Paracomplete truth theory with a definable hierarchy of determinateness operators

Marcos Cramer

International Center for Computational Logic, TU Dresden, Germany

Abstract. One way to deal with the Liar paradox is the paracomplete approach to theories of truth that gives up proofs by contradiction and the Law of the Excluded Middle. This allows one to reject both the Liar sentence and its negation. The simplest paracomplete theory of truth is *KFS* due to Saul Kripke. At face value, this theory suffers from the problem that it cannot say anything about the Liar paradox, so a defender of this theory cannot explain their rejection of the Liar sentence within the language of *KFS*. This was one of the motivations for Hartry Field to extend *KFS* with a conditional that is not definable within *KFS*. With the help of this conditional, Field defines a determinateness operator that can be used to explain one's rejection of the Liar sentence within the object language of his theory. The determinateness operator can be transfinitely iterated to create stronger notions of determinateness required to explain the rejection of paradoxical sentences involving the determinateness operator. In this paper, we show that Field's complex extension of *KFS* is not required in order to express rejection of paradoxical sentences like the Liar sentence. Instead one can work with a transfinite hierarchy of determinateness operators that are definable in *KFS*. After defining this hierarchy of determinateness operators, we compare their properties with the transfinitely iterable determinateness operator due to Field.

1 Introduction

In everyday conversations, in scientific texts and in philosophical discussions we often make use of the predicate "true". So to get a better understanding of how we communicate our ideas and how we judge the correctness of our arguments, it is desirable to have a reasonable *theory of truth*, i.e. a logical formalism that contains the predicate *True*, that captures the inferences involving this predicate that we would intuitively deem acceptable, and that satisfies further rationality criteria like consistency.

In any formalism that contains the predicate *True* and captures come basic arithmetical reasoning, one can construct a *Liar sentence*, i.e. a sentence that asserts of itself that it is not true. If we apply some classically valid inferences combined with some intuitive inferences for the predicate *True* to such a Liar sentence, we can derive an inconsistency, and thus by some further classically valid inferences, we can derive every sentence of the language, rendering the

formalism practically useless. This is called the *Liar paradox*. Any reasonable theory of truth needs to handle the Liar paradox in some way by putting some restrictions either on the intuitive inferences for the predicate *True* or on the rules of classical logic. Field [3] proposes a so-called *paracomplete* theory of truth that deals with the Liar paradox by restricting classical logic to the strong Kleene logic K_3, in which proof by contradictions are not unrestrictedly admissible and the Law of the Excluded Middle ($\varphi \vee \neg\varphi$ for any formula φ) does not hold unrestrictedly, but which differs from intuitionistic logic by still admitting double negation elimination and De Morgan's Laws.

This paracomplete approach can be traced back to the work of Kripke [4]. The formal theory of truth that is based on the semantic construction due to Kripke is usually called *KFS* and has K_3 as its underlying logic.

The stance towards the Liar paradox that a paracomplete theory of truth defends is that of rejecting both the Liar paradox and its negation. It seems desirable that this stance should be expressible and justifiable within the formal theory that the paracompletist puts forward. At face value, it seems that this cannot be achieved withing *KFS*. To overcome this limitation of *KFS*, Field [3] has introduced a determinateness operator, which allows one to say that the Liar paradox is not determinately true as a justification for rejecting the Liar sentence. The determinateness operator can be transfinitely iterated to create stronger notions of determinateness required to explain the rejection of paradoxical sentences involving the determinateness operator.

While Field's theory of truth is based on *KFS*, his determinateness operator is based on a conditional that is not definable within *KFS* but needs to be added to the theory. Field provides a complex semantic characterization of this conditional, but does not provide a proof-theoretic account of it.

In this paper we show that Field's complex extension of *KFS* is not required in order to express reasons for rejecting paradoxical sentences like the Liar sentence. Instead one can work with a transfinite hierarchy of determinateness operators that are definable in *KFS*. The definition of these determinateness operators is inspired by the well-founded semantics of logic programming [9,1]. We will define this hierarchy of determinateness operators, compare its properties to the transfinitely iterable determinateness operator due to Field, and briefly sketch how the construction of the determinateness operators can be modified to construct a conditional that satisfies some desirable properties.

2 The Liar Paradox

A Liar sentence is a sentence that asserts of itself that it is not true. An informal example of a Liar sentence is the following sentence that uses the determiner *this* to refer to itself:

This sentence is not true.

Given that the semantics of the word *this* depends a lot on the communicative context, logicians often prefer to work with more formal Liar sentences whose

interpretation is completely independent of the communicative context. For this purpose, one usually works in a formal language that extends the standard first-order language of arithmetic $\mathcal{L}^{\text{arithm}}$ with a truth predicate *True*, yielding the extended first-order language $\mathcal{L}^{\text{arithm}}_{True}$. It is well-known that assuming some basic formal theory of arithmetic, e.g. Peano Arithmetic, one can define a Gödel numbering of any given formal language, i.e. an encoding of the syntax of that language in terms of natural numbers which maps each formula φ of the formal language to a unique natural number $\langle\varphi\rangle$ that can be used to talk about φ in the formal language. Intuitively, the intended meaning of $True(n)$ is that there exists a sentence φ of $\mathcal{L}^{\text{arithm}}_{True}$ such that $\langle\varphi\rangle = n$ and φ is true.

