The Semantics of Negation

Kenneth Harris

Introduction

This is a paper about truth, or perhaps closer to what Gupta and Belnap call T-predicates [RTT,p.50], although the characterization provided below includes more than they would probably continence. The reason for the title will become apparent as the paper progresses, but in short I am holding everything else fixed: the language, its syntactic resources, the truth predicate; something has to give, and all that is left is the treatment of negation. I am not claiming the problem of truth reduces to a problem of finding an adequate treatment of negation, the web of implications in a language is much too intricate to pinpoint a single source, and besides there is much to be gained by fixing negation and varying other components of the language. There is a tendancy in the literature to fix a language, then add a truth predicate to it, and this would seem to implicate truth as the cause of all the problems. But, the concept of truth is part of a large family of concepts which cannot be given classical interpretations, all of which can be accomodated by changing the interpretation of negation. This gives some motivation for investigating the semantics of negation.

The investigations of this paper are heavily influenced by the paradigm of Logic Programming. What is important is not so much the paradigm of logic programming, but a core of underlying assumptions required in applying it. Much work on truth actually fits rather comfortably within this setting. There are benefits to presenting theories of truth in a general framework which includes the logic programming environment

- Allows exploitation of work on the semantics for logic programming setting as well as contributing to new semantics for logic programming. In particular, it opens up unexplored possibilities in the semantics of truth.
- Provides a general setting for definitions, whether explicit or circular.
- Provides a uniform setting for discussing the semantics of truth which
    - formalizes the basic assumptions
    - provides a common environment for investigating and comparing alternative semantics
    - allows more precise diagnosis of the problems of defining truth

The Languages

The language for which we will present a definition of truth is the object language, *OL*. This will be a formalized and interpreted first-order language, so that an implication relation between sets of sentences is well-circumscribed. Two points will be important in what follows

- (**FL-I**) Truth-in-*OL* is a property of sentences of *OL*.
- (**FL-II**) The meaning of any sentence is completely determined by the implication relations that hold involving it. The meaning of other expressions are derivative upon the meaning of the sentences in which they occur.

The language in which we will be presenting a definition of truth is the meta-language *ML*. This language will include a logical vocabulary of $\{\forall,\wedge,\neg,=\}$ and a nonlogical vocabulary including a one-place predicate symbol **T**. The meta-language will also be formalized and interpreted so that the above two conditions apply to it as well. An informal meta-meta-language (encompassing at least set theory and fully classical) will be employed, and it will be in this language where the interpretation of the predicate **T** of *ML* will be investigated. There will be cases where *OL* is *ML*, in which case I will refer to the meta-language as the language in which the semantics for true-in-*OL* will be investigated. In the meta-meta-language I will use lower case greek letters for variables ranging over names (α β γ) in *OL* and as place holders to denote a position in an expression that a name would go (ξ). I will use upper case greek letters for predicates in *OL* together with a place holder ($\Pi(\xi)$) and for sentences of *OL* (Φ). Several assumptions will be made about *ML*:

- (**ML-I**) There is a mapping from *tr* from *OL* to *ML* (except for sentences containing the predicate **T**) which preserves meaning (see (**FL-II**).) The goal will be to extend the translation to include all sentences.
- (**ML-II**) For every first-order sentence Φ of *OL* there is a name ϕ of Φ (I will use the lower case greek letter to name the sentence given by the upper-case letter, and which satisfies this condition.)

If, in addition *ML* is *OL* I will assume the names for sentences is rich enough to allow diagonalization

- (**ML-III**) For every predicate $\Pi(\xi)$ in *OL* there is a sentence Φ in *OL* and name α of $\Pi(\phi)$ such that $\alpha = \phi$ is assertible in *OL*

Classical Account of Truth

The goal is to define truth for *OL* in *ML*, so we need to provide necessary and sufficient conditions for the predicate symbol **T** of *ML* to be provided an interpretation in which it is a truth predicate for *OL*. The key to making some progress here will be conditions (**FL-I**) and (**FL-II**) on *OL*. It is also important to carefully circumscribe two tasks:

a. Describe a truth predicate for *OL* in *ML*
b. Give a systematic account of the semantics for sentences of *ML* which include the predicate **T**, and evaluate the adequacy of **T** for the tasks we wish to apply it.

We first must provide some conditions on when a predicate of *ML* is a truth predicate for *OL*. We will often be interested in capturing more than simply an extensionally adequate definition of truth, rather a definition that supports a vast web of inferences about the world from knowing the truth of a sentence. This is part of the analysis of the semantics carried out in the meta-meta-language for *ML*.

