant symbols and $\mathcal{L}$ is represented by a constant symbol.

Call the result of the obvious way of doing this, $\mathcal{L}^+$. Then it turns out that there will be no sentence λ in $\mathcal{L}^+$ such that $\lambda = \neg Tc_\lambda$. There are no liar sentence in $\mathcal{L}^+$.

This is a little weird. We're trying to develop a language capable of talking about its own syntax and we don't get a liar sentence. The usual solution to this is to just stipulate that there is such a sentence. So we say that $\mathcal{L}$ contains a sentence λ such that $\lambda = \neg Tc_\lambda$.

1.3. What should we do?

So now we know there is a problem with the truth predicate which has nothing to do with semantic reasoning or self-reference. How should we address it? Feferman partitions the space of solutions among those which revise:

(1) truth;
(2) language; or
(3) logic.

Along each path we must accept that something very intuitive and obvious does not work as we supposed it did [Feferman, 1991]. We'll see each of these in more detail below, but here is quick overview of the landscape.

1.3.1. Truth principle revisers.

The first group revise the way in which the truth predicate works. The only rule we have enforced for the truth predicate is the T-schema, so one direction must go.

For example, in Leitgeb's [2005], we are given a system in which:

$T\ulcorner\varphi\urcorner \to \varphi$ for all $\varphi \in Sent_{\mathcal{L}_T}$

is the case but we do **not** have

$\varphi \to T\ulcorner\varphi\urcorner$ for all $\varphi \in Sent_{\mathcal{L}_T}$.

1.3.2. Language revisers.

The second group opts to revise our conception of language and takes it that there are multiplicity of languages each of which has a truth predicate, but not of which contains that truth predicate. So for example, given the language of arithmetic, Tarski provides a truth predicate for that language such that

$T\ulcorner\varphi\urcorner \leftrightarrow \varphi$ for all $\varphi \in Sent_{\mathcal{L}_{Ar}}$.

So we do have a version of the truth predicate for $\mathcal{L}_{Ar}$, but this is not a truth predicate for the full language $\mathcal{L}_T$.

1.3.3. Logic revisers.

The final group, with which we'll be particularly concerned are the logic revisers. The two most famous examples here are:

(1) Kripke's gappy approach which allows a third truth value to represent a kind of indeterminacy [1975]; and

(2) Priest's glutty approach which takes it the argument to inconsistency is correct and the liar sentence is actually both true and false [1979].

1.4. Tarski's solution

1.4.1. Truth for arithmetic. We now examine Tarski's solution to the liar paradox. First of all we show how one might expand $\mathcal{L}_{Ar}$ with a truth predicate which satisfies the T-schema for sentences of $\mathcal{L}_{Ar}$.

Let $WFF_{\mathcal{L}_{Ar}}(x)$ be a formula of $\mathcal{L}_{Ar}$ which is true of x iff it is a well-formed formula of $\mathcal{L}_{Ar}$.

Let $Atom_{\mathcal{L}_{Ar}}(x)$ be a formula of $\mathcal{L}_{Ar}$ which is true of x iff it is an atomic formula of $\mathcal{L}_{Ar}$.

Let $TAtom_{\mathcal{L}_{Ar}}(x)$ be a formula of $\mathcal{L}_{Ar}$ which is true of x iff it is a true atomic formula of $\mathcal{L}_{Ar}$.

REMARK 18. Note that we can devise a simple pen-and-paper algorithm to verify whether an atomic formulae (i.e., something for the form $t = s$ for terms of $\mathcal{L}_{Ar}$) is correct or not; thus it may be represented by a formula of $\mathcal{L}_{Ar}$.

Let $x^\frown y$ be a function represented by a formula of $\mathcal{L}_{Ar}$ such that $x^\frown y$ is the concatenation of x and y.

Let $sub(y, n, i)$ be a function represented by a formula of $\mathcal{L}_{Ar}$ such that $sub(y, n, i)$ is the result of substituting the numeral n for v_i wherever it is very in the formula coded by y.

Let $\Phi(T)$ be the following sentence of $\mathcal{L}_{Ar}$ - for all x

- if $WFF_{\mathcal{L}_{Ar}}(x)$, then
  - if $Atom_{\mathcal{L}_{Ar}}(x)$, then $T(x) \leftrightarrow TAtom_{\mathcal{L}_{Ar}}(x)$;
  - $\forall y$, if $x = \ulcorner \neg \urcorner ^\frown y$, then $T(x) \leftrightarrow \neg T(y)$;
  - $\forall y \forall z$, if $x = y^\frown \ulcorner \wedge \urcorner ^\frown z$, then $T(x) \leftrightarrow T(y) \wedge T(z)$; and
  - $\forall y \forall i$, if $x = \ulcorner \forall v_i \urcorner ^\frown y$, then $T(x) \leftrightarrow \forall n \, T(sub(y, n, i))$; and
- if $\neg WFF_{\mathcal{L}_{Ar}}(x)$, then $\neg Tx$.

THEOREM 19. *Let $\mathcal{M}$ be a model for $\mathcal{L}_T$ which is an expansion of the standard model of arithmetic $\mathbb{N}$. This means that $\mathcal{L}$ interprets the vocabularly of $\mathcal{L}_{Ar}$*

in the same way as $\mathbb{N}$, but the interpretation of the truth predicate T is not constrained. Then if $\mathcal{M} \models \Phi(T)$, we have

$$\mathcal{M} \models \varphi \leftrightarrow T\ulcorner\varphi\urcorner$$

for all $\varphi \in \mathcal{L}_{Ar}$.

PROOF. By induction on the complexity of sentences. □

This is pretty close to what we want. We have the T-schema true for all sentence in the language $\mathcal{L}_{Ar}$, but this does not extend to the full language $\mathcal{L}_T$ with the truth predicate. For sentences $\varphi \in \mathcal{L}_T \backslash \mathcal{L}_{Ar}$ (the sentences involving the truth predicate) we have

$$\mathcal{M} \models \neg T\ulcorner\varphi\urcorner.$$

This is why the liar sentence λ is not a problem. If we look at how it is constructed, we see that it makes use of the truth predicate and thus $\mathcal{M} \models \neg\lambda$.

But at the same time this causes other problems. For example, while we have

$$\mathcal{M} \models T\ulcorner 0 = 0\urcorner,$$

we have

$$\mathcal{M} \not\models T\ulcorner T\ulcorner 0 = 0\urcorner\urcorner.$$

This seems very strange!

1.4.1.1. *Ordinary semantics.* There is a sense that when we do model-theoretic semantics we are taking up this limited solution. Standardly, we do semantics from the perspective of a metalanguage where some kind of satisfaction predicate $\models$ is used to tell us whether or not a sentence is true in a particular model. This allows us to develop a theory of the meaning of our expressions.

However, the satisfaction relation is not itself part of the object language. Thus, our semantic theory is not capable of giving us a theory of the meaning of the satisfaction relation: our semantics is, in some sense, incomplete.

