NIP Summer School 2011
‘Truth and Paradox’ Lecture #1

Colin Caret & Aaron Cotnoir

1. Tarskianism
2. Basic Paracomplete Theories
3. Basic Paraconsistent Theories

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$ as well as variables $x, y, z \ldots$ and perhaps some non-logical predicates and names.

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ as well as variables $x, y, z \ldots$ and perhaps some non-logical predicates and names.

- Why do we focus on theories of truth in formal languages?

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$ as well as variables $x, y, z \ldots$ and perhaps some non-logical predicates and names.

- Why do we focus on theories of truth in formal languages?
  - Modeling.

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$ as well as variables $x, y, z \dots$ and perhaps some non-logical predicates and names.

- Why do we focus on theories of truth in formal languages?
  - Modeling.
  - Regimentation.

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ as well as variables $x, y, z \ldots$ and perhaps some non-logical predicates and names.

- Why do we focus on theories of truth in formal languages?
  - Modeling.
  - Regimentation.

  to this end, we want...

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\forall$, $\exists$ as well as variables $x, y, z \ldots$ and perhaps some non-logical predicates and names.

- Why do we focus on theories of truth in formal languages?
  - Modeling.
  - Regimentation.

  to this end, we want...

  - Tractability.

Definition 1.1 (Formal Languages)

A *formal language* includes logical vocabulary $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$ as well as variables $x, y, z \dots$ and perhaps some non-logical predicates and names.

- Why do we focus on theories of truth in formal languages?
  - Modeling.
  - Regimentation.

  to this end, we want...

  - Tractability.
  - Precision.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.

- In such a language we can formulate *Peano Arithmetic* (PA), an axiom system which aims to capture all truths of basic number theory.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.

- In such a language we can formulate *Peano Arithmetic* (PA), an axiom system which aims to capture all truths of basic number theory.

- PA has some nice features which makes it a good testing ground for theories of truth. We note the following, without proof.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.
- In such a language we can formulate *Peano Arithmetic* (PA), an axiom system which aims to capture all truths of basic number theory.
- PA has some nice features which makes it a good testing ground for theories of truth. We note the following, without proof.
  - The vocabulary is countable and the grammar is recursive, so there is a recursive function that assigns each symbol and sentence a unique number.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.
- In such a language we can formulate *Peano Arithmetic* (PA), an axiom system which aims to capture all truths of basic number theory.
- PA has some nice features which makes it a good testing ground for theories of truth. We note the following, without proof.
  - The vocabulary is countable and the grammar is recursive, so there is a recursive function that assigns each symbol and sentence a unique number.
  - Given the above function, there are many properties of the language whose extensions can be represented by sets of code numbers of sentences.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.
- In such a language we can formulate *Peano Arithmetic* (PA), an axiom system which aims to capture all truths of basic number theory.
- PA has some nice features which makes it a good testing ground for theories of truth. We note the following, without proof.
  - The vocabulary is countable and the grammar is recursive, so there is a recursive function that assigns each symbol and sentence a unique number.
  - Given the above function, there are many properties of the language whose extensions can be represented by sets of code numbers of sentences.
  - All recursively definable properties have extensions which, represented by sets of code numbers, are expressible by formulas of the language itself.

- Consider a formal language for arithmetic, with a special name for zero and symbols for the successor, addition, and multiplication functions.

- In such a language we can formulate *Peano Arithmetic* (PA), an axiom system which aims to capture all truths of basic number theory.

- PA has some nice features which makes it a good testing ground for theories of truth. We note the following, without proof.

  - The vocabulary is countable and the grammar is recursive, so there is a recursive function that assigns each symbol and sentence a unique number.

  - Given the above function, there are many properties of the language whose extensions can be represented by sets of code numbers of sentences.

  - All recursively definable properties have extensions which, represented by sets of code numbers, are expressible by formulas of the language itself.

- For any sentence A of PA, we use '$\ulcorner A \urcorner$' to stand for the canonical term referring to *the code number of A*. We get the following result.

Lemma 1.1 (The Diagonalization Lemma)

For any sentence $\varphi(x)$ of PA open in one variable, there is a sentence S of PA such that the equivalence $S \leftrightarrow \varphi(\ulcorner S \urcorner)$ is a theorem of PA.

Lemma 1.1 (The Diagonalization Lemma)

For any sentence $\varphi(x)$ of PA open in one variable, there is a sentence S of PA such that the equivalence $S \leftrightarrow \varphi(\ulcorner S \urcorner)$ is a theorem of PA.

- Call the relevant sentence S the Gödel Sentence for $\varphi(x)$.

Lemma 1.1 (The Diagonalization Lemma)

For any sentence $\varphi(x)$ of PA open in one variable, there is a sentence S of PA such that the equivalence $S \leftrightarrow \varphi(\ulcorner S \urcorner)$ is a theorem of PA.

- Call the relevant sentence S the Gödel Sentence for $\varphi(x)$.
- Such sentences are interesting precisely because they involve a measure of **self-reference**, i.e. (crudely) S 'says that' it has property φ!

Lemma 1.1 (The Diagonalization Lemma)

For any sentence $\varphi(x)$ of PA open in one variable, there is a sentence S of PA such that the equivalence $S \leftrightarrow \varphi(\ulcorner S \urcorner)$ is a theorem of PA.

- Call the relevant sentence S the Gödel Sentence for $\varphi(x)$.
- Such sentences are interesting precisely because they involve a measure of **self-reference**, i.e. (crudely) S 'says that' it has property φ!

> *Gödel showed. . . that elementary syntax can be interpreted in number theory. In this way, Gödel put the issue of the legitimacy of self-referential sentences beyond doubt; he showed that they are as incontestably legitimate as arithmetic itself. (Kripke, 1975, p.692)*

- Generalizing away from details, PA offers a model for thinking about languages in which the paradoxes of self-reference can occur.

- Generalizing away from details, PA offers a model for thinking about languages in which the paradoxes of self-reference can occur.

- Going forward, we assume the legitimacy of formal languages with a device $\ulcorner\ \urcorner$ of sentence-naming, and the following property.

- Generalizing away from details, PA offers a model for thinking about languages in which the paradoxes of self-reference can occur.
- Going forward, we assume the legitimacy of formal languages with a device $\ulcorner\ \urcorner$ of sentence-naming, and the following property.

Definition 1.2 (Having Self-reference)

Language $\mathcal{L}$ *has self-reference* iff every sentence A of $\mathcal{L}$ has a standard name $\ulcorner A \urcorner$ in $\mathcal{L}$ and every predicate $\varphi(x)$ of $\mathcal{L}$ has a Gödel Sentence in $\mathcal{L}$.

- Tarski (1935) established an important limitation on languages with self-reference, by a method of proof inspired by the Liar paradox.

- Tarski (1935) established an important limitation on languages with self-reference, by a method of proof inspired by the Liar paradox.
- He argued that an adequate theory of truth should entail every instance of the T-Schema for sentences of the language to which it applies.

$$\mathrm{T}(\ulcorner A \urcorner) \leftrightarrow A$$

- Tarski (1935) established an important limitation on languages with self-reference, by a method of proof inspired by the Liar paradox.
- He argued that an adequate theory of truth should entail every instance of the T-Schema for sentences of the language to which it applies.

$$\mathrm{T}(\ulcorner A \urcorner) \leftrightarrow A$$

- He then demonstrated a case in which this is impossible.