I follow [TARSKI, p.155 (the semantical definition)], [Kripke, 715], [RTT, p. 1 (Aristotle's Rule)] in taking the following two principles to be necessary conditions on a truth predicate for a language, in some other language (possibly the same language)

> (**TRUTH**) A predicate **T** in a language *ML* is a *truth predicate* for a language *OL* if there is a translation *tr* from *OL* to *ML* (see (**ML-I**)) such that the following holds: for every sentence Φ of *OL* the name ϕ of Φ in *ML* (given by (**ML-I**)) is such that

- (**T1**) $\mathbf{T}(\phi)$ may be inferred from $tr(\Phi)$
- (**T2**) The only grounds for inferring $\mathbf{T}(\phi)$ is through $tr(\Phi)$

Without further explanation these two principles do not *fix* the meaning of truth since they leave open the nature of the implication relation. This is the *decisive issue* for giving an account of truth for a language (whether this account is to be given in the language itself, or in some other language). Once an adequate account for implication is given, the fully spelled-out conditions (**T1**) and (**T2**) provide sufficient conditions for a predicate to be a truth predicate. This I take to be the *fundemental intuition concerning the meaning truth*.

If we judge from our meta-meta-language perspective that the predicate **T** of *ML* is inadequate for capturing uses of true-in-*OL* to which we wish to apply it, then we should judge that the inadequacy lies in the language *ML* and the adequacy of the implication relation of this language. We may wish to reserve the term true-in-*OL* only for predicates in languages that are fully adequate for every use which the term is to be applied and such a predicate would satisfy (**TRUTH**). In this case we may consider languages which satisfy (**TRUTH**) those which have a possible-true-in-*OL* predicate, since they clear at least one (important and distinctive) hurdle for having a true-in-*OL* predicate.

It will be useful to have a suggestive notation for the clauses of (**TRUTH**). A good choice might be '$\Leftrightarrow$', for coimplication, but a more suggestive choice is a notation which is assymetric, '$\leftarrow$'. (**TRUTH**) can be restated as a set of clauses for each sentence Φ of *OL* with name ϕ in *ML* such that

$$\mathbf{T}(\phi) \leftarrow tr(\Phi)$$

It is not required that there be any sentence of *ML* which corresponds to the coimplication, and in most of the cases considered in this paper, there will be no such connective. Each of the coimplications of (**TRUTH**) is a *partial* definition of the predicate **T**, and their totality give the full definition of the predicate **T**. I will call the clauses given by the classical definition **T**-clauses. Notice, that for each **T**-clause there stands at the head of the arrow an atomic expression of *ML* with **T** and at the tail of the arrow a complex sentence of *ML*. Since each atomic expression $\mathbf{T}(\alpha)$ occurs at the head of the arrow in at most one clause, we call the complex expression at the tail of the arrow the *grounds* for $\mathbf{T}(\alpha)$.

We can be more specific about what still needs to be decided about the implication relations. What is the status of atomic expressions $\mathbf{T}(\alpha)$ for which no ground is given? There are a couple of possibilities here:

- α is not the name of a sentence.
- α is the name of a sentence, and there is a β such that a ground is given for $\mathbf{T}(\beta)$ and '$\alpha = \beta$' is assertible

What is the status of complex expressions built-up from atomic **T** expressions? We are often interested in how truth is related to other concepts expressible in *ML*. For example, if we have a relation in *ML* which expresses consequence in *OL*, we may be interested in whether this relation preserves truth. But there is another reason that we need to account for complex expressions involving truth, it is possible that the predicate **T** also occurs in the complex ground of some clause in the definition. If *ML* is *OL* and *tr* is the identity translation, then given

assumption (**ML-III**) we will have a name in the language λ which names the sentence $\neg\mathbf{T}(\lambda)$. Thus, it is possible that one of the **T**-clauses will be

$$\mathbf{T}(\lambda) \leftarrow \neg\mathbf{T}(\lambda)$$

The presence of assumption (**ML-III**) opens up the possibility that very complex interrelations between various atomic **T**-expressions may exist and which show-up in **T**-clauses.

It is the presence of the **T**-clauses of (**TRUTH**) with an well-circumscribed implication relation that constitutes necessary and sufficient conditions for a predicate to be a truth predicate. The conditions of adequacy for the implication relation include handling ungrounded names (those which do not occur at the head of some **T**-clause) and providing for complex expressions involving the **T** predicate. I will now discuss two classic accounts of truth, those of Tarski and of Kripke, within the framework put down here, before proposing a third account. But first, it will be helpful to consider an incoherent account.

An Incoherent Account of Truth

Let *OL* be identical to *ML* and in which all the classical inferences hold and in which the translation function *tr* of (**ML-I**) is the identity function. Suppose *OL* also includes the following implications

For each sentence Φ of *OL* and name α of Φ
$\mathbf{T}(\alpha) \Leftrightarrow \Phi$

In the presence of (**ML-II**) and (**ML-III**) this language is inconsistent, every sentence is implied by any other sentence. The predicate **T** is both true of and false of every sentence, in the sense that both $\mathbf{T}(\alpha)$ and $\neg\mathbf{T}(\alpha)$ will be valid.