We might then try to add the satisfaction relation to the object relation to the object language, but this will result in a paradox very similar to the liar.

EXERCISE 20. Describe the paradox of satisfaction.

1.4.2. A Tarskian Solution. So our problem was that we only got the T-schema for sentences of the restricted language $\mathcal{L}_{Ar}$ and not $\mathcal{L}_T$. One way of addressing this is to characterise a second truth predicate, $T^\dagger$, which gets the T-schema for sentences of $\mathcal{L}_T$. Let $\mathcal{L}_{T,T^\dagger}$ be $\mathcal{L}_T$ expanded by the new truth predicate $T^\dagger$.

Let $\Phi^\dagger(T^\dagger, T)$ say that for all x

- if $WFF_{\mathcal{L}_T}(x)$, then
  - if $Atom_{\mathcal{L}_{Ar}}(x)$, then $T^\dagger(x) \leftrightarrow TAtom_{\mathcal{L}_{Ar}}(x)$;
  - $\forall y$, if $x = \ulcorner Ty \urcorner$, then $T^\dagger x \leftrightarrow Ty$;
  - $\forall y$, if $x = \ulcorner \neg \urcorner {}^\frown y$, then $T^\dagger(x) \leftrightarrow \neg T^\dagger(y)$;
  - $\forall y \forall z$, if $x = y^\frown \ulcorner \wedge \urcorner {}^\frown z$, then $T^\dagger(x) \leftrightarrow T^\dagger(y) \wedge T^\dagger(z)$; and
  - $\forall y \forall i$, if $x = \ulcorner \forall v_i \urcorner {}^\frown y$, then $T^\dagger(x) \leftrightarrow \forall n\, T^\dagger(sub(y, n, i))$; and
- if $\neg WFF_{\mathcal{L}_T}(x)$, then $\neg T^\dagger x$.

Then we have the following:

THEOREM 21. *If $\mathcal{M}$ is a model of $\mathcal{L}_{T,T^\dagger}$ which is an expansion of $\mathbb{N}$ and $\mathcal{M} \models \Phi(T) \wedge \Phi^\dagger(T^\dagger, T)$ then for all $\varphi \in \mathcal{L}_T$*

$$\mathcal{M} \models T^{\dagger} \ulcorner \varphi \urcorner \leftrightarrow \varphi.$$

PROOF. By induction on the complexity of sentences. □

Of course, we can formulate a new liar sentence $\lambda^\dagger$ using the new truth predicate $T^\dagger$ which is such that:

$$\mathcal{M} \models \lambda^\dagger \leftrightarrow \neg T^{\dagger} \ulcorner \lambda^\dagger \urcorner$$

but if we look back to the diagonal lemma, we see that this sentence will involve the truth predicate and the argument to inconsistency is blocked since

$$\mathcal{M} \models \neg \lambda^\dagger.$$

But as before, we have counter-intuitive features as well. For example, while we have

$$\mathcal{M} \models T^{\dagger} \ulcorner 0 = 0 \urcorner \ \& \ \mathcal{M} \models T^{\dagger} \ulcorner T \ulcorner 0 = 0 \urcorner \urcorner$$

we also have

$$\mathcal{M} \models T^{\dagger} \ulcorner T^{\dagger} \ulcorner 0 = 0 \urcorner \urcorner.$$

1.4.2.1. *Going further still.* We can address this kind of worry by going even further. Let us consider a language with countably many truth predicates $\{T_n \mid n \in \omega\}$.

Let Γ be the infinite set of sentences consisting of the union of the following sentences for each $n \in \omega$,

- if $WFF_{\mathcal{L}_{T_n}}(x)$, then
- if $Atom_{\mathcal{L}_{Ar}}(x)$, then $T_{n+1}(x) \leftrightarrow TAtom_{\mathcal{L}_{Ar}}(x)$;
  - $\forall y$, if $x = \ulcorner T_n y \urcorner$, then $T_{n+1}x \leftrightarrow T_n y$;
  - $\forall y$, if $x = \ulcorner \neg \urcorner ^\frown y$, then $T_{n+1}(x) \leftrightarrow \neg T_{n+1}(y)$;
  - $\forall y \forall z$, if $x = y ^\frown \ulcorner \wedge \urcorner ^\frown z$, then $T_{n+1}(x) \leftrightarrow T_{n+1}(y) \wedge T_{n+1}(z)$; and
  - $\forall y \forall i$, if $x = \ulcorner \forall v_i \urcorner ^\frown y$, then $T_{n+1}(x) \leftrightarrow \forall n \; T_{n+1}(sub(y, n, i))$; and
- if $\neg WFF_{\mathcal{L}_{T_n}}(x)$, then $\neg T_{n+1}x$.

Then we have the following.

THEOREM 22. *in any model $\mathcal{M}$ expanding $\mathbb{N}$ with an intepretation for each T_n for $n \in \omega$ where $\mathcal{M} \models \Gamma$ we have for all $n \in \omega$ and all sentences φ from $\mathcal{L}_{T_n}$*

$$T_{n+1}\ulcorner \varphi \urcorner \leftrightarrow \varphi.$$

PROOF. By induction on truth level and complexity of sentences. □

We can go further if we like, but this will do.

For every level n, there will be a liar sentence λ_n for which the argument to inconsistency is blocked since we have

$$\mathcal{M} \models \neg \lambda_n.$$

1.4.3. Toward Kripke's approach.

1.4.3.1. *Kripke's objection.* Kripke observes that Tarski's solution is problematic. In the first instance, our natural language doesn't have labels on its truth predicates. We know that snow is white but we don't think it's true$_1$ that snow is white. We just say that it's true.

So we have to admit that Tarski's solution fails to provide a description of our ordinary use of the concept of truth.

But this is not the last word against a hierarchical approach. While it is clear that we do not make *explicit* use of indexed truth predicates, perhaps we use the *implicitly*. So perhaps when we use the truth predicate, it really ought to have a label on it, but most of the time it makes no difference so we omit it in our ordinary speech.

The first thing to note is that this retort appears to be drawing a very long bow if it's trying to *describe* our ordinary practice. While it's a fact that we actually don't get into trouble using the truth predicate, this fact is somewhat mysterious. As such, saying that we are actually just lazily using a the immensely hierarchical approach seems far fetched. To this, one might respond by drawing an analogy with grammar. Although people are quite able to speak English in accord with a set of grammatical rules, very few of speakers would be able to describe those rules. Similarly, one might argue that ordinary speaker are able to manipulate the truth concept according in accord with Tarksi's theory of levels, but that should not give us reason to think they were able to outline that theory.

More safely, we could stay with a *normative* approach to truth theories. As such, we might say that Tarksi's hierarchical approach gives us a way of using the concept of truth without falling into paradox. It strikes me that this is just correct. Moreover, I think Kripke's objection has no force here.

Kripke, however, has more to say about the implicit usage defence. He gives the following example, which will be helpful.

EXAMPLE 23. Consider the sentence, as uttered by Dean,

(A) All of Nixon's utterances about Watergate are false.

