A Do-It-Yourself Guide to the Construction of Full Satisfaction Classes

Ali Enayat & Albert Visser

Amsterdam Workshop on Truth

March 15, 2013

Universiteit Utrecht

What is the Game? 1

- As Tarski taught us, we can expand a model of any theory with a satisfaction predicate. We want to play a somewhat more refined game.
- What we want to do is use some of the internal resources of the model to construct the satisfaction predicate.
- The main resource is coding of sequences and coding of syntax. We can demand that this is done inside the model. A convenient way to realize this is to consider *sequential models*.
- We note that in the model we could have non-standard formulas. So we would have to build a satisfaction predicate also applying to these.

What is the Game? 1

- As Tarski taught us, we can expand a model of any theory with a satisfaction predicate. We want to play a somewhat more refined game.
- What we want to do is use some of the internal resources of the model to construct the satisfaction predicate.
- The main resource is coding of sequences and coding of syntax. We can demand that this is done inside the model. A convenient way to realize this is to consider *sequential models*.
- We note that in the model we could have non-standard formulas. So we would have to build a satisfaction predicate also applying to these.

What is the Game? 1

- As Tarski taught us, we can expand a model of any theory with a satisfaction predicate. We want to play a somewhat more refined game.
- What we want to do is use some of the internal resources of the model to construct the satisfaction predicate.
- The main resource is coding of sequences and coding of syntax. We can demand that this is done inside the model. A convenient way to realize this is to consider *sequential models*.
- We note that in the model we could have non-standard formulas. So we would have to build a satisfaction predicate also applying to these.

What is the Game? 1

- As Tarski taught us, we can expand a model of any theory with a satisfaction predicate. We want to play a somewhat more refined game.
- What we want to do is use some of the internal resources of the model to construct the satisfaction predicate.
- The main resource is coding of sequences and coding of syntax. We can demand that this is done inside the model. A convenient way to realize this is to consider *sequential models*.
- We note that in the model we could have non-standard formulas. So we would have to build a satisfaction predicate also applying to these.

What is the Game? 1

- As Tarski taught us, we can expand a model of any theory with a satisfaction predicate. We want to play a somewhat more refined game.
- What we want to do is use some of the internal resources of the model to construct the satisfaction predicate.
- The main resource is coding of sequences and coding of syntax. We can demand that this is done inside the model. A convenient way to realize this is to consider *sequential models*.
- We note that in the model we could have non-standard formulas. So we would have to build a satisfaction predicate also applying to these.

What is the Game? 2

- We can play the game in two distinct ways. We can ask whether we can do the construction of satisfaction for a *designated* choice of the internal sequences and syntax or we can ask whether we can *find* sequences and syntax that do the job. I am not sure yet which of these two ways is the more fruitful one.
- We can shift the perspective from models to theories. E.g. we can ask whether a certain theory can be expanded with a theory of satisfaction in a conservative way.
- A theory U has a conservative extension U^+ iff for every model $\mathcal{M}$ of U there is an elementary extension $\mathcal{M}^+$ satisfying U^+.

Universiteit Utrecht

What is the Game? 2

- We can play the game in two distinct ways. We can ask whether we can do the construction of satisfaction for a *designated* choice of the internal sequences and syntax or we can ask whether we can *find* sequences and syntax that do the job. I am not sure yet which of these two ways is the more fruitful one.
- We can shift the perspective from models to theories. E.g. we can ask whether a certain theory can be expanded with a theory of satisfaction in a conservative way.
- A theory U has a conservative extension U^+ iff for every model $\mathcal{M}$ of U there is an elementary extension $\mathcal{M}^+$ satisfying U^+.

Universiteit Utrecht

What is the Game? 2

- We can play the game in two distinct ways. We can ask whether we can do the construction of satisfaction for a *designated* choice of the internal sequences and syntax or we can ask whether we can *find* sequences and syntax that do the job. I am not sure yet which of these two ways is the more fruitful one.
- We can shift the perspective from models to theories. E.g. we can ask whether a certain theory can be expanded with a theory of satisfaction in a conservative way.
- A theory U has a conservative extension U^+ iff for every model $\mathcal{M}$ of U there is an elementary extension $\mathcal{M}^+$ satisfying U^+.

Universiteit Utrecht

What is the Game? 2

- We can play the game in two distinct ways. We can ask whether we can do the construction of satisfaction for a *designated* choice of the internal sequences and syntax or we can ask whether we can *find* sequences and syntax that do the job. I am not sure yet which of these two ways is the more fruitful one.
- We can shift the perspective from models to theories. E.g. we can ask whether a certain theory can be expanded with a theory of satisfaction in a conservative way.
- A theory U has a conservative extension U^+ iff for every model $\mathcal{M}$ of U there is an elementary extension $\mathcal{M}^+$ satisfying U^+.