We can provide a semantics for the language *OL* in a classical metalanguage. In the meta-language we can define two sets $\mathbf{TRUE}_{OL}$ and $\mathbf{FALSE}_{OL}$ so that

'$\mathbf{T}$(x)' $\in \mathbf{TRUE}_{OL} \leftrightarrow$ x $\in \mathbf{TRUE}_{OL}$, for all x in the domain of *OL*

'$\neg\mathbf{T}$(x)' $\in \mathbf{TRUE}_{OL} \leftrightarrow$ x $\in \mathbf{FALSE}_{OL}$, for all x in the domain of *OL*

In this case the two sets, $\mathbf{TRUE}_{OL}$ and $\mathbf{FALSE}_{OL}$, are exactly the same sets. This is as expected as *OL* fails to make any distinctions of any kind. From the perspective of the meta-language, negation in *OL* means something completely different from negation in the meta-language: negation is an identity operator, Φ and $\neg\Phi$ have the same truth-value. It is incoherent to translate negation in *OL* as negation in the meta-language

$$tr(\neg\Phi) = \neg tr(\Phi)$$

for then either the meta-language is inconsistent or the sentence

'$\mathbf{T}$(x)' $\in \mathbf{TRUE}_{OL} \leftrightarrow$ x $\in \mathbf{TRUE}_{OL}$, for all x in the domain of *OL*

is false. I will construct the argument here shortly, when presenting Kripke's account of truth.

Tarski's Account of Truth

Let *OL* and *ML* be fully classical languages and distinguished by at least the presence of **T** in *ML* and not in *OL*. In this case we can use the material biconditional in *ML* to give a precise formulation of the requirements of the classical definition of truth through Tarski's Convention T [Tarski, §3]

> A formally correct definition of the predicate **T** in *ML*will be called an *adequate definition of truth for OL* if it has the following consequences
>
> 1. the truths of *OL* are sentences of *OL*
> 2. For every sentence Φ of *OL* and some name of Φ in *ML*, ϕ, the biconditional
>
> $$\mathbf{T}(\phi) \leftrightarrow tr(\Phi)$$

The adequacy of the above definition depends on the fact that *ML* is classical, although as we'll see in the case of Kripke's three-valued logic, it does not depend upon *OL* being classical. Tarski showed that one can consistently define truth for *OL* in *ML* satisfying Convention T.

There are several points I would like to highlight about Tarski's account. First, it is clear that Convention T is sufficient to ensure that **T** is a truth predicate for *OL* in *ML*. Second, we can verify in the meta-meta-language that the predicate **T** is indeed a truth predicate for *OL*. In the meta-meta-language we can define a set $\mathbf{TRUE}_{OL}$ which holds for all and only sentences of *OL*; similarly we can define a predicate $\mathbf{TRUE}_{ML}$ which holds of all and only sentences of *ML*. We can then prove, in the meta-meta-language

$'\mathbf{T}(x)' \in \mathbf{TRUE}_{ML} \leftrightarrow x \in \mathbf{TRUE}_{OL}$, for all x in the domain of *ML*

Since the metalanguage is fully classical, we will also have

$'\neg\mathbf{T}(x)' \in \mathbf{TRUE}_{ML} \leftrightarrow x \notin \mathbf{TRUE}_{OL}$, for all x in the domain of *ML*

and thus the truth predicates for *ML* and the meta-meta-language both partition the sentences of *OL* into two classes, true sentences and false sentences.

Third, there are many uses we might wish for a truth predicate, and we may evaluate how satisfactory *ML* fairs within the meta-meta-language. Tarski's own interest were originally motivated for meta-mathematical reasons. His work on truth started with his investigation of definable sets of real numbers in the first-order theory of the real numbers. His work on truth was applied to other metamathematical investigations such as consistency, consequence and categoricity. It is not sufficient to have a truth predicate in *ML* for *OL* to carry out metamathematical investigations. This was made clear in the early fifties by examples from Mostowski and Wang of languages strong enough to define a truth predicate satisfying Convention T, but too weak for proving interesting metamathematical results. Tarski did not see their examples as casting doubt on the adequacy of Convention T, but as a failure of the underlying logic (the implication relation) for those systems. (see [Tarski, §4, p. 237, footnote †]) The only way we have for recognizing a predicate, as a truth predicate, is through its connection with sentences, as given by by Convention T for a classical system or more generally through co-implication. Having given a truth predicate for a language *OL* in a meta-language *ML* we can then evaluate in the meta-meta-language the usefulness of this predicate for whatever ends we might wish to apply it. A predicate which fails to live up to these ends, does not fail to be a truth predicate, it fails to be useful in the ways we would like a truth predicate.