If we were using the Tarski's hierachy, we might try to select a level for the falsity predicate higher than any level used in Nixon's utterances. Call this the *level selection principle*.

However, in ordinary practice, Dean will not be in a position to know the levels of Nixon's utterances. Moreover, Nixon may have said something like,

(B) Everything Dean says about Watergate is false.

In this situation, we are faced with a kind of bind. If we are to follow the same principle we used for (A) in labelling the truth predicate used implicitly in (B), we are supposed to chose a level higher than any used in Dean's relevant utterances. But clearly (A) is one of those utterances.

This means that the level selection principle is not viable.

The upshot of this the following conclusion:

> We cannot assign levels to sentence on the basis of syntactic form alone.

There are two related reasons for this:

(1) It's often implausible that we could know the levels of all the statements captured by a universal quantification; and
(2) Using one obvious principle for level assignment leads to problems.

Remark 24. This is a *terrible* argument, but I think the conclusion is correct. While (1) should be taken seriously in that it highlights a real problem for any plausible descriptive project, it has no impact on a normative project. However, (2) is just too weak for us to lever the conclusion. So what if one method of assigning levels doesn't work! The conclusion is that no method can do it. Charitably, I think we should read Kripke's use of the level selection principle as illustrative and suggestive of a deeper and more interesting argument, which can be provided.

Now where does Tarksi's argument stand after this. I think we should agree that there is no way of assigning levels to sentences on the basis of the syntactic form alone. But still there is a further turn available for the Tarskian. Perhaps there is some fact of the matter regarding which level a statement should be situated. This kind of metaphysical solution may seem somewhat implausible, but, in fact, we'll see that Kripke's solution leads us into exactly the same situation. Kripke's construction gives every statement a level. So while we might see this as a problem for Tarski, Kripke isn't giving us much better.

From this, I take it that while the argument lead us to an interesting result (which is not proven), it has limited impact on the Tarskian. The most troubling problem is the initial one: Tarksi forces us to use a hierarchy of truth predicates and this is in poor faith with our actual practice.

1.4.3.2. *Toward Kripke's solution.* With regard to this problem, Kripke provides us with a solution. We'll develop this properly next week, but for now I'll make a few motivating remarks.

The problem we're facing in the liar paradox should be reminiscient of the problem faced in set theory by Russell's paradox. Let's consider two kinds of solution to Russell's paradox.

(1) *Ramified type theory* - This was Russell's solution. It's very difficult to describe so I'll settle for givig a quick desciption of its simpler cousin: the simple theory of types. So you've probably heard of second order logic. In this logic we are able to quantify not just over individuals but also over classes of individuals. We can also have third order logic where we have individuals, classes and families of classes. In fact for any n we can set up n^{th} order logic. Simple type theory is theory we get when we combine all the n^{th} order logics together. It avoids Russell's paradox since the indexing of each level stops us from being able to form the Russell set.

(2) ZFC *set theory* - With this approach we start with a single relation symbol $\in$ and form a theory of sets, where sets can be members of sets which are members of sets and so on. Russell's paradox is avoided by weaking comprehension to the Separation Axiom.

We don't have time to get into too much detail here, but the important think to note is that Russell's type theoretic approah bears a strong resemblance to Tarski's hierarchy. With Tarski, we have a hierarchy of languages and with Russell we have a hierachy of types of collection. In each case, paradox is avoided by breaking things into discrete levels.

However, both approaches also have the problem of fragmenting intuitive concepts. Tarksi fragments truth, while Russell fragments the membership relation. ZFC, however, provides a way of keeping a univocal membership relation and still avoiding paradox. A similar approach to truth would provide a solution to Kripke's problem. We might describe the situation a follows:

Rammified type theory is to Tarski's truth hierarchies

as

ZFC set theory is to $?$

Next week, we'll see how Kripke's approach fills this gap.

CHAPTER 2

Kripke's construction, its cousins & what you can do with them

This week we're going to focus on modern solution to the liar paradox. We'll run through the following sections:

1. Kripke's solution - we describe the basics of Kripke's approach to the paradox.
2. The research landscape of formal theories of truth.
3. Consistency and non-triviality - what else can you do with Kripke's construction?

2.1. Kripke's solution

2.1.1. An intuitive sketch. We might see the idea for Kripke's construction emerging out of the following observations:

1. There's nothing weird about say that $0 = 0$ is true; or more formally, $T\ulcorner 0 = 0\urcorner$.
2. Similarly, there's nothing weird about saying $T\ulcorner T\ulcorner 0 = 0\urcorner\urcorner$ or $T\ulcorner T\ulcorner T\ulcorner 0 = 0\urcorner\urcorner\urcorner$.
3. No matter how many truth predicates you put in front of $0 = 0$, the result is still true.
4. But there's something different about a sentence which says of itself that it isn't true, or indeed, that it is true.

There's something *safe* about the sentences in (1) and (2), but the sentences described in (4) are risky or downright inconsistent.

The essential idea of Kripke's construction is to take truths that are safe and use them to find further safe truths. For example we know that $0 = 0$ is true, so $T\ulcorner 0 = 0\urcorner$ is true and so is $T\ulcorner T\ulcorner 0 = 0 = 0\urcorner\urcorner$. On the other hand, we know that the liar sentence is not safe at all, so we want to avoid it.

Our goal is to find an extension which we'll denote, Γ_{sK}, for the truth predicate T. Γ_{sK} will be a set of of true sentence.

We're going to define this inductively using transfinite recursion. We'll set this out formally soon, but for the moment we'll just outline the main idea of the induction.

We'll index the symbol Γ with an ordinal α to indicate how far we have gone in the process. (More formally, this means that we have a function $\Gamma_{\cdot} : \mathbf{On} \to \mathcal{P}(Sent_{\mathcal{L}_T})$.) The process works like this:

1. Start off by assuming that nothing is in the extension of the truth predicate. Call this Γ_0.
2. Then add all the sentences φ which are true in the standard model of arithmetic and call this Γ_1.
3. Now see what sentences would become true if we assumed Γ_1 was the extension of the truth predicate. So we treat Γ_1 as a kind of educated *guess*. Call the result Γ_2.
4. Use Γ_2 as a guess to obtain Γ_3.
5. Keep doing this, until nothing more can be done.

2.1.2. Formal definition. So this should give some feeling for how the construction works, but we need to be more precise if we're to see how this gets us of the trouble cause with the liar paradox.

The basic idea is to leave λ in a kind of semantic *gap*. To this we use a strong Kleene evaluation scheme. I'll give a slightly different version of this to the one you've seen. Rather than defining a semantic evaluation function with three values, I'm going to define it so that it only has two value and is sometimes undefined. A function of this kind is known as a *partial function*. From a logical point of view, the difference is merely cosmetic. From a philosophical view, this manner of presentation *perhaps* gives some insight as to why Kripke argued that his construction still had a classical semantics.