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Proof Sketch

Fix a language that has self-reference and is underwritten by classical logic, and suppose for *reductio* that it can formulate its own adequate, non-trivial truth theory using the unary predicate $\mathrm{T}(x)$. By self-reference there exists a sentence L that is the Gödel Sentence for the predicate $\neg\mathrm{T}(x)$, i.e. (crudely) a Liar sentence which 'says that' it is not true. We reason, in this language:

□

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Proof Sketch

Fix a language that has self-reference and is underwritten by classical logic, and suppose for *reductio* that it can formulate its own adequate, non-trivial truth theory using the unary predicate $\mathrm{T}(x)$. By self-reference there exists a sentence L that is the Gödel Sentence for the predicate $\neg\mathrm{T}(x)$, i.e. (crudely) a Liar sentence which 'says that' it is not true. We reason, in this language:

(1) $\mathrm{T}(\ulcorner L \urcorner) \leftrightarrow L$ — follows from the adequacy of $\mathrm{T}(x)$

□

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Proof Sketch

Fix a language that has self-reference and is underwritten by classical logic, and suppose for *reductio* that it can formulate its own adequate, non-trivial truth theory using the unary predicate $\mathrm{T}(x)$. By self-reference there exists a sentence L that is the Gödel Sentence for the predicate $\neg\mathrm{T}(x)$, i.e. (crudely) a Liar sentence which 'says that' it is not true. We reason, in this language:

(1) $\mathrm{T}(\ulcorner L \urcorner) \leftrightarrow L$ follows from the adequacy of $\mathrm{T}(x)$

(2) $L \leftrightarrow \neg\mathrm{T}(\ulcorner L \urcorner)$ follows from the definition of L

□

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Proof Sketch

Fix a language that has self-reference and is underwritten by classical logic, and suppose for *reductio* that it can formulate its own adequate, non-trivial truth theory using the unary predicate $\mathrm{T}(x)$. By self-reference there exists a sentence L that is the Gödel Sentence for the predicate $\neg\mathrm{T}(x)$, i.e. (crudely) a Liar sentence which 'says that' it is not true. We reason, in this language:

(1) $\mathrm{T}(\ulcorner L \urcorner) \leftrightarrow L$ — follows from the adequacy of $\mathrm{T}(x)$

(2) $L \leftrightarrow \neg\mathrm{T}(\ulcorner L \urcorner)$ — follows from the definition of L

(3) $\mathrm{T}(\ulcorner L \urcorner) \vee \neg\mathrm{T}(\ulcorner L \urcorner)$ — valid by classical logic (LEM)

□

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Proof Sketch

Fix a language that has self-reference and is underwritten by classical logic, and suppose for *reductio* that it can formulate its own adequate, non-trivial truth theory using the unary predicate $\mathrm{T}(x)$. By self-reference there exists a sentence L that is the Gödel Sentence for the predicate $\neg\mathrm{T}(x)$, i.e. (crudely) a Liar sentence which 'says that' it is not true. We reason, in this language:

(1)	$\mathrm{T}(\ulcorner L \urcorner) \leftrightarrow L$	follows from the adequacy of $\mathrm{T}(x)$
(2)	$L \leftrightarrow \neg\mathrm{T}(\ulcorner L \urcorner)$	follows from the definition of L
(3)	$\mathrm{T}(\ulcorner L \urcorner) \vee \neg\mathrm{T}(\ulcorner L \urcorner)$	valid by classical logic (LEM)
(4)	$\mathrm{T}(\ulcorner L \urcorner) \wedge \neg\mathrm{T}(\ulcorner L \urcorner)$	from **(3)** by applying **(1)** and **(2)**

$\square$

Theorem 1.2 (Tarski's Undefinability Theorem)

No language with self-reference and underwritten by classical logic can formulate an adequate, non-trivial truth theory that applies to itself.

Proof Sketch

Fix a language that has self-reference and is underwritten by classical logic, and suppose for *reductio* that it can formulate its own adequate, non-trivial truth theory using the unary predicate $\mathrm{T}(x)$. By self-reference there exists a sentence L that is the Gödel Sentence for the predicate $\neg\mathrm{T}(x)$, i.e. (crudely) a Liar sentence which 'says that' it is not true. We reason, in this language:

(1)	$\mathrm{T}(\ulcorner L \urcorner) \leftrightarrow L$	follows from the adequacy of $\mathrm{T}(x)$
(2)	$L \leftrightarrow \neg\mathrm{T}(\ulcorner L \urcorner)$	follows from the definition of L
(3)	$\mathrm{T}(\ulcorner L \urcorner) \vee \neg\mathrm{T}(\ulcorner L \urcorner)$	valid by classical logic (LEM)
(4)	$\mathrm{T}(\ulcorner L \urcorner) \wedge \neg\mathrm{T}(\ulcorner L \urcorner)$	from **(3)** by applying **(1)** and **(2)**

And this entails everything—*triviality*—by classical logic (EFQ). □

- Note that talk of theories is somewhat beside the point.

- Note that talk of theories is somewhat beside the point.
- We can sharpen the issue by thinking about a predicate of a language for which the instances of the T-Schema are simply true.

- Note that talk of theories is somewhat beside the point.
- We can sharpen the issue by thinking about a predicate of a language for which the instances of the T-Schema are simply true.
- In a slogan, a classical language with self-reference cannot contain its own truth predicate—in the above sense—on pain of triviality!

- Note that talk of theories is somewhat beside the point.
- We can sharpen the issue by thinking about a predicate of a language for which the instances of the T-Schema are simply true.
- In a slogan, a classical language with self-reference cannot contain its own truth predicate—in the above sense—on pain of triviality!

Tarski's Theorem is really far more than an undefinability theorem; it says that full truth can't appear in [a non-trivial] language even as a primitive predicate. . . Of course, there's something odd about this conclusion: it should lead us to suspect that we don't even understand a concept of full truth that obeys both the Tarski schema and classical logic. (Field, 2008, p.30)

- Note that talk of theories is somewhat beside the point.

- We can sharpen the issue by thinking about a predicate of a language for which the instances of the T-Schema are simply true.

- In a slogan, a classical language with self-reference cannot contain its own truth predicate—in the above sense—on pain of triviality!

Tarski's Theorem is really far more than an undefinability theorem; it says that full truth can't appear in [a non-trivial] language even as a primitive predicate. . . Of course, there's something odd about this conclusion: it should lead us to suspect that we don't even understand a concept of full truth that obeys both the Tarski schema and classical logic. (Field, 2008, p.30)

- This perfectly captures the usual interpretation of Tarski's Theorem as a *dilemma*: revise the concept of truth, or revise classical logic.

- Why not take the first route? Let's consider this briefly.

- Why not take the first route? Let's consider this briefly.

- Since a classical language with self-reference cannot contain its own truth predicate, perhaps some *other* language could do so.

- Why not take the first route? Let's consider this briefly.
- Since a classical language with self-reference cannot contain its own truth predicate, perhaps some *other* language could do so.
- This was exactly Tarski's proposal: ascend from language $\mathcal{L}_k$ to define theory X in a richer language $\mathcal{L}_{k+1}$ such that for all sentences A of $\mathcal{L}_k$:

$$X \vDash_{k+1} \mathrm{T_k}(\ulcorner A \urcorner) \leftrightarrow A$$

(think of 'T_k' as 'is a true sentence of $\mathcal{L}_k$')

- This circumvents the triviality immanent in Tarski's Theorem by biting the bullet and accepting that no language contains its own truth predicate.