Kripke's Account of Truth

Let the only logical vocabulary of *OL* be $\{\forall,\wedge,\neg,=\}$. We would like to extend the interpretation of *OL* to

include a truth predicate for itself, and so letting *ML* be *OL*. [KRIPKE] showed *OL* can contain a truth predicate for itself if we equip it with a strong Kleene three-valued consequence relation, $\Rightarrow$, and consider fixed-point models. Let the translation function *tr* be the identity function. Then in any fixed-point model $\mathbf{T}(\alpha)$ and Φ will have the same truth-value (where α is any name for Φ). In this case, we will have

$$\mathbf{T}(\alpha) \Leftrightarrow \Phi$$

where α is a name in *OL* for the sentence Φ

From (**TRUTH**) **T** is a truth predicate. We may also move up to a meta-language for *OL* to investigate the semantics of the truth predicate in a classical meta-language (again, which includes set theory.) In this case, there will be a three-way partition of the sentences into $\mathbf{TRUE}_{OL}$, $\mathbf{FALSE}_{OL}$ and $\mathbf{NEITHER}_{OL}$. We can again verify **T** is indeed a truth predicate for *OL* in the meta-language

$$\text{'}\mathbf{T}(x)\text{'} \in \mathbf{TRUE}_{OL} \leftrightarrow x \in \mathbf{TRUE}_{OL}, \text{ for all x in the domain of } OL$$

since $\mathbf{T}(\alpha)$ has the same truth value as Φ (where α is any name for Φ). Thus, we can translate the *OL* predicate **T** as $\mathbf{TRUE}_{OL}$.

We may have a puzzle at this point. The metalanguage predicate $\mathbf{TRUE}_{OL}$ is the predicate we obtain when we *close off* the *OL* predicate **T**, where not-$\mathbf{TRUE}_{OL}$ corresponds to $\mathbf{FALSE}_{OL}$ or $\mathbf{NEITHER}_{OL}$, but there is no such predicate in *OL*. As Kripke points out, $\mathbf{TRUE}_{OL}$ and **T** do not even have the same extension: the extension of $\mathbf{TRUE}_{OL}$ is a set, while that of **T** is a pair of sets given by its extension $\mathbf{TRUE}_{OL}$ and anti-extension $\mathbf{FALSE}_{OL}$ (see [Kripke, p. 715].) Phillip Kremer makes the same point [PKREMER, §3], and goes even further, $\mathbf{TRUE}_{OL}$ does not express truth-in-*OL* (Kripke's intuition is that $\mathbf{TRUE}_{OL}$ is actually the *genuine* [emphasis of Kripke] truth predicate and not **T**. It should be noted that Kripke is not denying that **T** is a truth predicate, only that we would probably prefer the classical predicate $\mathbf{TRUE}_{OL}$ over the three-valued **T**. Still, it is a problem if $\mathbf{TRUE}_{OL}$ should turn-out not to even be a truth predicate. I will have more to say about Kripke's remark shortly.)

Phillip Kremer denies $\mathbf{TRUE}_{OL}$ is a truth predicate on the grounds that not all the T-biconditionals given in Convention T can be true for $\mathbf{TRUE}_{OL}$. Since the meta-language is fully classical, if this is true, then $\mathbf{TRUE}_{OL}$ cannot be a truth predicate for *OL*. Suppose we let *tr*(**T**(x)) be translated as $x \in \mathbf{TRUE}_{OL}$. Let λ be the name of the liar sentence $\text{'}\neg\mathbf{T}(\lambda)\text{'}$ of *ML*. Let Λ be the name of this same sentence in the meta-language. Then

1. $\Lambda \in \mathbf{TRUE}_{OL} \leftrightarrow tr(\neg\mathbf{T}(\lambda))$
2. $tr(\neg\mathbf{T}(\lambda)) \leftrightarrow \Lambda \notin \mathbf{TRUE}_{OL}$
3. $\Lambda \in \mathbf{TRUE}_{OL} \leftrightarrow \Lambda \notin \mathbf{TRUE}_{OL}$