It is well-known that using a diagonalization technique due to Carnap, Gödel and Tarski one can construct a sentence $L \in \mathcal{L}^{\text{arithm}}_{True}$ for which one can prove in Peano Arithmetic (and indeed in various weaker theories of arithmetic) that

$$L \leftrightarrow \neg True(\langle L\rangle) \tag{1}$$

Given our intuitive interpretation of *True*, the sentence L is thus provably equivalent to the statement that L is not true. In other words, L is a Liar sentence, and unlike the informal Liar sentence presented above, it is a purely formal Liar sentence that does not depend on the interpretation of a context-dependent word like *this*.

Once we have constructed L, it seems like we can derive both $\neg L$ and L using standard rules of inference. We start with a proof by contradiction that establishes $\neg L$: Assume for a contradiction that L holds. In that case L is true, i.e. $True(\langle L\rangle)$ holds. But from (1) we get $True(\langle L\rangle) \rightarrow \neg L$, so by modus ponens we get $\neg L$. This contradicts our assumption that L is true. This completes the proof by contradiction, i.e. we can retract the assumption and deduce $\neg L$. But from (1) we have $\neg L \rightarrow True(\langle L\rangle)$, so by modus ponens we get $True(\langle L\rangle)$, i.e. we get that L is true. So we have deduced both $\neg L$ and L, a contradiction.

This is what is commonly called the *Liar paradox*. If we try to formalize this apparent proof in the proof calculus of natural deduction, we see that apart from the explicitly stated rules *modus ponens* (also called ($\rightarrow$-Elim)) and *proof by contradiction* (also called ($\neg$-Introd)), we implicitly made use of two further rules of inference that involve the predicate *True*:

$$\text{(T-Introd)} \qquad \varphi \models True(\langle\varphi\rangle)$$
$$\text{(T-Elim)} \qquad True(\langle\varphi\rangle) \models \varphi$$

These two rules are very compelling, because they seem to precisely characterize our intuitions about the meaning of the predicate *True*. What the Liar paradox shows is that these rules cannot be consistently combined with classical logic. Multiple avenues have been explored to deal with this:

- Those who want to keep classical logic fully in place need to reject at least one of (T-Introd) and (T-Elim). The most well-known theory of truth that works with classical logic is the so-called Kripke-Feferman theory KF, which accepts (T-Elim) but rejects (T-Introd) [2]. However, this theory has the awkward property of declaring "L, but L is not true".

- One can bite the bullet and accept that both L and $\neg L$ can be derived, but restrict classical logic so that this inconsistency does not lead to explosion, i.e. to the derivability of all sentences. This approach is called the *paraconsistent* approach and was first proposed by Priest [5].
- One can restrict the structural rules of inference that were left implicit in the above piece of informal reasoning and that allow one to use already derived sentences as premises for further derivations, as well as to use an assumption more than once in a derivation [6].
- One can give up one of those rules of inference that were explicitly used in the above elicitation of the Liar paradox. The most common rule of inference to be dropped is proof by contradiction ($\neg$-Introd). Dropping this rule is called the *paracomplete* approach, and it has recently gained traction due to Field's [3] defense of it.

In this paper we are working with a paracomplete approach to the Liar paradox, i.e. we are giving up proof by contradiction in its unrestricted form, which allows us to accept (T-Introd) and (T-Elim) unrestrictedly.

3 From Kripke to Field: Paracomplete Approaches to Semantic Paradoxes

Kripke [4] defined a construction that can be used to give a three-valued model-theoretic semantics for the language $\mathcal{L}^{\text{arithm}}_{True}$. This construction gives rise to the paracomplete theory *KFS* and is also at the heart of the paracomplete theory of truth presented by Field [3]. The same construction can also be used as the basis for approaches based on classical logic, e.g. for the Kripke-Feferman theory KF. We will now sketch this construction and explain how it serves as a basis for a paracomplete theory of truth.

Following Field [3], we use $\{0, ½, 1\}$ as the names for the three truth-values, in order to avoid confusion between the object language predicate *True* and the truth-value 1 that was called *true* by Kripke [4]. We assume that $\mathcal{L}^{\text{arithm}}$ contains the falsity constant $\bot$, the negation symbol $\neg$, the conjunction symbol $\wedge$ and the universal quantifier $\forall$. We write $(\varphi \vee \varphi)$ for $\neg(\neg\varphi \wedge \neg\psi)$, $(\varphi \supset \varphi)$ for $\neg(\varphi \wedge \neg\psi)$, and $\exists x : \varphi$ for $\neg\forall x : \neg\varphi$. We sometimes drop brackets when this does not cause confusion. As usual, we assume that $\mathcal{L}^{\text{arithm}}$ contains the equality symbol $=$ as its only predicate symbol and that it contains a constant symbol 0, a unary function symbol *succ* (*successor*) and two binary function symbols $+$ and $\cdot$, conventionally written in infix notation (e.g. $(s(0)\cdot(s(0)+s(0)))$). A countably infinite supply of variable symbols $(x, y, z, x_0, x_1, \dots)$ is assumed to be given. As usual, the constant symbol 0 and the variable symbols can be combined with the function symbols to form *terms*.