- This circumvents the triviality immanent in Tarski's Theorem by biting the bullet and accepting that no language contains its own truth predicate.
- It rules out the *existence* of Liar sentences (as ungrammatical).

- This circumvents the triviality immanent in Tarski's Theorem by biting the bullet and accepting that no language contains its own truth predicate.
- It rules out the *existence* of Liar sentences (as ungrammatical).
- And the method iterates, allowing us to re-apply it to metalangauge $\mathcal{L}_{k+1}$ by ascending to a richer metametalanguage $\mathcal{L}_{k+2}$, and so on.

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.

- This has serious drawbacks as a theory of the ordinary concept of truth (which, it is worth pointing out, was never Tarski’s intention).

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.
- This has serious drawbacks as a theory of the ordinary concept of truth (which, it is worth pointing out, was never Tarski’s intention).
- It gives the wrong analysis of certain mundane assertions. (Kripke)

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.
- This has serious drawbacks as a theory of the ordinary concept of truth (which, it is worth pointing out, was never Tarski’s intention).
- It gives the wrong analysis of certain mundane assertions. (Kripke)

Dean: Nothing that Nixon says about Watergate is true.

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.

- This has serious drawbacks as a theory of the ordinary concept of truth (which, it is worth pointing out, was never Tarski’s intention).

- It gives the wrong analysis of certain mundane assertions. (Kripke)

  Dean: Nothing that Nixon says about Watergate is true.

  Nixon: Nothing that Dean says about Watergate is true.

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.
- This has serious drawbacks as a theory of the ordinary concept of truth (which, it is worth pointing out, was never Tarski’s intention).
- It gives the wrong analysis of certain mundane assertions. (Kripke)

  Dean: Nothing that Nixon says about Watergate is true.

  Nixon: Nothing that Dean says about Watergate is true.

- And it places artificial limits on our grasp of certain concepts. (Priest)

- By design, a Tarskian approach ‘fragments’ truth into many distinct concept each of which is only applicable to a restricted language.

- This has serious drawbacks as a theory of the ordinary concept of truth (which, it is worth pointing out, was never Tarski’s intention).

- It gives the wrong analysis of certain mundane assertions. (Kripke)

    Dean: Nothing that Nixon says about Watergate is true.

    Nixon: Nothing that Dean says about Watergate is true.

- And it places artificial limits on our grasp of certain concepts. (Priest)

    If A is ‘a true sentence of $\mathcal{L}_k$’ for some level k of the ‘hierarchy’ of languages, then A is *simply true*, but this predicate is not contained in any language.

- Motivation in hand, we turn to non-classical theories.

- Motivation in hand, we turn to non-classical theories.
- Note the conspicuous role played in the Liar argument by the supposedly ‘vacuous’ inferences of *Excluded Middle* (LEM) and *Ex Falso* (EFQ).

- Motivation in hand, we turn to non-classical theories.
- Note the conspicuous role played in the Liar argument by the supposedly 'vacuous' inferences of *Excluded Middle* (LEM) and *Ex Falso* (EFQ).
- The theories ahead raise the question: could LEM or EFQ be invalid?

$$A \overset{?}{\nvDash} B \vee \neg B \qquad\qquad A \wedge \neg A \overset{?}{\nvDash} B$$

- Motivation in hand, we turn to non-classical theories.
- Note the conspicuous role played in the Liar argument by the supposedly 'vacuous' inferences of *Excluded Middle* (LEM) and *Ex Falso* (EFQ).
- The theories ahead raise the question: could LEM or EFQ be invalid?

$$A \overset{?}{\nvDash} B \vee \neg B \qquad\qquad A \wedge \neg A \overset{?}{\nvDash} B$$

- We focus on the *propositional* fragment of the logics we consider because this is where most of the 'revisionary' work needs to be done.

- A qualification is also in order: Gödel Sentences have been implicitly treated as (provable) equivalents using the material biconditional.

- A qualification is also in order: Gödel Sentences have been implicitly treated as (provable) equivalents using the material biconditional.
- This will not always be available in the theories ahead.

- A qualification is also in order: Gödel Sentences have been implicitly treated as (provable) equivalents using the material biconditional.
- This will not always be available in the theories ahead.
- For this reason, we will often be interested in a *strengthened* criterion for our truth predicate that is distinct from, but in the spirit of, the T-Schema.

- A qualification is also in order: Gödel Sentences have been implicitly treated as (provable) equivalents using the material biconditional.
- This will not always be available in the theories ahead.
- For this reason, we will often be interested in a *strengthened* criterion for our truth predicate that is distinct from, but in the spirit of, the T-Schema.

Definition 1.3 (Transparency)

Let β be any sentence in which sentence α occurs. Then the result of substituting $\mathrm{T}(\ulcorner\alpha\urcorner)$ for any occurrence of α in β has the same semantic value or same semantic status as β. (Beall, 2009, p.15)

- We begin by considering theories that reject LEM, the so-called *paracomplete* theories (*para-* ‘beyond’ the negation-*complete*).

- We begin by considering theories that reject LEM, the so-called *paracomplete* theories (*para*- 'beyond' the negation-*complete*).
- We will call sentences that are counterexamples to LEM GAPS.

- We begin by considering theories that reject LEM, the so-called *paracomplete* theories (*para-* ‘beyond’ the negation-*complete*).
- We will call sentences that are counterexamples to LEM GAPS.
- It is tempting to think of gaps as being ‘neither true nor false’. We shall see later why this temptation might need to be resisted. For now, it is useful to think of gaps simply as counterexamples to LEM.

- Why think the Liar is a gap?

- Why think the Liar is a gap?
- Kripke painted a picture about the ‘grounding’ of the concept of truth.

- Why think the Liar is a gap?
- Kripke painted a picture about the ‘grounding’ of the concept of truth.
- Imagine teaching someone to use ‘true’ by saying: assert $\mathrm{T}(\ulcorner A \urcorner)$ whenever you would assert A, deny $\mathrm{T}(\ulcorner A \urcorner)$ whenever you would deny A.

- Why think the Liar is a gap?
- Kripke painted a picture about the ‘grounding’ of the concept of truth.
- Imagine teaching someone to use ‘true’ by saying: assert $\mathrm{T}(\ulcorner A \urcorner)$ whenever you would assert A, deny $\mathrm{T}(\ulcorner A \urcorner)$ whenever you would deny A.
- Suppose she sorts through every sentence of her language and applies these instructions in all the cases that she can (in some idealized sense).

- Why think the Liar is a gap?
- Kripke painted a picture about the ‘grounding’ of the concept of truth.
- Imagine teaching someone to use ‘true’ by saying: assert $\mathrm{T}(\ulcorner A \urcorner)$ whenever you would assert A, deny $\mathrm{T}(\ulcorner A \urcorner)$ whenever you would deny A.
- Suppose she sorts through every sentence of her language and applies these instructions in all the cases that she can (in some idealized sense).
- With the Liar and other self-referential sentences, she will suspend judgment; the procedure never tells her to assert or deny them.

- This story involves an element like Tarskian semantic ascent.

- This story involves an element like Tarskian semantic ascent.
- But the difference is that it portrays an iterated filling out of a *univocal* concept of truth (rather than ‘truth-at-the-next-level’).