Kremer concludes that $\mathbf{TRUE}_{OL}$, nor indeed any predicate of the meta-language, could be the translation of **T**. The mistake in the argument rests in the second line. The predicates $\mathbf{TRUE}_{OL}$ of the meta-language and **T** of *OL* are both *true* of the same sentences of *OL*, but they do not agree on the sentences they are false of. Thus,

$$tr(\neg \mathbf{T}(\lambda)) \leftrightarrow \wedge \notin \mathbf{TRUE}_{OL}$$

is false: we cannot translate negation in *OL* as negation in the meta-language. In fact, the correct replacement of the second line of the argument is

$$tr(\neg \mathbf{T}(\lambda)) \leftrightarrow \wedge \in \mathbf{FALSE}_{OL}$$

and thus concluding

$$\wedge \in \mathbf{TRUE}_{OL} \leftrightarrow \wedge \in \mathbf{FALSE}_{OL}$$

which is equivalent to $\wedge \in \mathbf{NEITHER}_{OL}$. We actually want to translate negation of *OL* as $\mathbf{FALSE}_{OL}$ of the meta-language and **T** as $\mathbf{TRUE}_{OL}$, then we can prove all the T-biconditionals in the meta-language. As in the incoherent account, the truth predicate of *OL* makes perfectly good sense in the meta-language.

There is a separate question of the adequacy of the predicate **T**, and the way the language carves-up sentences: true, false and neither. The language has no resources to express a fully classical negation, and may (on these grounds) fail to adequately express true-in-*OL* (this seems to be behind Kripke's intuition above.) Can we make sense of these distinctions in a way to accords with how we want to use the concept true-in-*OL*. This is the essential question addressed in [PKREMER, §6]. For the smallest fixed point model [KRIPKE] tells a compelling story of how a speaker of a language *OL* unfamiliar with the word true (-in-*OL*), in the sense of being entirely uncommitted whether to affirm or deny any sentence involving this predicate, might yet come to grasp the extension of this predicate from only a knowledge of the language without the predicate, together with the clauses of (**TRUTH**). At least the story may strike one as compelling if one holds that the semantics of truth should be fully determined by the non-semantic facts (see [RTT, p. 18], [MT],[MKREMER]). Leaving aside whether we ought to ascribe to this position, the story is uncompelling for the reason that a speaker of a language would not come to learn a new predicate while eschewing how the sentences formed from this new predicate cohere with the rest of the language. In particular, Kripke's story does not take negation seriously.

Taking Negation Seriously

I have taken the logical vocabulary of *OL* to be very simple, $\{\forall, \wedge, \neg, =\}$. From Tarksi's indefinability of truth, no consistent classical language satisfying the constraints of (**ML-I**), (**ML-II**) and (**ML-III**) can contain its own truth predicate. In *OL* it is solely the presence of negation that causes the problem of defining truth, without negation it will be possible to consistently define a truth predicate for a classical language in itself. In the face of languages which contain their own truth-predicate like the Kripke three-valued language or the incoherent language, the challenge is to provide an interpretation of negation that is intelligible and useful given the purposes we wish to use the language.

Let's go back to Kripke's intuition that the meta-language predicate $\mathbf{TRUE}_{OL}$ for the three-valued language *OL* is the "genuine" truth predicate. I think the right way to look at matters is that the issue is not directly over the concept of truth, but over the treatment of negation (This is *not* to say that truth and negation

are separate matters, they are intimately related, as evidenced by the fact that a consistent language cannot have a classical negation and a truth predicate. In this paper, as does [KRIPKE] as well as many other writers on truth, we are fixing assumptions so that the only possible way to have a truth predicate in the language is to change the treatment of negation.) The meta-language has access to a classical negation unavailable to *OL* which allows it close-off the *OL* predicate **T**. We cannot consistently introduce a classical negation in *OL* and still insist on a truth predicate over the entire language. Can capture at least part of the intuition that negation should correspond to closing off the truth predicate? (This is a move we might take our native speaker as trying to make, given their understanding of negation before encountering a truth concept.)

Kripke's least fixed point construction yields a very weak negation, one constructed from below:

> Let α name Φ. Then $\neg\mathbf{T}(\alpha)$ is true if and only if Φ is determined to be false at some previous stage.

Our minimal stipulation on a truth predicate leaves open how we wish to treat negated truths, just as it does the conjunction of truths or universally quantification over truths. We would not expect to readily change the way we treat conjunctions or quantifications without compelling grounds for change. We might also be expected to minimize the extent of the change. My proposal is to consider how we might take seriously the proposal that the negation of a sentence is the failure of truth for that sentence, as given by the clauses of (**TRUTH**). We cannot capture this directly with a connective, but perhaps we can approach matters another way.

Negation as Failure

The proposal I make here has long been apart of the logic programming community to provide a semantics for logic programs. The particular semantic treatment I propose below borrows heavily from [Fitting], [GL] and [FINE]. I will make this clearer later. The idea is to treat the arrow '$\leftarrow$' from the truth clauses as a conditional and to add rules for deriving truths and negated truths from the clauses. A more appropriate logic for investigating this possible treatment of negation is Belnap's four-valued logic, allowing for truth value gaps and gluts as well as the two standard truth values, true and false (see [AVRON].)