A *variable assignment* s is a function that assigns a natural number to each variable. Given a variable assignment s, a variable x and a natural number n, $s[x : n]$ denotes the variable assignment that coincides with s on all variables

other than x and that assigns n to x. One can inductively define the interpretation t^s of a term t under a variable assignment s as follows:

$$\begin{aligned}
0^s &= \text{the natural number } 0\\
x^s &= \text{the number that } s \text{ assigns to the variable } x\\
succ(t)^s &= \text{the successor of the natural number } t^s\\
(t_1 + t_2)^s &= \text{the sum of the natural number } t_1^s \text{ and the natural number } t_2^s\\
(t_1 \cdot t_2)^s &= \text{the product of the natural number } t_1^s \text{ and the natural number } t_2^s
\end{aligned}$$

Kripke's construction is based on a transfinite recursion which starts with assigning the truth-value ½ to each formula and then recursively updates the truth-values of all formulas until a fixed point is reached after some transfinite number of iterations. At each step α of this transfinite recursion and for each variable assignment s, we assign to each formula $\varphi \in \mathcal{L}_{True}^{\text{arithm}}$ a truth-value $\varphi^{\alpha,s} \in \{0, ½, 1\}$. The transfinite recursion is defined as follows:

$$\begin{aligned}
\varphi^{0,s} &= ½ \text{ for every } \varphi \in \mathcal{L}_{True}^{\text{arithm}} \text{ and every variable assignment } s\\
(t_1 = t_2)^{\alpha+1,s} &= \begin{cases} 1 \text{ if } t_1^s = t_2^s \\ 0 \text{ otherwise} \end{cases}\\
(\bot)^{\alpha+1,s} &= 0\\
(\neg\varphi)^{\alpha+1,s} &= 1 - \varphi^{\alpha+1,s}\\
(\varphi \wedge \psi)^{\alpha+1,s} &= \min(\varphi^{\alpha+1,s}, \psi^{\alpha+1,s})\\
(\forall x : \varphi)^{\alpha+1,s} &= \min\{\varphi^{\alpha+1,s[x:n]} \mid n \in \mathbb{N}\}\\
(True(t))^{\alpha+1,s} &= \begin{cases} 1 & \text{if there is a } \varphi \in \mathcal{L}_{True}^{\text{arithm}} \text{ with } t^s = \langle\varphi\rangle \text{ and } \varphi^{\alpha,s} = 1 \\ ½ & \text{if there is a } \varphi \in \mathcal{L}_{True}^{\text{arithm}} \text{ with } t^s = \langle\varphi\rangle \text{ and } \varphi^{\alpha,s} = ½ \\ 0 & \text{otherwise} \end{cases}\\
\varphi^{\lambda,s} &= \begin{cases} ½ & \text{if } \lambda \text{ is a limit ordinal and } \varphi^{\alpha,s} = ½ \text{ for all } \alpha < \lambda \\ 1 & \text{if } \lambda \text{ is a limit ordinal and } \varphi^{\alpha,s} = 1 \text{ for some } \alpha < \lambda \\ 0 & \text{otherwise, if } \lambda \text{ is a limit ordinal.} \end{cases}
\end{aligned}$$

Clearly for sentences (formulas without free variables) the variable assignment has no impact on the assigned truth-value, so we can write φ^α instead of $\varphi^{\alpha,s}$ when φ is a sentence.

One can easily see that this transfinite recursion is monotonic, i.e. if $\varphi^{\alpha,s} \neq ½$ for some ordinal α, then $\varphi^{\beta,s} = \varphi^{\alpha,s}$ for all $\beta \geq \alpha$. This together with the fact that $\mathcal{L}_{True}^{\text{arithm}}$ is countable implies that a fixed point is reached at some countable ordinal α_0, i.e. for each formula $\varphi \in \mathcal{L}_{True}^{\text{arithm}}$, each ordinal $\alpha \geq \alpha_0$ and each variable assignment s, $\varphi^{\alpha,s} = \varphi^{\alpha_0,s}$. The (ultimate) truth-value of a sentence $\varphi \in \mathcal{L}_{True}^{\text{arithm}}$, denoted as $|\varphi|$, is defined to be φ^{α_0}.

The idea behind paracomplete theories of truth like that of Field [3] is that the only sentences of $\mathcal{L}_{True}^{\text{arithm}}$ that we should accept are the ones that get assigned truth-value 1 in Kripke's construction, while sentences with truth-value 0 or ½ should be rejected. The theory comprising all the sentences with truth-value 1 in Kripke's construction is usually called *KFS*.