- This story involves an element like Tarskian semantic ascent.
- But the difference is that it portrays an iterated filling out of a *univocal* concept of truth (rather than 'truth-at-the-next-level').
- The story allows that the predicate may not end up fully defined, e.g. with respect to the Liar, she will not assert $\mathrm{T}(\ulcorner L \urcorner)$ nor assert $\neg\mathrm{T}(\ulcorner L \urcorner)$.

- This story involves an element like Tarskian semantic ascent.
- But the difference is that it portrays an iterated filling out of a *univocal* concept of truth (rather than 'truth-at-the-next-level').
- The story allows that the predicate may not end up fully defined, e.g. with respect to the Liar, she will not assert $\mathrm{T}(\ulcorner L \urcorner)$ nor assert $\neg\mathrm{T}(\ulcorner L \urcorner)$.
- And now we have a counterexample to LEM.

- This story involves an element like Tarskian semantic ascent.
- But the difference is that it portrays an iterated filling out of a *univocal* concept of truth (rather than ‘truth-at-the-next-level’).
- The story allows that the predicate may not end up fully defined, e.g. with respect to the Liar, she will not assert $\mathrm{T}(\ulcorner L \urcorner)$ nor assert $\neg\mathrm{T}(\ulcorner L \urcorner)$.
- And now we have a counterexample to LEM.
- Let us make this a bit more precise.

- Let $\mathcal{V}$ be any ordered set, and let $x, y \in \mathcal{V}$.

- Let $\mathcal{V}$ be any ordered set, and let $x, y \in \mathcal{V}$.

Definition 2.1 (LUB)

The *least upper bound* of x, y ($\text{lub}\{x, y\}$) is the lowest value greater-than-or-equal-to x and y.

- Let $\mathcal{V}$ be any ordered set, and let $x, y \in \mathcal{V}$.

Definition 2.1 (LUB)

The *least upper bound* of x, y ($\text{lub}\{x, y\}$) is the lowest value greater-than-or-equal-to x and y.

Definition 2.2 (GLB)

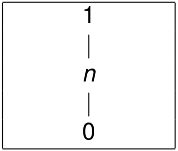
The *greatest lower bound* of x, y ($\text{glb}\{x, y\}$) is the highest value less-than-or-equal to x and y.

Semantics for K_3

- The set of values $\mathcal{V}$ for $\mathcal{L}_{K_3}$ is $\{1, n, 0\}$.

Semantics for K_3

- The set of values $\mathcal{V}$ for $\mathcal{L}_{K_3}$ is $\{1, n, 0\}$.
- These values are *ordered* in the following way.



Semantics for K_3

A K_3 valuation is *admissible* iff it satisfies the following clauses.

Semantics for K_3

A K_3 valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, n, 0\}$ for all atomic A.

Semantics for K_3

A K_3 valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, n, 0\}$ for all atomic A.

(2) $\nu(A \vee B) = \text{lub}\{\nu(A), \nu(B)\}$.

Semantics for K_3

A K_3 valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, n, 0\}$ for all atomic A.

(2) $\nu(A \vee B) = \text{lub}\{\nu(A), \nu(B)\}$.

(3) $\nu(A \wedge B) = \text{glb}\{\nu(A), \nu(B)\}$.

Semantics for K_3

A K_3 valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, n, 0\}$ for all atomic A.

(2) $\nu(A \vee B) = \text{lub}\{\nu(A), \nu(B)\}$.

(3) $\nu(A \wedge B) = \text{glb}\{\nu(A), \nu(B)\}$.

(4) $\nu(\neg A) = \begin{cases} 1 \text{ if } \nu(A) = 0 \\ n \text{ if } \nu(A) = n \\ 0 \text{ if } \nu(A) = 1 \end{cases}$

Representation in Truth Tables

$\neg$	
1	0
n	n
0	1

$\wedge$	1	n	0
1	1	n	0
n	n	n	0
0	0	0	0

$\vee$	1	n	0
1	1	1	1
n	1	n	n
0	1	n	0

$\rightarrow$	1	n	0
1	1	n	0
n	1	n	n
0	1	1	1

$\leftrightarrow$	1	n	0
1	1	n	0
n	n	n	n
0	0	n	1

NB: the conditional here is the material conditional $(A \rightarrow B) =_{\text{df.}} (\neg A \vee B)$

Semantics for K_3

Let $\mathcal{D} = \{1\}$ be a set of *designated* values.

Semantics for K_3

Let $\mathcal{D} = \{1\}$ be a set of *designated* values.

- Sentence A is *designated* by an K_3-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.

Semantics for K_3

Let $\mathcal{D} = \{1\}$ be a set of *designated* values.

- Sentence A is *designated* by an K_3-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.
- Sentence A is *satisfied* iff it is designated.

Semantics for K_3

Let $\mathcal{D} = \{1\}$ be a set of *designated* values.

- Sentence A is *designated* by an K_3-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.
- Sentence A is *satisfied* iff it is designated.
- $\models_{K_3} A$ iff A is satisfied by every K_3-admissible valuation.

Semantics for K_3

Let $\mathcal{D} = \{1\}$ be a set of *designated* values.

- Sentence A is *designated* by an K_3-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.
- Sentence A is *satisfied* iff it is designated.
- $\models_{K_3} A$ iff A is satisfied by every K_3-admissible valuation.
- $\mathcal{X} \models_{K_3} A$ iff every K_3-admissible valuation that satisfies $\mathcal{X}$ satisfies A.

- K_3 logic is called *Strong Kleene* because it is due to Kleene (1950).

- K_3 logic is called *Strong Kleene* because it is due to Kleene (1950).
- It is truth-functional, and has classical logic as an extension.

- K_3 logic is called *Strong Kleene* because it is due to Kleene (1950).
- It is truth-functional, and has classical logic as an extension.
- There are counterexamples to the *Excluded Middle* (LEM): $\nvDash_{K_3} A \vee \neg A$.

- K_3 logic is called *Strong Kleene* because it is due to Kleene (1950).
- It is truth-functional, and has classical logic as an extension.
- There are counterexamples to the *Excluded Middle* (LEM): $\not\models_{K_3} A \vee \neg A$.
- As such, K_3 is *paracomplete*.

K_3TT

- K_3 is the background logic for our basic paracomplete truth theory.

K_3TT

- K_3 is the background logic for our basic paracomplete truth theory.
- Our base language $\mathcal{L}$ is classical *Peano Arithmetic* (PA).

K_3TT

- K_3 is the background logic for our basic paracomplete truth theory.
- Our base language $\mathcal{L}$ is classical *Peano Arithmetic* (PA).
- We want to extend this language with a transparent truth predicate T.

K_3TT

- K_3 is the background logic for our basic paracomplete truth theory.
- Our base language $\mathcal{L}$ is classical *Peano Arithmetic* (PA).
- We want to extend this language with a transparent truth predicate T.
- Kripke (1975) showed us how to do this, one ‘stage’ at a time.

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^{+}$.

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
- In classical logic it follows that $\nu(\neg\mathrm{T}(x)) = 1$ iff $\hat{x} \notin \mathrm{T}^+$. But this doesn't leave room for objects that fail to be T and also fail to be $\neg\mathrm{T}$.