This way of doing semantics requires us to revise the way we interpret relation symbols. Consider a 1-place relation symbol P in the ordinary semantics for first order logic. When we intepret P in a model we get an extension $P^{\mathcal{M}}$. This the set of things from the domain M of $\mathcal{M}$ which satisfy P, i.e.,

$$\{d \in M \mid \mathcal{M} \models Pd\}.$$

If we want to get the things that don't satisfy P, i.e.,

$$\{d \in M \mid \mathcal{M} \not\models Pd\}.$$

In the classical setting, this is the same as the set of objects which satisfy the formula $\neg Pv_0$, i.e.,

$$\{d \in M \mid \mathcal{M} \models \neg Pd\}.$$

There is a sense in which $\neg$ and $\models$ commute; i.e., their order is unimportant. However, with a partial semantics we do things differently. Given a 1-place relation symbol P, the set of objects that don't satisfy P and those which satisfy $\neg Pv_0$ are not necessarily the same. The reason for this is that there may be objects $d \in M$ such that it is not defined whether d is in the extension of P or not. These objects are not in the extension of P but still may not satisfy $\neg Pv_0$.

To make sense of this, we interpret relations as having both an extension an an anti-extension. More formally, given a language $\mathcal{L} = \{P\}$, we say that $\mathcal{M} = \langle M, \langle P^{+\mathcal{M}}, P^{-\mathcal{M}} \rangle\rangle$ is a model for $\mathcal{L}$ if

- $P^{+\mathcal{M}} \subseteq M$ (the extension of P);

- $P^{-\mathcal{M}} \subseteq M$ (the anti-extension of P); and
- $P^{+\mathcal{M}} \cap P^{-\mathcal{M}} = \emptyset$.

This can be easily generalised to models with constant symbols and multiple relation symbols. Function symbols are more difficult.

REMARK 25. The main difference between a partial model and a classical model is that the extension and anti-extension of a relation symbols are not required to partition the (appropriate product of the) domain into an extension and anti-extension which *exhausts* the domain.

In our application to truth, we're going to make things a little easier on ourself:

- We'll generate our construction over the standard model of arithmetic;
- We'll fix the interpretation of all non-logical vocabularly except the truth predicate; and
- We'll make the truth predicate the only partial predicate in the model.

This means that the only thing that can change is our interpretation of the truth predicate. Intuitively, the construction will allow us to build better and better interpretations or guesses about the extension and anti-extension of truth.

We'll need some way of indicating which guesses we are considering. Let's define this more formally.

DEFINITION 26. Let $Guess$ be the set of pairs $\langle \Phi^+, \Phi^- \rangle$ of sets of sentences from $\mathcal{L}_T$ such that

$$\Phi^+ \cap \Phi^- = \emptyset.$$

We use upper case Greek letters, $\Phi, \Psi, \Gamma, \Delta$... to denote guesses; and write Φ^+ and Φ^- to denote the extension component and anti-extension component of Φ respectively. Moreover, $\Phi = \langle \Phi^+, \Phi^- \rangle$.

We are now ready to define our semantic evaluation predicate. Recall that our sketch construction was set up so that we used the truth *guess* from the previous level in order to define the new level. Our semantic evaluation predicate is designed to do this - as can be seen from the initial typing.

Intuitively, Val takes a *guess* (which is a partial interpretation of the truth predicate) and a sentences and tell us if it is true or not according to that guess, if it can.

DEFINITION 27. We define the partial function $Val : Guess \times Sent_{\mathcal{L}_T} \rightharpoondown 2$ by recursion on the complexity of sentences as follows:

$$\begin{array}{lll} Val_\Phi(\varphi) = 1 & \text{iff} & (\varphi \in AAtom \wedge \varphi \in AArith) \vee \\ & & (\varphi := T\ulcorner\psi\urcorner \wedge \psi \in \Phi^+) \vee \\ & & (\varphi := (\neg\psi) \wedge Val_\Phi(\psi) = 0) \vee \\ & & (\varphi := (\psi \wedge \chi) \wedge Val_\Phi(\psi) = 1 \wedge Val_\Phi(\chi) = 1) \vee \\ & & (\varphi := (\forall x\psi) \wedge \forall n \in \omega\ Val_\Phi(\psi_x(\underline{\mathbf{n}})) = 1)) \\ Val_\Phi(\varphi) = 0 & \text{iff} & (\varphi \in AAtom \wedge \varphi \notin AArith) \vee \\ & & (\varphi := T\ulcorner\psi\urcorner \wedge \psi \in \Phi^-) \vee \\ & & (\varphi := (\neg\psi) \wedge Val_\Phi(\psi) = 1) \vee \\ & & (\varphi := (\psi \wedge \chi) \wedge (Val_\Phi(\psi) = 0 \vee Val_\Phi(\chi) = 0) \vee \\ & & (\varphi := (\forall x\psi) \wedge \exists n \in \omega\ Val_\Phi(\psi_x(\underline{\mathbf{n}})) = 0)) \end{array}$$

If $Val_\Phi(\varphi)$ is undefined for some guess Φ and sentence φ, we shall write $Val_\Phi(\varphi) = \infty$.

EXAMPLE 28. Suppose $\Phi = \langle \emptyset, \emptyset \rangle$. Then $Val_\Phi(T\ulcorner 0 = 0\urcorner) = \infty$.

EXERCISE 29. Let $\Phi = \langle \emptyset, \emptyset \rangle$. What is the value of:

(1) $Val_\Phi(0 = 0)$;
(2) $Val_\Phi(\neg T\ulcorner 0 = 0\urcorner)$?

2.1.2.1. *Jump function.* With the Val function defined, we are ready to define the operation used at the successor stage of our induction definition. This is the logical engine which allows us to move from stage to stage in the definition. This is often known as the *jump function*. It takes us from one guess to a (hopefully) better guess - note the initial typing of the function. It is defined as follows:

DEFINITION 30. Let $j : Guess \to \mathcal{P}(Sent_{\mathcal{L}_T})^2$ be such that:

$$j(\Phi) = \langle \{\varphi \mid Val_\Phi(\varphi) = 1\}, \{\varphi \mid Val_\Phi(\varphi) = 0\} \rangle$$

So the intuitive idea here is that given guess Φ, we may form a better guess $j(\Phi)$ by:

- taking as our new extension, $j(\Phi)^+$, the sentences which are true according to Φ; and
- taking as our new anti-extension $j(\Phi)^-$, the sentences which are false according to Φ.

REMARK 31. Note that the range of j is subset of $Guess$. Since Val is a function, given any Φ and φ we never have

$$Val_\Phi(\varphi) = 1 \ \ \& \ \ Val_\Phi(\varphi) = 0.$$

2.1.2.2. *Truth definition (sK).* With the jump function in hand, we can now set out the induction construction which gives us a sequence of improving guesses about the extension and anti-extension of the truth predicate.