A sentence φ for which $(\varphi \vee \neg\varphi)$ is accepted is called *bivalent*. All sentences that do not involve the predicate *True* are bivalent in *KFS*. Also any sentence in which *True* is only applied to Gödel codes of sentences not involving *True* is bivalent. This process can be continued to the point that can be informally characterized by saying that all sentences in which there are no infinite nestings of the predicate *True* are bivalent. Note that "n is the Gödel code of a bivalent formula" is itself not generally bivalent, so when we want to bivalently restrict ourselves to bivalent formulas, we need to use a syntactic criterion like "the formula does not contain the predicate *True*" (but this way we always miss out some bivalent formulas).

The three-valued logic that is underlying the theory *KFS* is usually called K_3. Tamminga [8] has defined a natural deduction calculus for K_3, which differs from the standard natural deduction calculus for classical logic only in two respects: While the $\neg$-introduction rule (*proof by contradiction*) is dropped, five rules are added to ensure that we can still perform a sufficient amount of reasoning with negation, one rule for double negation introduction ($\varphi \vDash \neg\neg\varphi$) and four rules that correspond to the two De Morgan's laws $\neg(\varphi \wedge \psi) \dashv\vDash (\neg\varphi \vee \neg\psi)$ and $\neg(\varphi \vee \psi) \dashv\vDash (\neg\varphi \wedge \neg\psi)$. So we can say that the paracomplete approach gives up proof by contradiction while replacing it by weaker reasoning principles.

Due to Gödel's second incompleteness theorem, it is not possible to give a complete recursive axiomatization of the set of sentences of $\mathcal{L}^{\text{arithm}}_{True}$ that get assigned truth-value 1 in Kripke's construction. But a natural axiomatization to use for deriving a considerable subset of these sentences is the one that we get by adding the rules $\varphi \models True(\langle\varphi\rangle)$ and $True(\langle\varphi\rangle) \models \varphi$ as well as the axioms of classical Peano Arithmetic and the axiom scheme "$(\varphi \vee \neg\varphi)$ is an axiom for $\varphi \in \mathcal{L}^{\text{arithm}}$" to the natural deduction calculus for K_3. We call the resulting proof system PA_{KFS}. We write $Pr_{KFS}(n)$ to denote the statement that n is the Gödel code of a formula that can be derived in PA_{KFS}. We say "PA_{KFS} is sound" for the statement $\forall n : (Pr_{KFS}(n) \supset True(n))$.

One can easily see that the Liar sentence L gets assigned truth-value ½ in Kripke's construction, so both L nor its negation $\neg L$ will be rejected in a paracomplete theory of truth.

Usually when we reject a certain statement, we can explain this rejection by explaining that we believe the negation of the statement in question, and maybe additionally give reasons for this belief in the statement's negation. For a defender of a paracomplete theory of truth, this kind of explanation for their rejection of L is not possible, because they also reject $\neg L$. So how could a paracompletist explain their rejection of L?

One thing that they could do is to step outside the object language $\mathcal{L}^{\text{arithm}}_{True}$ and use the metalinguistic vocabulary of Kripke's construction to explain that L does not get truth-value 1 in that construction. But this solution is unsatisfying, because it relies on going to a metalanguage rather than staying within a given language. This immediately raises the question why we don't immediately start with a language (e.g. the language of set theory) in which Kripke's construction can be performed. Actually Field [3] does start with the language of set theory,

but in that case the set-class distinction implies that Kripke's construction cannot be performed with $\forall$ interpreted a unrestrictedly quantifying over all sets, but can only be performed with $\forall$ interpreted a quantifying over the members of a fixed set U. And no matter what set U we choose, we always get false sentences that have value 1 with respect to quantification over U. In that case Kripke's construction is not a trustworthy criterion for truth, so reference to it as an explication for one's rejection of a certain sentence is not convincing.

Instead of proposing such a metalinguistic response to the question of how to explain one's rejection of L, Field [3] introduces a determinateness operator D, where $D\varphi$ intuitively means 'determinately φ'. With the help of this operator, Field can say $\neg DL$, i.e. say that the Liar sentence L is not determinately true; this is taken to be a reason for rejecting L that is expressible in the object language. In Field's account, D is not a primitive notion, but is defined in terms of a non-material conditional $\rightarrow$ that Field introduces: $D\varphi$ is taken to mean $\varphi \wedge \neg(\varphi \rightarrow \neg\varphi)$ (or equivalently $\varphi \wedge (\top \rightarrow \varphi)$). The semantics of $\rightarrow$ is explicated through a transfinite revision-rule construction. This is a construction that has some resemblance to Kripke's construction, but it does not have the monotonicity property of that construction. Instead, truth-values of sentences involving $\rightarrow$ can oscillate between different truth-values. If such an oscillation occurs all the way towards a limit ordinal λ, then the truth-value at step λ will be ½.