Let *OL* be an interpreted first-order classical language satisfying the clause (**FL**) and (**ML**), and also have a name for every object in the domain of its interpretation. It will be convenient to put the clauses of our truth definition in a canonical form. The current set of clauses defining **T** will be replaced allowing the same head formula to occur in multiple clauses and canonicalizing each clause so that it has the form:

> $\mathbf{T}(\alpha) \leftarrow \Gamma$
> where Γ is (a possibly infinite) set of literals: atomic sentences or negations of atomic sentences.

The sets constituting the body of the clauses will be treated as conjunctions of the literals that occur in them; a pair of clauses with the same head will be treated as disjunctions of their bodies. The transformation replaces universal quantifiers with (possibly infinite) conjunctions of its instances, existential quantifiers with (possibly infinite) disjunctions of its instances. These new (possibly infinite) sentences are placed into disjunctive normal form. Finally, each disjunct will give rise to a new clause with the same head, and whose body is the set of literals which make up the conjunction. The underlying logic of *OL* must satisfy some basic equivalences to justify the transformation

- $\neg\neg\Phi$ and Φ
- $\neg\forall x\Phi$ and $\exists x\neg\Phi$
- $\neg(\Phi \wedge \Psi)$ and $\neg\Phi \vee \neg\Psi$

which the four-valued logic does.

For simplicity, I will assume that if $\alpha = \beta$ is a true identity then in every clause in which α occurs there is a clause in which β occurs in exactly the positions which α occupied. To this new set of clauses, we'll add clauses which capture the true literals of *OL* in the language of *OL* minus **T**. For each true literal Φ, we'll have a clause, $\Phi \leftarrow \{\}$. These clauses will constitute the *axioms*. There will be two rules of inference. The first is a rule that combines adjunction and modus ponens

(**MPA**) If each $\Psi \in \Gamma$ is derivable, and $\Phi \leftarrow \Gamma$ is an axiom, then infer Ψ.

The second rule captures the idea that negation is the failure of deriving a formula from the axioms and rules of inference

(**NEG**) If Ψ is not derivable then infer $\neg\Psi$.

Both rules (**MPA**) and (**NEG**) are self-referential, but while (**MPA**) can be expressed as a monotonic operator on sentences, (**NEG**) cannot and so we cannot expect that these rules will give rise to a consistent and complete set of derivable formulas. In fact they do not, with the liar sentence:

$\mathbf{T}(\lambda) \leftarrow \neg\mathbf{T}(\lambda)$

the rules give rise to contradictory directives.

Instead of fully satisfying the rules we can try to approximate them. A natural way to try to do this is to borrow an idea from [GL]. Let *A* be the set of clauses from (**TRUTH**) in canonical form. We start with a guess of those instances of $\mathbf{T}(\alpha)$ which are true (the set *T*) and those instances of $\neg\mathbf{T}$(alpha;) which are true (the set *NT*), where all sentences of *OL* which do not include the predicate **T** are on one list or the other (as appropriate given the interpretation.) Using these guesses we modify the clauses which make-up the axioms to get a new set of clauses (*A′*). On the basis of this new set of clauses we can update our initial guesses of *T* and *NT*. We then modify the set of clauses (*A′*) in light of the new guesses. This continues until the clauses no longer change, a fixed-point. At this stage we will have sets of the true instances of $\mathbf{T}(\alpha)$ and $\neg\mathbf{T}(\alpha)$. The final step will be to use these lists and the clauses remaining to assign truth values.

I will define an operator Ξ which will have arguments: *A* a set of clauses, *T* and *NT* sets of sentences, and which will produce a set of clauses *A′* and sets of sentences *T′* and *NT′*.

Let *A* be a set of clauses, *T* and *NT* sets of sentences. The rule (**STEP**) for the operator Ξ on this triple returns a triple $\langle A', T', NT' \rangle$ where

- *A′* is a set of clauses obtained from *A* by the following transformation
  a. For each clause $\Phi \leftarrow \Gamma$, let $\Gamma' = \Gamma - T$. Replace this clause with $\Phi \leftarrow \Gamma'$

b. Remove any clause $\Phi \leftarrow \Gamma$ where $\Gamma \cap NT$ is non-empty. (This transformation is performed only after (a).)

- Let $T' = T \cup \{\Phi : \Phi \leftarrow \varnothing \text{ is a clause of } A'\}$
- Let $NT' = NT \cup \{\Phi : \Phi \text{ is not at the head of any clauses of } A'\}$

A *Kripke fixed point* of the operator Ξ is a triple $\langle A,T,NT \rangle$ such that applying (**STEP**) returns $\langle A,T,NT \rangle$.