DEFINITION 32. Let $\Gamma_\cdot : \mathbf{On} \rightharpoondown Guess$ be a partial function defined as follows:

$$\begin{aligned} \Gamma_0 &:= \langle \emptyset, \emptyset \rangle \\ \Gamma_{\alpha+1} &:= j(\Gamma_\alpha) \\ \Gamma_\lambda &:= \langle \bigcup_{\alpha<\lambda} \Gamma_\alpha^+, \bigcup_{\alpha<\lambda} \Gamma_\alpha^- \rangle \text{ for limit } \lambda. \end{aligned}$$

This function is partial in the sense that it could fail to be defined at some ordinal. The only thing that could go wrong is that we may not be able to apply the jump function at some ordinal. The only reason this could occur is that $\Gamma_\alpha \notin Guess$ for some α; i.e., $\Gamma_\alpha^+ \cap \Gamma_\alpha^- \neq \emptyset$. If that were to occur then the jump function simply could not be applied.

This means that we cannot apply the Transfinite Recursion Lemma just yet. We need to show a little more. But once that is we'll actually see that the function is total; i.e., defined over all the ordinals. We prove a short sequence of lemmas to establish the result, after which we'll be able to define Γ_{sK}.

DEFINITION 33. (i) Let Φ and Ψ be guesses. Let $\Phi \sqsubseteq \Psi$ if $\Phi^+ \subseteq \Psi^+$ and $\Phi^- \subseteq \Psi^-$;
(ii) Let $\Phi \sqcup \Psi = \langle \Phi^+ \cup \Psi^+, \Phi^- \cup \Psi^- \rangle$; and
(iii) Let $\bigsqcup_{\alpha<\beta} \Phi_\alpha = \langle \bigcup_{\alpha<\beta} \Phi_\alpha^+, \bigcup_{\alpha<\beta} \Phi_\alpha^- \rangle$.

LEMMA 34. *j is* montonic*: i.e., if $\Delta \subseteq \Gamma$, then $j(\Delta) \sqsubseteq j(\Gamma)$.*

PROOF. By induction on the complexity of sentences. Here are a couple of cases.

Suppose $\varphi := \psi \wedge \chi$. Suppose $(\psi \wedge \chi) \in j(\Delta)^+$. Then

$$Val_\Delta(\psi \wedge \chi) = 1 = Val_\Delta(\psi) = Val_\Delta(\chi).$$

Then by induction,

$$1 = Val_\Gamma(\psi) = Val_\Gamma(\chi) = Val_\Gamma(\psi \wedge \chi).$$

Suppose $\varphi := T\ulcorner\psi\urcorner$. Then suppose$T\ulcorner\psi\urcorner \in j(\Delta)^+$. This means that $Val_\Delta(T\ulcorner\psi\urcorner) = 1$ and so $\ulcorner\psi\urcorner \in \Delta^+ \subseteq \Gamma^+$. This then means that $Val_\Gamma(T\ulcorner\psi\urcorner) = 1$ and so $T\ulcorner\psi\urcorner \in j(\Gamma)^+$.

□

LEMMA 35. *If* $\Gamma_\beta \in Guess$ *for all* $\beta < \alpha$*, then* $\Gamma_\alpha \subseteq \Gamma_{\alpha+1}$*.*

PROOF. By transfinite induction. Suppose for all $\beta < \alpha$, $\Gamma_\beta \sqsubseteq \Gamma_{\beta+1}$ (the induction hypothesis). We show that the claim also holds for α.

Suppose $\alpha = 0$. Then

$$\emptyset = \Gamma_0^+ \subseteq \{\varphi \in \mathcal{L}_{Ar} \mid \mathbb{N} \models \varphi\} = \Gamma_1.$$

Similarly for the anti-extension.

Suppose $\alpha = \beta + 1$. Then since $\Gamma_\beta \sqsubseteq \Gamma_{\beta+1}$ we have $j(\Gamma_\beta) \sqsubseteq j(\Gamma_{\beta+1})$ by Lemma 34. Thus,

$$\Gamma_\alpha = \Gamma_{\beta+1} = j(\Gamma_\beta) \sqsubseteq j(\Gamma_{\beta+1}) = \Gamma_{\beta+2} = \Gamma_{\alpha+1}.$$

Suppose α is a limit. Then $\Gamma_\alpha = \bigcup_{\beta<\alpha} \Gamma_\beta$. Thus, for all $\beta < \alpha$, we have:

- $\Gamma_\beta \sqsubseteq \Gamma_{\beta+1} = j(\Gamma_\beta)$;
- $\Gamma_\beta \sqsubseteq \Gamma_\alpha$; and
- $j(\Gamma_\beta) \sqsubseteq j(\Gamma_\alpha)$.

Thus for all $\beta < \alpha$, $\Gamma_\beta \sqsubseteq j(\Gamma_\alpha)$; i.e.,

$$\Gamma_\alpha = \bigsqcup_{\beta<\alpha} \Gamma_\beta \sqsubseteq j(\Gamma_\alpha) = \Gamma_{\alpha+1}.$$

□

We now come to our key lemma. It has two parts. The first shows us that our Γ function is totally well-defined and the second part will allow us to define Γ_{sK}.

LEMMA 36. *(i)* $\Gamma^+_\beta \cap \Gamma^-_\beta = \emptyset$ *for all* β *(*Γ_β *is always a guess); and*

(ii) For $\alpha \leq \beta$*,* $\Gamma_\alpha \sqsubseteq \Gamma_\beta$ *(Non-strict increasing-ness).*

PROOF. We prove (i) and (ii) together by transfinite induction. Let us assume as an induction hypothesis that for all $\delta < \beta$ we have:

(1) $\Gamma^+_\delta \cap \Gamma^-_\delta = \emptyset$; and
(2) $\forall \alpha \leq \delta$, $\Gamma_\alpha \sqsubseteq \Gamma_\delta$.

We show that these claims also hold for β.

We first prove (i).

Suppose $\beta = 0$. Then $\Gamma_0 = \langle \emptyset, \emptyset \rangle$; thus, $\Gamma^+_0 \cap \Gamma^-_0 = \emptyset$.

Suppose $\beta = \delta + 1$. The definition of the jump function ensures that the claim is upheld. See Remark 31.

Suppose β is a limit ordinal and for a contradiction suppose that $\Gamma^+_\beta \cap \Gamma^-_\beta \neq \emptyset$. Then there must be some φ such that $\varphi \in \Gamma^+_{\delta_1}$and $\varphi \in \Gamma^-_{\delta_2}$ for $\delta_1, \delta_2 < \beta$, since nothing new is added at stage β. Suppose $\delta_1 < \delta_2$. But by (2) of our induction hypothesis $\Gamma_{\delta_1} \sqsubseteq \Gamma_{\delta_2}$ thus, we have $\varphi \in \Gamma_{\delta_2}$ which means that $\Gamma^+_{\delta_2} \cap \Gamma^+_{\delta_1} \neq \emptyset$ contradicting (1) of our induction hypothesis. Similarly if $\delta_2 < \delta_1$.

Now we prove (ii).