Once the determinateness operator is introduced, one can form a strengthened Liar sentence L_1 that is provably equivalent to $\neg D\,True(\langle L_1\rangle)$. This brings up the question what the status of this strengthened Liar sentence is. It turns out that accepting it or its negation would be problematics, but we cannot express rejection of it in the same way as in the case of the Liar sentence, because accepting $\neg DL_1$ would amount to accepting $\neg D\,True(\langle L_1\rangle)$, i.e. to accepting L_1. What we can do instead is to explain our rejection of L_1 by claiming $\neg DDL_1$, also written as $\neg D^2 L_1$. So iterating the determinateness operator yields a stronger notion of determinateness, and this stronger notion can be used to explain our stance towards a sentence involving a weaker notion of determinateness.

But then we can construct a sentence L_2 that is provably equivalent to $\neg D^2\,True(\langle L_2\rangle)$, and to explain our rejection of L_2 we need an even stronger notion of determinacy, namely D^3. Field shows that this process of iterating D can even be continued into the transfinite, but this involves some technical difficulties that go beyond the scope of this paper. It turns out that the question of how far precisely this can be meaningfully continued into the transfinite is also a very tricky one, as it touches on König's paradox of the least undefinable ordinal [7]. Field observes that for any given hereditarily definable ordinal α (i.e. for any α such that α and all its predecessors are definable), D^α is a definable and well-behaved operator. However, one cannot assume that "α is a hereditarily definable ordinal" is bivalent, because that would lead to a contradiction by König's paradox.

4 A Definable Transfinite Hierarchy of Determinateness Operators

In this section we show how a transfinite hierarchy of determinateness operators can be defined within *KFS*. Before giving the formal definition, let us first start with an informal motivation. We want to be able to say of some formulas that they have a determinate truth-value, namely being determinately true or determinately false, while saying of other formulas that they do not have a determinate truth-value. Additionally, we want this notion of determinateness to have a sensible compositional behavior with respect to the logical connectives and quantifiers, namely:

1. When t_1 and t_2 are variable-free terms that denote the same natural number, then $t_1 = t_2$ is determinately true.
2. When t_1 and t_2 are variable-free terms that denote different natural numbers, then $t_1 = t_2$ is determinately false.
3. $\bot$ is determinately false.
4. When φ is determinately true, $\neg\varphi$ is determinately false.
5. When φ is determinately false, $\neg\varphi$ is determinately true.
6. When φ is determinately false, $\varphi \wedge \psi$ is determinately false.
7. When ψ is determinately false, $\varphi \wedge \psi$ is determinately false.
8. When φ and ψ are both determinately true, $\varphi \wedge \psi$ is determinately true.
9. When $\varphi(t)$ is determinately false, $\forall x : \varphi(x)$ is determinately false.
10. When $\varphi(\bar{n})$ is determinately true for all $n \in \mathbb{N}$, $\forall x : \varphi(x)$ is determinately true. ($\bar{n}$ denotes the term $succ(\dots succ(0)\dots)$ with n occurrences of $succ$.)
11. When n is the Gödel code of a determinately true sentence, then $True(n)$ is determinately true.
12. When n is the Gödel code of a determinately false sentence, then $True(n)$ is determinately false.

One can interpret the above criteria for "determinately true" and "determinately false" as an implicit inductive definition of these two notions. If one steps out of the object language of *KFS* and allows a metatheoretic definition, one can transform the inductive definition into an explicit definition: For this, one needs to choose for the extensions of "determinately true" and "determinately false" a pair of sets that satisfies the above criteria such that the sets are minimal with respect to set inclusion among the sets with this property. We cannot quantify over sets of formulas within *KFS*, so we cannot use this strategy to turn the implicit inductive definition into an explicit definition within *KFS*.

Denecker and Vennekens [1] show that various kinds of inductive definitions used in mathematics can be given a unified semantic account with the help of the well-founded semantics of logic programs. In the following we take inspiration from the formal definition of the well-founded semantics to transform the above implicit inductive definition of determinate truth and determinate falsity into an explicit definition of a determinateness operator Δ_1 within *KFS*. Formulas of the

form $\Delta_1\varphi$ are not in general bivalent, but whenever they are bivalent, the truth-value of $\Delta_1\varphi$ is identical to the one that gets assigned to the statement "φ is definitely true" in the metatheoretic explicit definition of "determinately true".

In order to explain how our definition is inspired by the well-founded semantics, we give a brief informal sketch of the definition of the well-founded semantics; readers interested in the formal details may consult the paper by Denecker and Vennekens [1]. An *inductive definition* is a set of clauses consisting of a definiendum called *head* and a definiens called *body*. The head is always an atomic formula, and all the predicates that appear in at least one head are considered to be simultaneously defined by this inductive definition. As an example, the above enumerated list can be read as a simultaneous inductive definition of "determinately true" and "determinately false", where each item represents a clause and in each clause, the part between "When" and the comma is the body and the part after the comma is the head (as the clause about $\bot$ shows, the head can also be empty, in which case it is considered to be always true). The well-founded model of an inductive definition can be defined as the limit of a *well-founded induction*, which is a transfinite sequence of approximations to the well-founded model. In each approximation, some atoms involving one of the defined predicates are already known to be true, other such atoms are already known to be false, and others still have unknown truth-value. At each successor step, we refine the previous approximation in one of two possible ways: If our current approximation makes the body of some clause true, we may add the head of that clause to the atoms that have been accepted to be true. And if adding some atoms to the set of atoms considered false results in all bodies that define those atoms to be false, then we may indeed add those atoms to the set of atoms considered false. We continue this process until no more refinement is possible, at which point the well-founded model of the inductive definition has been reached. Note the asymmetry between making atoms true and making atoms false: We are free to assume that atoms are false, as long as this prophecy turns out to fulfill itself, whereas for considering something true at some step, we must have reasons for considering it true already at a previous step.