The motivation for Ξ is that if we have determined a sentence Φ to be true we can eliminate it from the body of any clause; and if we have determined that the negation of Φ is true, we can eliminate any clause which has Φ in its body (since the body cannot be satisfied.) We'll use the fix points of Ξ to determine the truth assignments. I'll first define the Kripke valuation, which is straightforward:

Given sets of sentences T and NT a *valuation* assigns truth values to as follows

- Φ is true if $\Phi \in T$ and $\Phi \notin NT$
- Φ is false if $\Phi \notin T$ and $\Phi \in NT$
- Φ is indeterminate if $\Phi \notin T$ and $\Phi \notin NT$
- Φ is contradictory if $\Phi \in T$ and $\Phi \in NT$

A *Kripke valuation* is a valuation from a (**STEP**) fixed point.

There is a smallest and largest Kripke fixed point (both sets T and NT monotonically increase, while the sentences contained in all clauses in A monotonically decrease, under the application of each rule.) The smallest Kripke fixed point is obtained by starting with T and NT both empty, and the largest Kripke fixed point is obtained by starting with every sentence in T and NT. The smallest Kripke fixed point is three valued and coincides with Kripke's own construction of a smallest three-valued fixed point using the Kleene three-valued logic. (This logic agrees with Belnap's four-valued logic on the connectives true, false and neither.)

Now, let's think about the current fixed point from the perspective of treating negation as failure of derivation. Suppose $\mathbf{T}(\alpha) \notin T$. Should we allow that we have now failed to derive $\mathbf{T}(\alpha)$? If we add $\mathbf{T}(\alpha)$ to NT, where $\mathbf{T}(\alpha) \notin NT$, we may admit new derivable formulas, perhaps even $\mathbf{T}(\alpha)$ itself. We want to avoid this. Consider the truth teller, where τ names $\mathbf{T}(\tau)$

$\mathbf{T}(\tau) \leftarrow \mathbf{T}(\tau)$

Suppose at this stage, this clause will still be among the axioms in our fixed point so that $\mathbf{T}(\tau) \notin T$. We can safely add $\mathbf{T}(\tau)$ to NT without worrying about it being in the future derivable. We can generalize this by tracking all possible derivation paths to $\mathbf{T}(\alpha)$ and checking that every sentence on any path is a positive literal (an atomic sentence.)

Let $\langle A,T,NT \rangle$ be a (**STEP**) fixed point. The *dependencies* of $\mathbf{T}(\alpha)$, *depend*($\mathbf{T}(\alpha)$), is the union of the sets

a. $\{\Phi : \mathbf{T}(\alpha) \leftarrow \Gamma \in A \text{ and } \Phi \in \Gamma\}$
b. *depend*(Φ) for $\mathbf{T}(\alpha) \leftarrow \Gamma \in A$ and $\Phi \in \Gamma$

Say that **T**(α) is *safe* if *depend*(**T**(α)) consists of only positive literals. Let the rule (**FAILURE**) determine the triple given by

- $A' = A$
- $T' = T$
- $NT' = NT \cup \{T(\alpha) : T(\alpha)$ is safe and $T(\alpha) \notin T\}$

for the (**STEP**) fixed point $\langle A, T, NT \rangle$. A (**FAILURE**) fixed point is a triple in which $NT' = NT$ after application of the (**FAILURE**) rule.

Since NT' may have expanded after an application of (**FAILURE**), we will need to continue to apply (**STEP**) until the next (**STEP**) fixed point, then re-apply (**FAILURE**).

A *negation stable fixed point* is a triple which is a fixed point of Ξ starting with T empty.
A *negation stable valuation* is a valuation from a negation stable fixed point.

Given any starting point $\langle A, T, NT \rangle$ there must be a negation stable fixed point (both sets T and NT monotonically increase, while the sentences contained in all clauses in A monotonically decrease, under the application of each rule.) There is smallest such fixed point, starting from empty NT and a largest fixed point, starting with all **T** sentences in NT. The smallest fixed point is three-valued.

Comparisons

The smallest Kripke fixed point and smallest negation stable fixed point are different: the truth teller is indeterminate in the Kripke fixed point and false in the negation stable fixed point. The liar is indeterminate in both. In fact, every negation stable fixed point is a Kripke fixed point, by choosing the right starting position. The smallest negation stable fixed point is not the largest consistent negation stable fixed point. Both of the following sentences will come out indeterminate

$$\mathbf{T}(\alpha) \rightarrow \neg \mathbf{T}(\beta)$$
$$\mathbf{T}(\beta) \rightarrow \neg \mathbf{T}(\alpha)$$

yet it is consistent to make **T**(β) false and **T**(α) true by starting with $NT = \{\mathbf{T}(\beta)\}$ (the reverse characterization is possible.) Like the smallest Kripke fixed point, the smallest negation stable fixed point is fully determinate given the nonsemantic facts, and takes the pre-truth role of negation in the language more seriously. Still, negation as failure (as given by the rules of Ξ) is not full negation, nor will it satisfy tautologous appearing sentences like **T**(λ) $\vee \neg$**T**(λ), where **T**(λ) is the liar.