Universiteit Utrecht

What is the Game? 3

- In this talk we will zoom in on a specific part of the game. We will consider strong sequential theories, mainly theories with full induction and even more specifically extensions of Peano Arithmetic. We will work with designated numbers and, if we have full induction, we take as our numbers a choice of numbers that satisfies full induction.
- We will consider the second more relaxed notion of extension: not just expansion but elementary extension. So we will ask how to find an elementary extension of a sequential model of full induction that has a satisfaction predicate satisfying further properties.
- http://igitur-archive.library.uu.nl/ph/2013-0212-200423/preprint303.pdf

Universiteit Utrecht

What is the Game? 3

- In this talk we will zoom in on a specific part of the game. We will consider strong sequential theories, mainly theories with full induction and even more specifically extensions of Peano Arithmetic. We will work with designated numbers and, if we have full induction, we take as our numbers a choice of numbers that satisfies full induction.
- We will consider the second more relaxed notion of extension: not just expansion but elementary extension. So we will ask how to find an elementary extension of a sequential model of full induction that has a satisfaction predicate satisfying further properties.
- http://igitur-archive.library.uu.nl/ph/2013-0212-200423/preprint303.pdf

Universiteit Utrecht

What is the Game? 3

- In this talk we will zoom in on a specific part of the game. We will consider strong sequential theories, mainly theories with full induction and even more specifically extensions of Peano Arithmetic. We will work with designated numbers and, if we have full induction, we take as our numbers a choice of numbers that satisfies full induction.
- We will consider the second more relaxed notion of extension: not just expansion but elementary extension. So we will ask how to find an elementary extension of a sequential model of full induction that has a satisfaction predicate satisfying further properties.
- http://igitur-archive.library.uu.nl/ph/2013-0212-200423/preprint303.pdf

Universiteit Utrecht

What is the Game? 3

- In this talk we will zoom in on a specific part of the game. We will consider strong sequential theories, mainly theories with full induction and even more specifically extensions of Peano Arithmetic. We will work with designated numbers and, if we have full induction, we take as our numbers a choice of numbers that satisfies full induction.
- We will consider the second more relaxed notion of extension: not just expansion but elementary extension. So we will ask how to find an elementary extension of a sequential model of full induction that has a satisfaction predicate satisfying further properties.
- http://igitur-archive.library.uu.nl/ph/2013-0212-200423/preprint303.pdf

Universiteit Utrecht

What is a Base Theory?

- PA and ZF are the most familiar examples of base theories. Intuitively speaking, a base theory B should have a modicum of coding machinery to support the notion of a *satisfaction class*.
- B is a base theory if it is *sequential*, i.e., if we can define a binary relation $\in$ satisfying Adjunctive Set Theory AS which is given by the following axioms:
  a. there is an empty set
  b. for any x and y there is some z whose elements are precisely $x \cup \{y\}$ (adjunction)

  We do not demand extensionality.

Universiteit Utrecht

What is a Base Theory?

- PA and ZF are the most familiar examples of base theories. Intuitively speaking, a base theory B should have a modicum of coding machinery to support the notion of a *satisfaction class.*
- B is a base theory if it is *sequential*, i.e., if we can define a binary relation $\in$ satisfying Adjunctive Set Theory AS which is given by the following axioms:
  a. there is an empty set
  b. for any x and y there is some z whose elements are precisely $x \cup \{y\}$ (adjunction)

  We do not demand extensionality.

Universiteit Utrecht

What is a Base Theory?

- PA and ZF are the most familiar examples of base theories. Intuitively speaking, a base theory B should have a modicum of coding machinery to support the notion of a *satisfaction class.*
- B is a base theory if it is *sequential*, i.e., if we can define a binary relation $\in$ satisfying Adjunctive Set Theory AS which is given by the following axioms:
  a. there is an empty set
  b. for any x and y there is some z whose elements are precisely $x \cup \{y\}$ (adjunction)

  We do not demand extensionality.

Universiteit Utrecht

Axioms for Satisfaction 1

- Given a base theory B we wish to define certain canonical associated *satisfaction* theories here denoted B^{fs}, B^{is}, and B^{fis}, all of which are formulated in an *expansion* of the language $\mathcal{L}_B$ by adding a unary predicate F and a binary predicate S.
- We extend B with the axioms Tarski(S, F):
- $\mathrm{tarski}_0(\mathsf{S},\mathsf{F}) := (\mathsf{F}(x) \to \mathrm{Form}(x)) \wedge$
  $(\mathsf{S}(x,\alpha) \to (\mathsf{F}(x) \wedge \mathrm{asn}(\alpha, x))) \wedge ((y \lhd x \wedge \mathsf{F}(x)) \to F(y))$.
- $\mathrm{tarski}_{1,R}(\mathsf{S},\mathsf{F}) :=$

$$(\mathsf{F}(x) \wedge (x = \ulcorner R(v_0, \cdots, v_{n-1})\urcorner) \wedge \mathrm{asn}(\alpha, x)) \to \\ \left(\mathsf{S}(x,\alpha) \leftrightarrow R\left([\alpha]_{v_0} \cdots, [\alpha]_{v_{n-1}}\right)\right).$$