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
- In classical logic it follows that $\nu(\neg\mathrm{T}(x)) = 1$ iff $\hat{x} \notin \mathrm{T}^+$. But this doesn't leave room for objects that fail to be T and also fail to be $\neg\mathrm{T}$.
- In K_3 we interpret predicates via extensions and *anti-extensions*:

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
- In classical logic it follows that $\nu(\neg\mathrm{T}(x)) = 1$ iff $\hat{x} \notin \mathrm{T}^+$. But this doesn't leave room for objects that fail to be T and also fail to be $\neg\mathrm{T}$.
- In K_3 we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.

- In classical logic it follows that $\nu(\neg\mathrm{T}(x)) = 1$ iff $\hat{x} \notin \mathrm{T}^+$. But this doesn't leave room for objects that fail to be T and also fail to be $\neg\mathrm{T}$.

- In K_3 we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
  - $\nu(\mathrm{T}(x)) = 0$ iff $\hat{x} \in \mathrm{T}^-$.

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
- In classical logic it follows that $\nu(\neg\mathrm{T}(x)) = 1$ iff $\hat{x} \notin \mathrm{T}^+$. But this doesn't leave room for objects that fail to be T and also fail to be $\neg\mathrm{T}$.
- In K_3 we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
  - $\nu(\mathrm{T}(x)) = 0$ iff $\hat{x} \in \mathrm{T}^-$.
  - $\nu(\mathrm{T}(x)) = n$ iff $\hat{x} \notin \mathrm{T}^+$ and $\hat{x} \notin \mathrm{T}^-$

K_3TT

- The classical way of interpreting predicates is via *extensions*: $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
- In classical logic it follows that $\nu(\neg\mathrm{T}(x)) = 1$ iff $\hat{x} \notin \mathrm{T}^+$. But this doesn't leave room for objects that fail to be T and also fail to be $\neg\mathrm{T}$.
- In K_3 we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^+$.
  - $\nu(\mathrm{T}(x)) = 0$ iff $\hat{x} \in \mathrm{T}^-$.
  - $\nu(\mathrm{T}(x)) = n$ iff $\hat{x} \notin \mathrm{T}^+$ and $\hat{x} \notin \mathrm{T}^-$
  - (We assume nothing is in both the extension and anti-extension.)

K_3TT

- If $\mathcal{L}$ is the intended model of PA, let $\mathcal{L}_0$ be the result of adding T to the syntax, but leaving the extension T^+ and anti-extension T^- empty. This is stage 0; sometimes called the 'empty ground model'.

K_3TT

- If $\mathcal{L}$ is the intended model of PA, let $\mathcal{L}_0$ be the result of adding T to the syntax, but leaving the extension T^+ and anti-extension T^- empty. This is stage 0; sometimes called the 'empty ground model'.
- At stage 1:

K_3TT

- If $\mathcal{L}$ is the intended model of PA, let $\mathcal{L}_0$ be the result of adding T to the syntax, but leaving the extension T^+ and anti-extension T^- empty. This is stage 0; sometimes called the 'empty ground model'.
- At stage 1:
  - Throw every sentence of $\mathcal{L}_0$ that receives value 1 into T^+.

K_3TT

- If $\mathcal{L}$ is the intended model of PA, let $\mathcal{L}_0$ be the result of adding T to the syntax, but leaving the extension T^+ and anti-extension T^- empty. This is stage 0; sometimes called the 'empty ground model'.
- At stage 1:
  - Throw every sentence of $\mathcal{L}_0$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_0$ that receives value 0 into T^-.

K_3TT

- If $\mathcal{L}$ is the intended model of PA, let $\mathcal{L}_0$ be the result of adding T to the syntax, but leaving the extension T^+ and anti-extension T^- empty. This is stage 0; sometimes called the 'empty ground model'.
- At stage 1:
  - Throw every sentence of $\mathcal{L}_0$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_0$ that receives value 0 into T^-.
  - Now, we close this under the rules of K_3. The result is called $\mathcal{L}_1$

K_3TT

- If $\mathcal{L}$ is the intended model of PA, let $\mathcal{L}_0$ be the result of adding T to the syntax, but leaving the extension T^+ and anti-extension T^- empty. This is stage 0; sometimes called the ‘empty ground model’.
- At stage 1:
  - Throw every sentence of $\mathcal{L}_0$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_0$ that receives value 0 into T^-.
  - Now, we close this under the rules of K_3. The result is called $\mathcal{L}_1$
- So, at stage 1, we have $\nu(\mathrm{T}(\ulcorner A \urcorner)) = 1$ for any true sentence of arithmetic, and $\nu(\mathrm{T}(\ulcorner A \urcorner)) = 0$ and hence $\nu(\neg\mathrm{T}(\ulcorner A \urcorner)) = 1$ for any false sentence of arithmetic.

K_3TT

- The process continues as follows.

K_3TT

- The process continues as follows.
- At stage 2:

K_3TT

- The process continues as follows.
- At stage 2:
  - Throw every sentence of $\mathcal{L}_1$ that receives value 1 into T^+.

K_3TT

- The process continues as follows.
- At stage 2:
  - Throw every sentence of $\mathcal{L}_1$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_1$ that receives value 0 into T^-.

K_3TT

- The process continues as follows.
- At stage 2:
  - Throw every sentence of $\mathcal{L}_1$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_1$ that receives value 0 into T^-.
  - Now, we close this under the rules of K_3. The result is called $\mathcal{L}_2$

K_3TT

- The process continues as follows.
- At stage 2:
  - Throw every sentence of $\mathcal{L}_1$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_1$ that receives value 0 into T^-.
  - Now, we close this under the rules of K_3. The result is called $\mathcal{L}_2$
- So, at stage 2, we have $\nu(\mathrm{T}(\ulcorner \mathrm{T}(\ulcorner A \urcorner)\urcorner)) = 1$ for any true sentence of arithmetic, and $\nu(\mathrm{T}(\ulcorner \mathrm{T}(\ulcorner A \urcorner)\urcorner)) = 0$ and hence $\nu(\neg\mathrm{T}(\ulcorner \mathrm{T}(\ulcorner A \urcorner)\urcorner)) = 1$ for any false sentence of arithmetic.

K_3TT

- The process continues as follows.
- At stage 2:
  - Throw every sentence of $\mathcal{L}_1$ that receives value 1 into T^+.
  - Throw every sentence of $\mathcal{L}_1$ that receives value 0 into T^-.
  - Now, we close this under the rules of K_3. The result is called $\mathcal{L}_2$
- So, at stage 2, we have $\nu(\mathrm{T}(\ulcorner \mathrm{T}(\ulcorner A \urcorner)\urcorner)) = 1$ for any true sentence of arithmetic, and $\nu(\mathrm{T}(\ulcorner \mathrm{T}(\ulcorner A \urcorner)\urcorner)) = 0$ and hence $\nu(\neg\mathrm{T}(\ulcorner \mathrm{T}(\ulcorner A \urcorner)\urcorner)) = 1$ for any false sentence of arithmetic.
- Continue on!

K_3TT

- At some stage this process will stop; there will be a ‘fixed point’.

K_3TT

- At some stage this process will stop; there will be a 'fixed point'.
- Our 'jumps' from stage to stage will eventually result in us running out of sentences which our 'instructions' will add to either T^+ or T^-. All the sentences that *can* receive value 1 or 0 will have by this fixed point.