Inspired by the treatment of falsity in the definition of the well-founded semantics, we want to be able to say that a formula does not have a determinate truth-value if assuming it to not have a determinate truth-value turns out to be a self-fulfilling prophecy. For example, if we assume the Liar sentence L and the sentence $\mathit{True}(L)$ not to have a determinate truth-value, then for $n = \langle L\rangle$ the bodies of the clauses 11 and 12 in the above enumerated list are false, which confirms the non-determinateness of $\mathit{True}(\langle L\rangle)$, which in turn by clause 4 confirms the non-determinateness of $\neg\mathit{True}(\langle L\rangle)$, i.e. of L.

This motivates the definition of the function $\mathit{conf\_ind}$, which maps a pair $(\langle\varphi\rangle, \langle\psi(x)\rangle)$ of Gödel codes of formulas to a Gödel code $\langle\chi\rangle = \mathit{conf\_ind}(\langle\varphi\rangle, \langle\psi(x)\rangle)$, where the intuition is that when χ is satisfied then assuming the indeterminateness of all formulas whose Gödel codes satisfy ψ confirms the indeterminateness of φ:

$$
\begin{aligned}
\mathit{conf\_ind}(\langle t_1 = t_2\rangle, \langle\psi(x)\rangle) &= \langle\bot\rangle \\
\mathit{conf\_ind}(\langle\bot\rangle, \langle\psi(x)\rangle) &= \langle\bot\rangle \\
\mathit{conf\_ind}(\langle\neg\varphi\rangle, \langle\psi(x)\rangle) &= \langle\psi(\langle\varphi\rangle)\rangle \\
\mathit{conf\_ind}(\langle\varphi_1 \wedge \varphi_2\rangle, \langle\psi(x)\rangle) &= \langle(\psi(\langle\varphi_1\rangle) \vee \mathit{Pr}_{KFS}(\varphi_1)) \wedge (\psi(\langle\varphi_2\rangle) \vee \mathit{Pr}_{KFS}(\varphi_2)) \\
&\quad \wedge (\psi(\langle\varphi_1\rangle) \vee \psi(\langle\varphi_2\rangle))\rangle \\
\mathit{conf\_ind}(\langle\forall x : \varphi(x)\rangle, \langle\psi(x)\rangle) &= \langle\forall n : (\psi(\langle\varphi(\bar{n})\rangle) \vee \mathit{Pr}_{KFS}(\langle\varphi(\bar{n})\rangle)) \wedge \exists n : \psi(\langle\varphi(\bar{n})\rangle)\rangle \\
\mathit{conf\_ind}(\langle \mathit{True}(n)\rangle, \langle\psi(x)\rangle) &= \langle\psi(n)\rangle
\end{aligned}
$$

The following lemma, which one can easily derive from the definition of *conf_ind* and Kripke's construction, formalizes the idea behind the intuitive meaning of $\mathit{conf\_ind}(\langle\varphi\rangle, \langle\psi(x)\rangle)$ mentioned above:

Lemma 1. *Let $\varphi, \psi \in \mathcal{L}^{\mathrm{arithm}}_{True}$ and let α be an ordinal. If PA_{KFS} is sound, $|\mathit{conf\_ind}(\langle\varphi\rangle, \langle\psi(x)\rangle)| = 1$ and for all formulas χ such that $|\psi(\langle\chi\rangle)| = 1$, we have $\chi^\alpha = ½$, then $\varphi^{\alpha+1} = ½$.*

Now we use this function *conf_ind* to define the predicate Ind_1, where the intuitive meaning of $\mathit{Ind}_1(\langle\chi\rangle)$ is that χ has indeterminate truth-value. $\mathit{Ind}_1(\langle\chi\rangle)$ is defined to be the *KFS* formalization of the statement "There exists a number n that is the Gödel code of a formula $\psi(x)$ such that $\psi(x)$ does not contain the predicate *True*, $\psi(\langle\chi\rangle)$ is true and for any number m, if $\psi(m)$ is true then the formula whose Gödel code is $\mathit{conf\_ind}(m, n)$ is true." Intuitively, this definition says that there are some formulas (namely those whose Gödel codes satisfy $\psi(x)$) that contain χ and that have the property that assuming them to be indeterminate confirms their indeterminateness. For choosing a collection of formulas that we assume to be indeterminate, we make use of a predicate $\psi(x)$ that does not contain *True*. The fact that it does not contain *True* ensures that it is bivalent. If we allowed for an arbitrary (possibly non-bivalent) formula at this place, we would never be able to accept $\neg\mathit{Ind}_1(\chi)$ for any χ, because a non-bivalent $\psi(x)$ will ensure that the truth-value of $\mathit{Ind}_1(\chi)$ in Kripke's construction is at most ½.