The construction presented in the last section draws from [FINE], [GL] and [FITTING]. From [FINE] I took the characterization of the rule (**NEG**) and (**MPA**), from [GL] the rule (**STEP**), although both authors were strictly interested in only classical models generated from a set of clauses (so, in particular, were not concerned with the case of truth.) From [FITTING] I took the idea of making a guess of those instances of **T**(α) which are true and those instances of ¬**T**(alpha;) which are true. The negation stable fixed points are a distinct class of fixed points from those considered by [FITTING], the GLF stable fixed points. In particular, every GLF stable fixed point will assign the pair of sentences above both indeterminate or both contradictory values. I believe

every GLF stable fixed point will be negation stable, but I have not convinced myself of this. There is also a least GLF stable fixed point, and I suspect it is the same as the least negation stable fixed point, but I am not sure.

Fitting has presented a rich and elegant algebraic characterization, through GLF stable fixed points, but I felt that the construction is a bit removed from the intuitions of [FINE] and [GL] that have more substantive philosophical motivation. (Fitting views the GLF stable fixed points correctly treat the case of the pair given above--they don't make arbitrary judgements. While this is true, one would like a construction that doesn't rule such choices out of hand, they are still consistent with the negation as failure motivation. The problem with Fitting's construction is that he builds *T* and *NT* separately from each other, which produces an elegant algebraic characterization, but I think is less motivated by the grounding intuitions then my construction above.) It had been my intent to give a more compact algebraic characterization of negation stable fixed points. I have a pretty good idea how to accomplish this (modifying Fitting's GLF stable fixed point characterization), but there are still details that need work. Such a characterization would allow a more direct comparison with Fitting's own construction.

Conclusions

The presentation here has fixed a number of features of language:

- The conditions of adequacy for truth (**TRUTH**)
- The resources available for talking about syntax (**ML-II**) and (**ML-III**)
- The logical connectives available $\{\forall,\wedge,\neg,=\}$
- The translation function when *OL* was *ML*

Given this, there was only one direction to modify the language to allow for a language to have its own truth predicate, modify the treatment of negation. This is not intended to implicate negation, but to give a more careful consideration of options for accomodating a truth predicate. The account has the added benefit that it can be extended to a broad range of troublesome predicates (provided they can be given implicit definitions through clauses as was done in (**TRUTH**).)

Bibliography

[AVRON] Arnon Avron, "Classical Gentzen-Type Methods in Propositional Many-Valued Logics" in Beyond Two: Theory and Applications of Many-Valued Logics (M.Fitting and E. Orlowska, eds.), *Studies in Fuzziness and Soft Computing*, Physica Verlag, 2003, Vol. 114, pp. 117-155.

[FINE] Kit Fine, "The Justification of Negation as Failure", in Logic, Methodology and Philosophy of Science vol. VIII (J.E. Fenstad, I.T.Frolov, R. Hilpinen, eds.), pp. 263-301.

[FITTING] Melvin Fitting, "A Theory of Truth that Prefers Falsehood" in *Journal of Philosophical Logic* **26**, pp. 477-500.

[GL] M. Gelfond and V. Lifschitz, "The Stable Model Semantics for Logic Programming" in Logic Programming: Proceeedings of the Fifth International Conference and Symposium (R. Kowalski and K. Bowen, eds.), pp. 1010-1080.

[GUPTA] Anil Gupta, "Truth and Paradox", in *Journal of Philosophical Logic* **11**, pp. 1-60.

[KRIPKE] Saul Kripke, "Outline of a Theory of Truth" in *Journal of Philosophy* **72**, pp. 690-716.

[MKREMER] Michael Kremer, "Kripke and the Logic of Truth" in *Journal of Philosophical Logic* **17**, pp. 225-78.

[MT] Anil Gupta, "The Meaning Theory of Truth", in New Directions in Semantics (Ernest LePore, ed.), pp. 453-80.

[PKREMER] Phillip Kremer, "On the "Semantics" for Languages with their own Truth Predicate", in Truth, Definition and Circularity (A. Chapuis and A. Gupta, eds.), pp. 217-46.

[RTT] Anil Gupta and Nuel Belnap, The Revision Theory of Truth.

[TARSKI] Alfred Tarski, "The Concept of Truth in Formalized Languages" in Logic, Semantics and Metamathematics: Papers from 1923-38 (translated by J.H. Woodger), pp.152-278.