Suppose $\beta = 0$. Suppose $\beta = 0$; thus, $\Gamma_0 \sqsubseteq \Gamma_0$.

Suppose $\beta = \delta + 1$. Suppose $\alpha = \delta + 1 = \beta$; then it is clear that $\Gamma_\alpha \sqsubseteq \Gamma_\beta$.

So suppose $\alpha \leq \delta$. Then by (2) of the induction hypothesis, we have $\Gamma_\alpha \sqsubseteq \Gamma_\delta$. Thus,

$$\Gamma_{\alpha+1} = j(\Gamma_\alpha) \sqsubseteq j(\Gamma_\delta) = \Gamma_{\delta+1} = \Gamma_\beta$$

Then by (1) of the induction hypothesis, we see that $\Gamma_\alpha \in Guess$. Thus by Lemma 35, $\Gamma_\alpha \sqsubseteq \Gamma_{\alpha+1}$, and so $\Gamma_\alpha \sqsubseteq \Gamma_\beta$ as required.

Suppose β is a limit ordinal. Then $\Gamma^+_\beta = \bigcup_{\delta \in \beta} \Gamma^+_\delta$, so $\Gamma^+_\alpha \subseteq \Gamma^+_\beta$. Similar for the anti-extension. □

COROLLARY 37. $\Gamma_{\cdot} : \mathbf{On} \rightarrow Guess$ *is a well-defined total function.*

PROOF. By transfinite recursion. □

THEOREM 38. *There is some countable* α *such that* $\Gamma_\alpha = \Gamma_{\alpha+1}$*.*

PROOF. First observe that $|Sent_{\mathcal{L}_T}| = \omega$. The easiest way to see this is by recalling that we have a method of coding every sentence by a natural number. Thus there are at least as many natural numbers as sentences of $\mathcal{L}_T$. It should also be clear that there is a term of $\mathcal{L}_T$ for every natural numbers. Putting this together we see that the cardinality of the sentences of $\mathcal{L}_T$ must be ω.

For a contradiction suppose that for $\alpha \in \aleph_1$, $\Gamma_{\alpha+1} \neq \Gamma_\alpha$. Then by Lemma 36, we must have

$$\Gamma^+_{\alpha+1} \supsetneq \Gamma^+_\alpha$$

and

$$\Gamma^-_{\alpha+1} \supsetneq \Gamma^-_\alpha$$

for all $\alpha \in \aleph_1$. But this would mean that we could find an injection from the $\aleph_1$ into $Sent_{\mathcal{L}_T}$. But this gives us

$$\aleph_1 \leq |Sent_{\mathcal{L}_T}| = \omega$$

which is impossible. □

REMARK. Intuiviely speaking, the idea of this proof there must be some $\alpha < \omega_1$ such that $\Gamma_\alpha = \Gamma_{\alpha+1}$ for at some countable point in the construction we must run our of sentences to add.

DEFINITION 39. Let Γ_{sK} be Γ_α for the least α such that $\Gamma_\alpha = \Gamma_{\alpha+1}$. Let α_{sK} be that ordinal.

PROPOSITION 40. $\Gamma_{sK} = j(\Gamma_{sK})$.

REMARK 41. Γ_{sK} is known as a *fixed point* of the jump function j since when we apply j to Γ_{sK} it remains fixed.

THEOREM 42. *For $\varphi \in Sent_{\mathcal{L}_T}$ the following are equivalent:*

(1) $Val_{\Gamma_{sK}}(T\ulcorner\varphi\urcorner) = 1$;
(2) $Val_{\Gamma_{sK}}(\varphi) = 1$; and
(3) $\varphi \in \Gamma^+_{sK}$.

PROOF. (1↔3) By the definition of Val, we have

$$Val_{\Gamma_{sK}}(T\ulcorner\varphi\urcorner) = 1 \;\Leftrightarrow\; \varphi \in \Gamma^+_{sK}.$$

(3→2) Observe that

$$\begin{aligned} \varphi \in \Gamma^+_{sK} &\Leftrightarrow \varphi \in j(\Gamma_{sK})^+ \\ &\Leftrightarrow Val_{\Gamma_{sK}}(\varphi) = 1. \end{aligned}$$

The first $\Leftrightarrow$ is from Proposition 40 and the second is by definition of j. □

So this pretty good.

2.1.2.3. *Examples.*

EXAMPLE 43. $T\ulcorner 0 = 0\urcorner \in \Gamma^+_{sK}$.

We now want to look again at what happens to the liar sentence λ. However, we need to make a little remark first. Since we've moved into a partial model, we might wonder whether the diagonal lemma still works. It turns out that, in the sense that we need, that it does still work.

LEMMA 44. *Let $\psi(v_0)$ be any formula of $\mathcal{L}_T$, then there is some sentence γ such that*

$$Val_\Phi(\gamma) = Val_\Phi(\psi\ulcorner\gamma\urcorner).$$

PROOF. Clearly for any *r.e.* function $f : \omega \to \omega$, there will be a formula $\varphi_f(x, y)$ such that

$$f(m) = n \Leftrightarrow Val_\Gamma(\varphi_f(\underline{\mathbf{m}}, \underline{\mathbf{n}})) = \top.$$

Thus the Sub predicate will do the usual thing.

The hitch comes in that the truth table for $\leftrightarrow$ works as follows:

$\leftrightarrow$	1	∞	0
1	1	∞	0
∞	∞	∞	∞
0	0	∞	1

Let's work through the standard argument and trace the blocking point.

Let $\beta(x) \leftrightarrow_d \exists y(Diag(x, y) \wedge \psi(y))$.

Let $b = \ulcorner\beta(x)\urcorner$.

Let $\gamma = \beta(b)$. Let $g = \ulcorner\gamma\urcorner$.

Then it is easy to see that:

$$Val_\Phi(\forall y(y = g \leftrightarrow Sub(b, b, y)) = 1.$$

Moreover, it is clear that no undefined values will be involved here.

We then try to proceed by claiming that:

$$Val_\Phi(\gamma \leftrightarrow \gamma) = 1.$$

However, since we are in strong Kleene there is no guarantee that this will be the case. In those cases where it does hold, the usual proof suffices.

So we have another case to deal with; i.e., where

$$Val_\Phi(\gamma \leftrightarrow \gamma) = \infty.$$

This tell us that $Val_\Gamma(G) = \infty$. Now unpacking the definition of G we then see that

$$Val_\Phi(\exists y(Diag(b, y) \wedge \psi(y))) = \infty.$$

But we already know that $Val_\Phi(Diag(b, g)) = 1$ (it's just a simple syntax calculation).

But then we see that $Val_\Phi(\psi(g)) = \infty$, as required.

Thus, we see that for any formula $\psi(x)$, there is some γ such that

$$Val_\Phi(\psi\ulcorner\gamma\urcorner)) = Val_\Phi(\gamma).$$

□

REMARK 45. So we use exactly the same syntactic technique to get our liar sentence, it just has slightly different properties in the partial semantics.