K_3TT

- At some stage this process will stop; there will be a ‘fixed point’.
- Our ‘jumps’ from stage to stage will eventually result in us running out of sentences which our ‘instructions’ will add to either T^+ or T^-. All the sentences that *can* receive value 1 or 0 will have by this fixed point.
- But there will be some sentences – the ‘ungrounded’ ones – which will never resolve to a classical value. These sentences, like the Liar, will be stuck with value *n* forever.

K_3TT

- At some stage this process will stop; there will be a 'fixed point'.
- Our 'jumps' from stage to stage will eventually result in us running out of sentences which our 'instructions' will add to either T^+ or T^-. All the sentences that *can* receive value 1 or 0 will have by this fixed point.
- But there will be some sentences – the 'ungrounded' ones – which will never resolve to a classical value. These sentences, like the Liar, will be stuck with value *n* forever.
- Kripke also showed, although we won't prove it here, that the resulting interpretation for T is *transparent*. That is a great achievement!

Problems with K_3TT

- But not everything is brilliant about this approach to truth.

Problems with K_3TT

- But not everything is brilliant about this approach to truth.
- There are counterexamples to *Material Identity*! $\nvDash_{K_3} A \rightarrow A$.

$\rightarrow$	1	n	0
1	1	n	0
n	1	n	n
0	1	1	1

Problems with K_3TT

- But not everything is brilliant about this approach to truth.
- There are counterexamples to *Material Identity*! $\nvDash_{K_3} A \rightarrow A$.

$\rightarrow$	1	n	0
1	1	n	0
n	1	n	n
0	1	1	1

- Note connection with *Conditional Proof*, and the *Deduction Theorem*: if $A \vDash A$ then $\vDash A \rightarrow A$

Problems with K_3TT

- But not everything is brilliant about this approach to truth.
- There are counterexamples to *Material Identity*! $\not\models_{K_3} A \rightarrow A$.

$\rightarrow$	1	n	0
1	1	n	0
n	1	n	n
0	1	1	1

- Note connection with *Conditional Proof*, and the *Deduction Theorem*: if $A \models A$ then $\models A \rightarrow A$
- As a result, there are counterexamples to the *Material T-Biconditionals*! $\not\models_{K_3} \mathrm{T}(\ulcorner A \urcorner) \leftrightarrow A$

- Finally, there is no way of asserting of gaps that they are ‘gappy’.

- Finally, there is no way of asserting of gaps that they are ‘gappy’.
- Notice, the intuitive conception of gaps as ‘neither true nor false’ would naturally be formalized as $\neg(\mathrm{T}(\ulcorner A \urcorner) \vee \neg\mathrm{T}(\ulcorner A \urcorner))$.

- Finally, there is no way of asserting of gaps that they are ‘gappy’.
- Notice, the intuitive conception of gaps as ‘neither true nor false’ would naturally be formalized as $\neg(\mathrm{T}(\ulcorner A \urcorner) \vee \neg\mathrm{T}(\ulcorner A \urcorner))$.
- But in the logics we have considered (and will consider in lecture 2) this is equivalent to $\neg\mathrm{T}(\ulcorner A \urcorner) \wedge \mathrm{T}(\ulcorner A \urcorner)$, which is the assertion of a glut.

- Finally, there is no way of asserting of gaps that they are ‘gappy’.
- Notice, the intuitive conception of gaps as ‘neither true nor false’ would naturally be formalized as $\neg(\mathrm{T}(\ulcorner A \urcorner) \vee \neg\mathrm{T}(\ulcorner A \urcorner))$.
- But in the logics we have considered (and will consider in lecture 2) this is equivalent to $\neg\mathrm{T}(\ulcorner A \urcorner) \wedge \mathrm{T}(\ulcorner A \urcorner)$, which is the assertion of a glut.
- Furthermore, when the sentence in question, $\mathrm{T}(\ulcorner A \urcorner)$, fails to receive a classical value in any fixed point, so does $\neg\mathrm{T}(\ulcorner A \urcorner)$, and thus so do the sentences $\mathrm{T}(\ulcorner A \urcorner) \wedge \neg\mathrm{T}(\ulcorner A \urcorner)$ and $\neg(\mathrm{T}(\ulcorner A \urcorner) \vee \neg\mathrm{T}(\ulcorner A \urcorner))$.

- Ł$_3$ is nearly identical to K_3 except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.

- Ł$_3$ is nearly identical to K_3 except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.

- Ł$_3$-admissible valuations are K_3-admissible valuations that follow the truth table for $\rightarrow$ given below.

$\rightarrow$	1	n	0
1	1	n	0
n	1	1	n
0	1	1	1

- $Ł_3$ is nearly identical to K_3 except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.

- $Ł_3$-admissible valuations are K_3-admissible valuations that follow the truth table for $\rightarrow$ given below.

$\rightarrow$	1	n	0
1	1	n	0
n	1	1	n
0	1	1	1

- This patches one obvious weakness of K_3, Identity: $\vDash_{Ł_3} A \rightarrow A$

- Ł$_3$ is nearly identical to K_3 except that it treats the conditional $\to$ as a syntactic primitive, not definable in terms of the other connectives.
- Ł$_3$-admissible valuations are K_3-admissible valuations that follow the truth table for $\to$ given below.

$\to$	1	n	0
1	1	n	0
n	1	1	n
0	1	1	1

- This patches one obvious weakness of K$_3$, Identity: $\vDash_{Ł_3} A \to A$
- Which means that a truth theory on this logic can have T-biconditionals.

Problems with Ł$_3$

- But the Ł$_3$ conditional *contracts*: $A \rightarrow (A \rightarrow B) \vDash_{Ł_3} A \rightarrow B$

Problems with Ł$_3$

- But the Ł$_3$ conditional *contracts*: $A \rightarrow (A \rightarrow B) \vDash_{Ł_3} A \rightarrow B$
- Contraction is (often) thought to be the problematic feature of conditionals that lead to Curry's paradox.

Problems with Ł$_3$

- But the Ł$_3$ conditional *contracts*: $A \rightarrow (A \rightarrow B) \vDash_{Ł_3} A \rightarrow B$
- Contraction is (often) thought to be the problematic feature of conditionals that lead to Curry's paradox.
- The *advanced* paracomplete theories to be discussed in lecture 2 attempt to address these problems.

- We now move on to consider theories that reject EFQ, the so-called *paraconsistent* theories (*para*- 'beyond' the negation-*consistent*).

- We now move on to consider theories that reject EFQ, the so-called *paraconsistent* theories (*para*- ‘beyond’ the negation-*consistent*).
- We will call sentences that are counterexamples to EFQ GLUTS.

- We now move on to consider theories that reject EFQ, the so-called *paraconsistent* theories (*para*- 'beyond' the negation-*consistent*).
- We will call sentences that are counterexamples to EFQ GLUTS.
- We will speak of gluts as being 'both true and false'. This is accurate, but in the logics we consider $\mathrm{T}(\ulcorner A \urcorner) \wedge \neg\mathrm{T}(\ulcorner A \urcorner)$ and $\neg(\mathrm{T}(\ulcorner A \urcorner) \vee \neg\mathrm{T}(\ulcorner A \urcorner))$ are equivalent, so gluts are 'gappy' in the sense we *couldn't* say of GAPS.

- Why think the Liar is a glut?

- Why think the Liar is a glut?
- Well, the Liar sentence *means that* that sentence itself is not true, it is a sentence whose truth-conditions are its falsity-conditions.