The following lemma formalizes the idea that $\mathit{Ind}_1(\langle\varphi\rangle)$ expresses the indeterminateness of φ:

Lemma 2. *If PA_{KFS} is sound and $|\mathit{Ind}_1(\langle\varphi\rangle)| = 1$, then $|\varphi| = ½$.*

Proof. The soundness of PA_{KFS}, the definition of Ind_1 and the fact that $|\mathit{Ind}_1(\langle\varphi\rangle)| = 1$ together imply that there exists a formula $\psi(x)$ such that $\psi(x)$ does not contain the predicate *True*, $|\psi(\langle\varphi\rangle)| = 1$ and for any $\chi \in \mathcal{L}^{\mathrm{arithm}}_{True}$, if $|\psi(\langle\chi\rangle)| = 1$ then $|\mathit{conf\_ind}(\langle\chi\rangle, \langle\psi(x)\rangle)| = 1$. Let Γ be the set of all $\chi \in \mathcal{L}^{\mathrm{arithm}}_{True}$ such that $|\psi(\langle\chi\rangle)| = 1$.

By a transfinite induction one can prove that for every α and every $\chi \in \Gamma$, $\chi^\alpha = ½$. The inductive step directly follows from Lemma 1. Since $|\psi(\langle\varphi\rangle)| = 1$, we have that $\varphi \in \Gamma$, i.e. that $\varphi^\alpha = ½$ for all α, as required. □

Now we define the determinateness operator Δ_1 as follows: $\Delta_1\varphi$ is defined to be shorthand notation for $\varphi \wedge \neg Ind_1(\langle\varphi\rangle)$. The following theorem, which one can easily derive from the definition of Δ_1 and Lemma 2, ensures that $\neg\Delta_1\varphi$ can be used to explain one's rejection of φ.

Theorem 1. *If PA_{KFS} is sound and $|\neg\Delta_1\varphi| = 1$, then $|\varphi| = 0$ or $|\varphi| = ½$.*

Now using this determinateness operator Δ_1, we can explain our rejection of L by saying $\neg\Delta_1 L$. In order to see that $|\neg\Delta_1 L| = 1$, note that the definition of *conf_ind* implies that $|conf\_ind(\langle True(L)\rangle, \langle x = \langle L\rangle \vee x = \langle True(L)\rangle\rangle)| = 1$. The standard construction of L together with the definition of *conf_ind* furthermore implies that $|conf\_ind(\langle L\rangle, \langle x = \langle L\rangle \vee x = \langle True(L)\rangle\rangle)| = 1$. These two facts together with the definition of Ind_1 imply that $|Ind_1(\langle L\rangle)| = 1$, i.e. that $|\neg\Delta_1 L| = 1$.

Once we have successfully dealt with the Liar sentence L in this way, the obvious next question to ask is what happens to a strengthened version of the Liar that makes use of Δ_1. One can construct a strengthened Liar sentence L_1' that is provably equivalent to $\neg\Delta_1 True(\langle L_1'\rangle)$. Similarly to the case of the strengthened Liar sentence L_1 based on Field's determinateness operator D, one can show that $|L_1'| = ½$, but that the rejection of L_1' cannot be explained in the same way as the rejection of L, because $|\neg\Delta_1 L_1| = |\neg\Delta_1 True(\langle L_1'\rangle)| = |L_1'| = ½$. Unlike in the case of Field's approach, one cannot get around this problem by iterating the determinateness operator, because if $|\neg\Delta_1\Delta_1 L_1'|$ were 1, then there would be a witness $\psi(x)$ for $Ind_1(\langle\Delta_1 L_1'\rangle)$, in which case $\psi'(x) := (x = \langle L_1'\rangle \vee x = \langle\Delta_1 True(\langle L_1'\rangle)\rangle \vee \psi(x))$ would be a witness for $Ind_1(L_1')$, which would be a contradiction.

What one can do instead is to define a stronger determinateness operator Δ_2: $\Delta_2\varphi$ is shorthand for $\varphi \wedge \neg Ind_2(\langle\varphi\rangle)$, where Ind_2 is defined just like Ind_1 with the only difference being that the formula $\psi(x)$ is not required to lack the predicate *True*, but instead has to satisfy the criterion that *True* is only applied to Gödel codes of sentences not involving *True*. By choosing $\psi(x)$ to be $Ind_1(x)$, we can establish that $|Ind_2(L_1')| = 1$, i.e. that $|\neg\Delta_2 L_1'| = 1$. So we can use the stronger determinateness operator Δ_2 to explain our rejection of L_1'.