LEMMA 46. $\lambda \notin \Gamma^+_{sK} \cup \Gamma^-_{sK}$.

PROOF. Suppose for a contradiction that $\lambda \in \Gamma^+_{sK}$. But then

$$\begin{array}{rcl} \lambda \in \Gamma^+_{sK} & \Leftrightarrow & \lambda \in j(\Gamma_{sK})^+ \\ & \Leftrightarrow & Val_{\Gamma_{sK}}(\lambda) = 1 \\ & \Leftrightarrow & Val_{\Gamma_{sK}}(\neg T\ulcorner\lambda\urcorner) = 1 \\ & \Leftrightarrow & Val_{\Gamma_{sK}}(T\ulcorner\lambda\urcorner) = 0 \\ & \Leftrightarrow & \lambda \in \Gamma^-_{sK} \end{array}$$

which contradicts (i) of Lemma 36. A similar argument shows $\lambda \notin \Gamma^-_{sK}$. □

Let *app* be the primitive recursive function which behaves as follows:

- $app(0) = \ulcorner 0 = 0\urcorner$;
- $app(n + 1) = Tapp(n)$.

Since there is a obvious set of instructions for computing this function, we know that it can be represented by a formula of $\mathcal{L}_{Ar}$.

EXAMPLE 47. $\forall n\ T\ app(n) \in \Gamma^{+}_{sK}$.

2.1.2.4. *Limitations.* The T-schema is not true in our intended model: i.e.,

LEMMA 48. $Val_{\Gamma_{sK}}(\varphi \leftrightarrow T\ulcorner\varphi\urcorner) \neq 1$ *for some* $\varphi \in Sent_{\mathcal{L}_T}$.

PROOF. Let λ be a liar sentence. Then by definition of λ we have

$$Val_{\Gamma_{sK}}(\lambda) = Val_{\Gamma_{sK}}(\neg T\ulcorner\lambda\urcorner).$$

Moreover, by Theorem 42, we have

$$Val_{\Gamma_{sK}}(\lambda) = Val_{\Gamma_{sK}}(T\ulcorner\lambda\urcorner).$$

Thus

$$Val_{\Gamma_{sK}}(\neg T\ulcorner\lambda\urcorner) = Val_{\Gamma_{sK}}(T\ulcorner\lambda\urcorner)$$

and so $Val_{\Gamma_{sK}}(\lambda) = Val_{\Gamma_{sK}}(T\ulcorner\lambda\urcorner) = \infty$ and this means that

$$Val_{\Gamma_{sK}}(\lambda \leftrightarrow T\ulcorner\lambda\urcorner) = \infty.$$

Thus,

$$Val_{\Gamma_{sK}}(\lambda \leftrightarrow T\ulcorner\lambda\urcorner) = \infty.$$

□

The conditional in strong Kleene logic is too weak for this.

2.1.3. Revision theory. What would happen if we used the jump function on a guess that was exhaustive.

- Belnap and Gupta

2.2. Contemporary research in formal theories of truth

There are three main projects:

(1) Semantic theories of truth;
(2) Axiomatic theories of truth; and
(3) Logics of truth.

2.2.1. Semantic theories of truth.

- Given a particular model or world, which sentences are true? [Kripke, 1975, Leitgeb, 2005, Gupta and Belnap, 1993]

2.2.2. Axiomatic theories of truth.

- What is the best theory of truth? [Halbach, 2011, Cantini, 1990, Halbach and Horsten, 2006]

2.2.3. Logics of truth.

- How should we reason about truth? [Kremer, 1988, Beall, 2009, Priest, 1979, Ripley, Forthcoming]

2.3. Consistency & non-triviality

2.3.1. KF.

2.3.1.1. *The theory KF.*

DEFINITION. (15.2) The theory KF is given by the axioms of PA with induction revised to accommodate sentences involving the truth predicate and the following axioms:

$$
\begin{array}{ll}
(KF1) & \forall s \forall t (T(s \dot{=} t) \leftrightarrow s^o = t^o) \\
(KF2) & \forall s \forall t (T(s \dot{\neq} t) \leftrightarrow s^o \neq t^o) \\
(KF3) & \forall x (Sent_T(x) \rightarrow (T(\dot{\neg}\dot{\neg} x) \leftrightarrow Tx)) \\
(KF4) & \forall x (Sent_T(x \dot{\wedge} y) \rightarrow (T(x \dot{\wedge} y) \leftrightarrow Tx \wedge Ty)) \\
(KF5) & \forall x (Sent_T(x \dot{\wedge} y) \rightarrow (T\dot{\neg}(x \dot{\wedge} y) \leftrightarrow T(\dot{\neg} x) \vee T(\neg \dot{y}))) \\
(KF8) & \forall v \forall x (Sent_T(\dot{\forall} v x) \rightarrow (T(\dot{\forall} v x) \leftrightarrow \forall t\, T(x(t/v)))) \\
(KF9) & \forall v \forall x (Sent_T(\dot{\forall} v x) \rightarrow (T(\dot{\forall} v x) \leftrightarrow \exists t\, T(\dot{\neg} x(t/v)))) \\
(KF12) & \forall t (T(\dot{T} t) \leftrightarrow T t^o) \\
(KF13) & \forall t (T \dot{\neg} \dot{T} t \leftrightarrow (T \dot{\neg} t^o \vee \neg Sent_T(t^o)))
\end{array}
$$

REMARK. In the above definition we write s^o to denote the value of the term s. This function can quite easily be seen to be primitive recursive. Thus, it can be represented by a (Σ^0_1) arithmetic formula. To take an example, let $t = 6543$ and that in our coding system 6543 represents the term '$5 + 4$'. Then $t^o = 9$.

2.3.1.2. *Closed off theories.*

FACT 49. *For all $\varphi \in Sent_{\mathcal{L}_T}$, $\varphi \in \Gamma^+_{sK}$ iff $(\neg\varphi) \in \Gamma^-_{sK}$.*

So we can just get by with an extension for the truth predicate and revise the definitions above accordingly.

However, the T-schema is not true in our intended model: i.e.,

$$Val_{\Gamma_{sK}}(\varphi \leftrightarrow T\ulcorner\varphi\urcorner) = \infty.$$

FACT 50. *(i) $\lambda \notin \Gamma_{sK}$; (ii) but $\langle \mathbb{N}, \Gamma_{sK}\rangle \models \lambda$.*

PROOF. (i) Suppose there is some α such that $\lambda \in \Gamma_\alpha$.

Then $Val_{\Gamma_\alpha}(\neg T\ulcorner\lambda\urcorner) = Val_{\Gamma_\alpha}(\lambda) = 0$, so $\lambda \notin j(\Gamma_\alpha) = \Gamma_{\alpha+1}$: contradicting non-strict increasingness. □

2.3.1.3. *The consistency of KF.*

THEOREM 51. *Let $\mathcal{M}$ be the standard model of arithmetic $\mathbb{N}$, expanded with an interpretation for T which is the extension of the truth predicate given by the minimal fixed point of the strong Kleene evaluation scheme. Then*

$$\mathcal{M} \models KF.$$

PROOF. So we let Γ^+_{sK} be the extension of the truth predicate in a classical model.