- Why think the Liar is a glut?

- Well, the Liar sentence *means that* that sentence itself is not true, it is a sentence whose truth-conditions are its falsity-conditions.

- That's suggestive, but a more interesting route is to argue that we can *prove* that the Liar is a glut with minimal resources.

$$
\dfrac{\dfrac{\overline{\mathrm{T}(\ulcorner L \urcorner) \rightarrow \neg\mathrm{T}(\ulcorner L \urcorner)}\ (\text{TS})}{\neg\mathrm{T}(\ulcorner L \urcorner)}\ (\neg\text{E}) \qquad \dfrac{\dfrac{\dfrac{\vdots}{\neg\mathrm{T}(\ulcorner L \urcorner)}}{L}\ (\text{def. } L)}{\mathrm{T}(\ulcorner L \urcorner)}\ (\text{Nec})}{\mathrm{T}(\ulcorner L \urcorner) \wedge \neg\mathrm{T}(\ulcorner L \urcorner)}\ (\wedge\text{I})
$$

- We can also dualise the Kripkean picture in a ‘falsificationist’ vein.

- We can also dualise the Kripkean picture in a ‘falsificationist’ vein.

- Imagine teaching someone the rules: assert $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to deny A, deny $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to assert A.

- We can also dualise the Kripkean picture in a ‘falsificationist’ vein.
- Imagine teaching someone the rules: assert $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to deny A, deny $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to assert A.
- Suppose she sorts through every sentence of her language and applies these instructions in all the cases that she can (. . . idealized. . .).

- We can also dualise the Kripkean picture in a ‘falsificationist’ vein.
- Imagine teaching someone the rules: assert $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to deny A, deny $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to assert A.
- Suppose she sorts through every sentence of her language and applies these instructions in all the cases that she can (. . . idealized. . .).
- With the Liar and other self-referential sentences, she will go back and forth so she will never have decisive reason to *only* assert or deny it.

- We can also dualise the Kripkean picture in a ‘falsificationist’ vein.
- Imagine teaching someone the rules: assert $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to deny A, deny $\mathrm{T}(\ulcorner A \urcorner)$ unless you have reason to assert A.
- Suppose she sorts through every sentence of her language and applies these instructions in all the cases that she can (. . . idealized. . .).
- With the Liar and other self-referential sentences, she will go back and forth so she will never have decisive reason to *only* assert or deny it.
- She won’t as a result of this indecisiveness accept *everything*, so we have a counterexample to EFQ. Let us make this a bit more precise.

Semantics for LP

- The set of values $\mathcal{V}$ for *LP* is $\{1, b, 0\}$.

Semantics for LP

- The set of values $\mathcal{V}$ for *LP* is $\{1, b, 0\}$.
- These values are *ordered* in the following way.

$$\begin{array}{c} 1 \\ | \\ b \\ | \\ 0 \end{array}$$

Semantics for LP

A *LP* valuation is *admissible* iff it satisfies the following clauses.

Semantics for LP

A *LP* valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, b, 0\}$ for all atomic *A*.

Semantics for LP

A *LP* valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, b, 0\}$ for all atomic A.

(2) $\nu(A \vee B) = \text{lub}\{\nu(A), \nu(B)\}$.

Semantics for LP

A *LP* valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, b, 0\}$ for all atomic A.

(2) $\nu(A \vee B) = \text{lub}\{\nu(A), \nu(B)\}$.

(3) $\nu(A \wedge B) = \text{glb}\{\nu(A), \nu(B)\}$.

Semantics for LP

A *LP* valuation is *admissible* iff it satisfies the following clauses.

(1) $\nu(A) \in \{1, b, 0\}$ for all atomic A.

(2) $\nu(A \vee B) = \text{lub}\{\nu(A), \nu(B)\}$.

(3) $\nu(A \wedge B) = \text{glb}\{\nu(A), \nu(B)\}$.

(4) $\nu(\neg A) = \begin{cases} 1 \text{ if } \nu(A) = 0 \\ b \text{ if } \nu(A) = b \\ 0 \text{ if } \nu(A) = 1 \end{cases}$

Representation in Truth Tables

$\neg$	
1	0
b	b
0	1

$\wedge$	1	b	0
1	1	b	0
b	b	b	0
0	0	0	0

$\vee$	1	b	0
1	1	1	1
b	1	b	b
0	1	b	0

$\rightarrow$	1	b	0
1	1	b	0
b	1	b	b
0	1	1	1

$\leftrightarrow$	1	b	0
1	1	b	0
b	b	b	b
0	0	b	1

NB: same as K_3, with only difference which elements are designated

Semantics for LP

Let $\mathcal{D} = \{1, b\}$ be a set of *designated* values.

Semantics for LP

Let $\mathcal{D} = \{1, b\}$ be a set of *designated* values.

- Sentence *A* is *designated* by an *LP*-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.

Semantics for LP

Let $\mathcal{D} = \{1, b\}$ be a set of *designated* values.

- Sentence A is *designated* by an LP-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.
- Sentence A is *satisfied* iff it is designated.

Semantics for LP

Let $\mathcal{D} = \{1, b\}$ be a set of *designated* values.

- Sentence A is *designated* by an *LP*-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.
- Sentence A is *satisfied* iff it is designated.
- $\models_{\text{LP}} A$ iff A is satisfied by every *LP*-admissible valuation.

Semantics for LP

Let $\mathcal{D} = \{1, b\}$ be a set of *designated* values.

- Sentence A is *designated* by an *LP*-admissible valuation ν iff $\nu(A) \in \mathcal{D}$.
- Sentence A is *satisfied* iff it is designated.
- $\vDash_{\mathrm{LP}} A$ iff A is satisfied by every *LP*-admissible valuation.
- $\mathcal{X} \vDash_{\mathrm{LP}} A$ iff every *LP*-admissible valuation that satisfies $\mathcal{X}$ satisfies A.

- *LP* is called the *Logic of Paradox* due to Asenjo (1966)

- *LP* is called the *Logic of Paradox* due to Asenjo (1966)
- It is truth-functional, and has classical logic as an extension.

- *LP* is called the *Logic of Paradox* due to Asenjo (1966)
- It is truth-functional, and has classical logic as an extension.
- There are counterexamples to *Ex Falso* (EFQ): $A \wedge \neg A \nvDash_{LP} B$.

- *LP* is called the *Logic of Paradox* due to Asenjo (1966)
- It is truth-functional, and has classical logic as an extension.
- There are counterexamples to *Ex Falso* (EFQ): $A \wedge \neg A \not\models_{LP} B$.
- As such, *LP* is *paraconsistent*.

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.
- Same base language PA, same procedure, extending with T 'by stages'.

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.
- Same base language PA, same procedure, extending with T 'by stages'.
- In *LP* we interpret predicates via extensions and *anti-extensions*:

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.
- Same base language PA, same procedure, extending with T ‘by stages’.
- In *LP* we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^{+}$.

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.
- Same base language PA, same procedure, extending with T 'by stages'.
- In *LP* we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^{+}$.
  - $\nu(\mathrm{T}(x)) = 0$ iff $\hat{x} \in \mathrm{T}^{-}$.