Similarly one can construct a determinateness operator Δ_3 based on a predicate Ind_3 in which the formula $\psi(x)$ may apply the predicate *True* only to formulas that apply *True* only to *True*-free formulas. In order to define these determinateness operators further, we need the following depth predicate:

- We say $depth(1, n)$ if n if the Gödel code of a formula in $\mathcal{L}^{\text{arithm}}$.
- We say $depth(\alpha+1, n)$ if α is an ordinal notation and n is the Gödel code of a formula in which *True* is only applied to a term t if t has been restricted by a syntactic criterion that implies $depth(\alpha, t)$.
- We say $depth(\lambda, n)$ if λ is a limit ordinal notation and n is the Gödel code of a formula that satisfies $depth(\alpha, n)$ for all $\alpha < \lambda$.

Now we can define Ind_α for any ordinal notation α by modifying the above definition of Ind_1 by allowing ψ to be any formula that satisfies $depth(\alpha, \langle\psi\rangle)$, and we can define $\Delta_\alpha\varphi$ to be shorthand for $\varphi \wedge \neg Ind_\alpha(\langle\varphi\rangle)$.

This defines a transfinite hierarchy of ever stronger determinateness operators. Similarly as in the case of the transfinite iterations of Field's determinateness operator, we can explain the rejection of a strengthened Liar sentence that uses a determinateness operator by using a stronger determinateness operator.

5 Conclusion

Just like Field [3], we have defined a determinateness operator to explain the rejection of the Liar sentence within the object language of our theory of truth. Unlike Field's determinateness operator, our determinateness operator can be defined within the theory *KFS* and does not require *KFS* to be extended by a conditional that is undefinable within *KFS*. Field's approach, on the other hand, is based on the idea of extending *KFS* through a semantic construction that involves the combination of a revision-rule construction for the semantics of the conditional $\rightarrow$ and Kripke's construction for the semantics of *True*, so the overall semantic construction is rather complicated. Additionally, he does not provide a proof-theoretic characterization of the ensuing theory. Our approach, on the other hand, can be completely developed within the theory *KFS* that comes out of Kripke's construction and that has a natural proof theory. Thus we avoid the complications of the extended theory for the conditional that Field has developed while achieving an equally good resolution of the Liar paradox and strengthened versions of it.

Unlike Field's determinateness operator, our determinateness operator cannot be strengthened by iterating it. Instead, we have defined a transfinite hierarchy of ever stronger determinateness operators that are not definable as iterations of the weakest determinateness operator. With this transfinite hierarchy of determinateness operators we can explain the Liar paradox and strengthened versions of it in much the same way as Field does, only that we use our stronger determinateness operators where Field uses iterations of his determinateness operator.

Field motivates his extension of *KFS* to a theory with a conditional not only through the fact that this allows him to express his rejection of the Liar sentence in the object language, but also based on the argument that it is desirable to have a conditional that satisfies certain basic properties that one would expect of a conditional but that are not satisfied by the material implication $\supset$ of *KFS*, e.g. that $\varphi \rightarrow \varphi$ and $\varphi \rightarrow (\varphi \vee \psi)$ are logical truths for any choice of φ and ψ. A thorough discussion of this issue would go beyond the scope of this paper, but let me very briefly sketch how the formal apparatus developed in this paper could also be used to define a conditional $\varphi \Rightarrow \psi$ that has such desirable properties.

For this purpose we define a predicate $Cond\_Ind(m,n)$ for conditional indeterminateness as follows: $Cond\_Ind(\langle\varphi\rangle,\langle\chi\rangle)$ is defined to be the *KFS* formalization of the statement "There exists a number n that is the Gödel code of a formula $\psi(x)$ such that $\psi(x)$ does not contain the predicate *True*, $\psi(\langle\varphi\rangle)$ is true, $\psi(\langle\chi\rangle)$ is true and for any number $m \neq \langle\varphi\rangle$, if $\psi(m)$ is true then the formula whose Gödel code is $conf\_ind(m,n)$ is true." Using this predicate, we define

$\varphi \Rightarrow \psi$ to be shorthand for $(\varphi \supset \psi) \vee \mathit{Cond\_Ind}(\langle\varphi\rangle, \langle\psi\rangle)$. One can easily verify that this conditional satisfies modus ponens and that $\varphi \Rightarrow \varphi$ and $\varphi \Rightarrow (\varphi \vee \psi)$ are true for any choice of φ and ψ. The further exploration of this conditional is left to future work.

References

1. Denecker, M., Vennekens, J.: The Well-Founded Semantics Is the Principle of Inductive Definition, Revisited. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014 (2014)
2. Feferman, S.: Reflecting on incompleteness. The Journal of Symbolic Logic **56**(1), 1–49 (1991)
3. Field, H., et al.: Saving truth from paradox. Oxford University Press (2008)
4. Kripke, S.: Outline of a theory of truth. The journal of philosophy **72**(19), 690–716 (1976)
5. Priest, G., et al.: In contradiction. Oxford University Press (2006)
6. Ripley, D.: Comparing substructural theories of truth. Ergo, an Open Access Journal of Philosophy **2** (2015)
7. Simmons, K.: Paradoxes of denotation. Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition **76**(1), 71–106 (1994)
8. Tamminga, A.: Correspondence analysis for strong three-valued logic. Логические исследования (20) (2014)
9. Van Gelder, A., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. Journal of the ACM (JACM) **38**(3), 619–649 (1991)