We show that each axiom of KF is true in $\mathcal{M}$.

Since $\mathcal{M}$ is an expansion of $\mathbb{N}$, it is clear that $\mathcal{M} \models PA$.

($KF1$) $\forall s\forall t(T\ulcorner s = t\urcorner \leftrightarrow s = t)$ Suppose $\mathcal{M} \models s = t$. Then $(s = t) \in \Gamma^+_0 \subseteq \Gamma^+_{sK}$.

($KF2$) $\forall s\forall t(T(\ulcorner s \neq t\urcorner) \leftrightarrow s \neq t)$ Similar.

($KF3$) $Sent_{\mathcal{L}_T}(x) \rightarrow (T(\ulcorner\neg\neg\urcorner{}^\frown x) \leftrightarrow Tx)$ Suppose $x = \ulcorner\varphi\urcorner \in Sent_{\mathcal{L}}$. Suppose $\mathcal{M} \models T\ulcorner\varphi\urcorner$. Then $\varphi \in \Gamma^+_{sK}$. Thus $Val_{\Gamma_{sK}}(\varphi) = 1$ and so $Val_{\Gamma_{sK}}(\neg\neg\varphi) = 1$ giving us that $(\neg\neg\varphi) \in \Gamma_{sK}$ and so $\mathcal{M} \models T\ulcorner\neg\neg\varphi\urcorner$.

($KF4$) $Sent_{\mathcal{L}_T}(x^\frown\ulcorner\wedge\urcorner{}^\frown y) \rightarrow (T(x^\frown\ulcorner\wedge\urcorner{}^\frown y) \leftrightarrow Tx \wedge Ty))$ Similar.

($KF5$) $\forall x(Sent_T(x\dot{\wedge}y) \rightarrow (T\dot{\neg}(x\dot{\wedge}y) \leftrightarrow T(\dot{\neg}x) \vee T(\neg\dot{y})))$ Similar.

($KF8$) $\forall v\forall x(Sent_T(\dot{\forall}vx) \rightarrow (T(\dot{\forall}vx) \leftrightarrow \forall t\, T(x(t/v)))$ Similar.

($KF9$) $\forall v\forall x(Sent_T(\dot{\forall}vx) \rightarrow (T(\dot{\forall}vx) \leftrightarrow \exists t\, T(\dot{\neg}x(t/v)))$ Similar.

($KF12$) $\forall t(T(\dot{T}t) \leftrightarrow Tt^o)$ Only sentences can be in the extension of the truth predicate so let $t = \ulcorner\varphi\urcorner$. Then

$$\begin{aligned}\mathcal{M} \models T\ulcorner\varphi\urcorner &\Leftrightarrow Val_{\Gamma_{sK}}(T\ulcorner\varphi\urcorner) = 1\\ &\Leftrightarrow Val_{\Gamma_{sK}}(T\ulcorner T\ulcorner\varphi\urcorner\urcorner) = 1\\ &\Leftrightarrow \mathcal{M} \models T\ulcorner T\ulcorner\varphi\urcorner\urcorner.\end{aligned}$$

($KF13$) $\forall t(T\dot{\neg}\dot{T}t \leftrightarrow (T\dot{\neg}t^o \vee \neg Sent_T(t^o)))$. We see that

$$\begin{aligned}\mathcal{M} \models T\ulcorner\neg Tx\urcorner &\Leftrightarrow Val_{\Gamma_{sK}}(T\ulcorner\neg Tx\urcorner) = 1\\ &\Leftrightarrow Val_{\Gamma_{sK}}(\neg Tx) = 1\\ &\Leftrightarrow Val_{\Gamma_{sK}}(Tx) = 0\\ &\Leftrightarrow x \notin Sent_{\mathcal{L}_T} \vee (x = \ulcorner\varphi\urcorner \wedge \varphi \in \Gamma^-_{sK})\\ &\Leftrightarrow x \notin Sent_{\mathcal{L}_T} \vee (x = \ulcorner\varphi\urcorner \wedge Val_{\Gamma_{sK}}(\neg\varphi) = 1)\\ &\Leftrightarrow x \notin Sent_{\mathcal{L}_T} \vee (x = \ulcorner\varphi\urcorner \wedge Val_{\Gamma_{sK}}(T\ulcorner\neg\varphi\urcorner) = 1)\\ &\Leftrightarrow x \notin Sent_{\mathcal{L}_T} \vee \mathcal{M} \models T\ulcorner\neg\urcorner{}^\frown x.\end{aligned}$$

□

COROLLARY 52. *KF is consistent*

2.3.2. LP.

2.3.3. Other approaches - Field, Weber, Brady.

Bibliography

Jody Azzouni. The strengthened liar, the expresssive strength of natural languages and regimentation. *The Philosophical Forum*, 34:329.

Jody Azzouni. The inconsistency of natural languages: How we live with it. *Inquiry*, 50(6):590–605, 2007.

JC Beall. *Spandrels of Truth*. Oxford University Press, 2009.

Andrea Cantini. A theory of truth arithmetically equivalent to ID_1^1. *The Journal of Symbolic Logic*, 55(1):244–259, 1990.

Matti Eklund. Inconsistent langugages. *Philosophy and Phenomenological Research*, 64(2):251–275, 2002.

Solomon Feferman. Reflecting on incompleteness. *Journal of Symbolic Logic*, 56(1):1–49, 1991.

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

Volker Halbach. *Axiomatic Theories of Truth*. Cambridge University Press, London, 2011.

Volker Halbach and Leon Horsten. Axiomatizing kripke's theory of truth. *Journal of Symbolic Logic*, 71:677–712, 2006.

David Kaplan. On the logic of demonstratives. *The Journal of Philosophical Logic*, 8:81–98, 1978.

David Kaplan. Demonstratives. In *Themes from David Kaplan*. OUP, Oxford, 1989.

Michael Kremer. Kripke and the logic of truth. *The Journal of Philosophical Logic*, 17(3):225–278, 1988.

Saul Kripke. Outline of a theory of truth. *Journal of Philosophy*, 72:690–716, 1975.

Hannes Leitgeb. What truth depends on. *The Journal of Philosophical Logic*, 34:155–192, 2005.

Doug Patterson. Tarski, the liar and inconsistent languages. *The Monist*, 89, 2006.

Graham Priest. The logic of paradox. *Journal of Philosophical Logic*, 8:219–241, 1979.

Graham Priest. *An Introduction to Non-Classical Logic: From If to Is*. Cambridge University Press, Melbourne, 2008.

W. V. Quine. Definition of substitution. In *Selected Logic Papers*. Harvard University Press, Cambridge, 1996.

Dave Ripley. Conservatively exending classical logic with transparent truth. *The Review of Symbolic Logic*, Forthcoming.

Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Routledge, New York, 2001.