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.
- Same base language PA, same procedure, extending with T 'by stages'.
- In *LP* we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^{+}$.
  - $\nu(\mathrm{T}(x)) = 0$ iff $\hat{x} \in \mathrm{T}^{-}$.
  - $\nu(\mathrm{T}(x)) = b$ iff $\hat{x} \in \mathrm{T}^{+}$ and $\hat{x} \in \mathrm{T}^{-}$

LPTT

- *LP* is the background logic for our basic paraconsistent truth theory.
- Same base language PA, same procedure, extending with T 'by stages'.
- In *LP* we interpret predicates via extensions and *anti-extensions*:
  - $\nu(\mathrm{T}(x)) = 1$ iff $\hat{x} \in \mathrm{T}^{+}$.
  - $\nu(\mathrm{T}(x)) = 0$ iff $\hat{x} \in \mathrm{T}^{-}$.
  - $\nu(\mathrm{T}(x)) = b$ iff $\hat{x} \in \mathrm{T}^{+}$ and $\hat{x} \in \mathrm{T}^{-}$
  - We assume everything is either in the extension or anti-extension.

LPTT

- Again, let $\mathcal{L}_0$ be the result of adding T to the syntax, but this time we let every sentence be in both T^+ and T^-. This is stage 0, a 'full ground model' stage 0.

LPTT

- Again, let $\mathcal{L}_0$ be the result of adding T to the syntax, but this time we let every sentence be in both T^+ and T^-. This is stage 0, a 'full ground model' stage 0.
- At stage 1:

LPTT

- Again, let $\mathcal{L}_0$ be the result of adding T to the syntax, but this time we let every sentence be in both T^+ and T^-. This is stage 0, a 'full ground model' stage 0.
- At stage 1:
  - Remove every sentence of $\mathcal{L}_0$ that receives value 1 from T^-.

LPTT

- Again, let $\mathcal{L}_0$ be the result of adding T to the syntax, but this time we let every sentence be in both T^+ and T^-. This is stage 0, a 'full ground model' stage 0.
- At stage 1:
  - Remove every sentence of $\mathcal{L}_0$ that receives value 1 from T^-.
  - Remove every sentence of $\mathcal{L}_0$ that receives value 0 from T^+.

LPTT

- Again, let $\mathcal{L}_0$ be the result of adding T to the syntax, but this time we let every sentence be in both T^+ and T^-. This is stage 0, a ‘full ground model’ stage 0.
- At stage 1:
  - Remove every sentence of $\mathcal{L}_0$ that receives value 1 from T^-.
  - Remove every sentence of $\mathcal{L}_0$ that receives value 0 from T^+.
  - Now, we close this under the rules of *LP*. The result is called $\mathcal{L}_1$

LPTT

- Again, let $\mathcal{L}_0$ be the result of adding T to the syntax, but this time we let every sentence be in both T^+ and T^-. This is stage 0, a 'full ground model' stage 0.
- At stage 1:
  - Remove every sentence of $\mathcal{L}_0$ that receives value 1 from T^-.
  - Remove every sentence of $\mathcal{L}_0$ that receives value 0 from T^+.
  - Now, we close this under the rules of *LP*. The result is called $\mathcal{L}_1$
- This process continues just as before up through the stages.

K_3TT

- Again, at some stage this process will stop; there will be a ‘fixed point’. All the sentences that can receive value 1 or 0 will.

K_3TT

- Again, at some stage this process will stop; there will be a 'fixed point'. All the sentences that can receive value 1 or 0 will.
- But there will be some sentences – the very same 'ungrounded' ones – which will never resolve to a classical value. They will be stuck with value *b* forever, always in both the extension and anti-extension of T.

K_3TT

- Again, at some stage this process will stop; there will be a 'fixed point'. All the sentences that can receive value 1 or 0 will.
- But there will be some sentences – the very same 'ungrounded' ones – which will never resolve to a classical value. They will be stuck with value *b* forever, always in both the extension and anti-extension of T.
- Likewise, the resulting interpretation for T is *transparent*. Great news!

Problems with LPTT

- But again, not everything is brilliant about this theory.

Problems with LPTT

- But again, not everything is brilliant about this theory.
- There are counterexamples to MODUS PONENS! $A \rightarrow B, A \nvDash_{LP} B$.

$\rightarrow$	1	b	0
1	1	b	0
b	1	b	b
0	1	1	1

Problems with LPTT

- But again, not everything is brilliant about this theory.
- There are counterexamples to MODUS PONENS! $A \rightarrow B, A \nvDash_{LP} B$.

$\rightarrow$	1	b	0
1	1	b	0
b	1	b	b
0	1	1	1

- Note connection with *'Converse' Conditional Proof* and the *Deduction Theorem*: if $A \rightarrow B \vDash A \rightarrow B$ then $A \rightarrow B, A \vDash B$

Problems with LPTT

- But again, not everything is brilliant about this theory.
- There are counterexamples to MODUS PONENS! $A \rightarrow B, A \nvDash_{LP} B$.

$\rightarrow$	1	b	0
1	1	b	0
b	1	b	b
0	1	1	1

- Note connection with *'Converse' Conditional Proof* and the *Deduction Theorem*: if $A \rightarrow B \vDash A \rightarrow B$ then $A \rightarrow B, A \vDash B$
- Notice also that some of the *Material T-Biconditionals*, the ones instantiantiated on gluts, are *false* (though they are also true).

- *RM* is nearly identical to *LP* except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.

- *RM* is nearly identical to *LP* except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.
- *RM*-admissible valuations are *LP*-admissible valuations that follow the truth table for $\rightarrow$ given below.

- *RM* is nearly identical to *LP* except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.
- *RM*-admissible valuations are *LP*-admissible valuations that follow the truth table for $\rightarrow$ given below.

- *RM* is nearly identical to *LP* except that it treats the conditional $\rightarrow$ as a syntactic primitive, not definable in terms of the other connectives.

- *RM*-admissible valuations are *LP*-admissible valuations that follow the truth table for $\rightarrow$ given below.

$\rightarrow$	1	b	0
1	1	0	0
b	1	b	0
0	1	1	1

- This patches one obvious weakness of *LP*, MPP: $A \rightarrow B, A \vDash_{RM} B$

Problems with *RM*

- But the *RM* conditional also *contracts*: $A \rightarrow (A \rightarrow B) \vDash_{RM_3} A \rightarrow B$

Problems with *RM*

- But the *RM* conditional also *contracts*: $A \rightarrow (A \rightarrow B) \vDash_{RM_3} A \rightarrow B$
- Again, this leads to the complicated trouble of Curry's paradox.

Problems with *RM*

- But the *RM* conditional also *contracts*: $A \rightarrow (A \rightarrow B) \vDash_{RM_3} A \rightarrow B$
- Again, this leads to the complicated trouble of Curry's paradox.
- The *advanced* paraconsistent theories to be discussed in lecture 3 attempt to address these problems.

References

F.G. Asenjo. A Calculus of Antinomies. *Notre Dame Journal of Formal Logic*, 7(1):103–105, 1966.

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

Hartry Field. *Saving Truth From Paradox*. Oxford University Press, 2008.

Saul Kripke. Outline of a Theory of Truth. *The Journal of Philosophy*, 72(19): 690–716, 1975.

Alfred Tarski. Der Wahrheitsbegriff in Den Formaliserten Sprachen. *Studia Philosophica*, 1:261–405, 1935. Reprinted as 'The Concept of Truth in Formalized Languages' in *Logic, Semantics, Metamathematics*, trans. J.H. Woodger, Oxford University Press, 1956.