Universiteit Utrecht

Axioms for Satisfaction 1

- Given a base theory B we wish to define certain canonical associated *satisfaction* theories here denoted B^{fs}, B^{is}, and B^{fis}, all of which are formulated in an *expansion* of the language $\mathcal{L}_B$ by adding a unary predicate F and a binary predicate S.
- We extend B with the axioms Tarski(S, F):
- $\mathrm{tarski}_0(\mathsf{S},\mathsf{F}) := (\mathsf{F}(x) \to \mathrm{Form}(x)) \wedge$
  $(\mathsf{S}(x,\alpha) \to (\mathsf{F}(x) \wedge \mathrm{asn}(\alpha, x))) \wedge ((y \lhd x \wedge \mathsf{F}(x)) \to F(y))$.
- $\mathrm{tarski}_{1,R}(\mathsf{S},\mathsf{F}) :=$

$$(\mathsf{F}(x) \wedge (x = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner) \wedge \mathrm{asn}(\alpha, x)) \to \\ (\mathsf{S}(x,\alpha) \leftrightarrow R([\alpha]_{v_0} \cdots, [\alpha]_{v_{n-1}}))\,.$$

Universiteit Utrecht

Axioms for Satisfaction 1

- Given a base theory B we wish to define certain canonical associated *satisfaction* theories here denoted B^{fs}, B^{is}, and B^{fis}, all of which are formulated in an *expansion* of the language $\mathcal{L}_B$ by adding a unary predicate F and a binary predicate S.
- We extend B with the axioms Tarski(S, F):
- $\mathrm{tarski}_0(\mathsf{S},\mathsf{F}) := (\mathsf{F}(x) \to \mathsf{Form}(x)) \wedge$
  $(\mathsf{S}(x,\alpha) \to (\mathsf{F}(x) \wedge \mathsf{asn}(\alpha, x))) \wedge ((y \lhd x \wedge \mathsf{F}(x)) \to F(y))$.
- $\mathrm{tarski}_{1,R}(\mathsf{S},\mathsf{F}) :=$

$$(\mathsf{F}(x) \wedge (x = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner) \wedge \mathsf{asn}(\alpha, x)) \to \\ (\mathsf{S}(x,\alpha) \leftrightarrow R([\alpha]_{v_0} \cdots, [\alpha]_{v_{n-1}}))\,.$$

Universiteit Utrecht

Axioms for Satisfaction 1

- Given a base theory B we wish to define certain canonical associated *satisfaction* theories here denoted B^{fs}, B^{is}, and B^{fis}, all of which are formulated in an *expansion* of the language $\mathcal{L}_B$ by adding a unary predicate F and a binary predicate S.
- We extend B with the axioms Tarski(S, F):
- $\mathrm{tarski}_0(\mathsf{S},\mathsf{F}) := (\mathsf{F}(x) \to \mathsf{Form}(x)) \wedge$
  $(\mathsf{S}(x,\alpha) \to (\mathsf{F}(x) \wedge \mathsf{asn}(\alpha, x))) \wedge ((y \lhd x \wedge \mathsf{F}(x)) \to F(y))$.
- $\mathrm{tarski}_{1,R}(\mathsf{S},\mathsf{F}) :=$

$$(\mathsf{F}(x) \wedge (x = \ulcorner R(v_0, \cdots, v_{n-1})\urcorner) \wedge \mathsf{asn}(\alpha, x)) \to (\mathsf{S}(x,\alpha) \leftrightarrow R([\alpha]_{v_0} \cdots, [\alpha]_{v_{n-1}}))\,.$$

Universiteit Utrecht

Axioms for Satisfaction 1

- Given a base theory B we wish to define certain canonical associated *satisfaction* theories here denoted B^{fs}, B^{is}, and B^{fis}, all of which are formulated in an *expansion* of the language $\mathcal{L}_B$ by adding a unary predicate F and a binary predicate S.
- We extend B with the axioms Tarski(S, F):
- $\mathrm{tarski}_0(\mathsf{S},\mathsf{F}) := (\mathsf{F}(x) \to \mathsf{Form}(x)) \wedge$
  $(\mathsf{S}(x,\alpha) \to (\mathsf{F}(x) \wedge \mathsf{asn}(\alpha, x))) \wedge ((y \lhd x \wedge \mathsf{F}(x)) \to F(y))$.
- $\mathrm{tarski}_{1,R}(\mathsf{S},\mathsf{F}) :=$

$$(\mathsf{F}(x) \wedge (x = \ulcorner R(v_0, \cdots, v_{n-1})\urcorner) \wedge \mathsf{asn}(\alpha, x)) \to \\ \left(\mathsf{S}(x,\alpha) \leftrightarrow R\left([\alpha]_{v_0} \cdots, [\alpha]_{v_{n-1}}\right)\right).$$

Universiteit Utrecht

Axioms for Satisfaction 2

- $\mathsf{tarski}_2(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \neg y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow (\mathsf{S}(x,\alpha) \leftrightarrow \neg \mathsf{S}(y,\alpha)).$$
- $\mathsf{tarski}_3(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner y_1 \vee y_2 \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow$$
$$\left(\mathsf{S}(x,\alpha) \leftrightarrow \left(\begin{array}{c}\mathsf{S}(y_1, \alpha \restriction \mathsf{fv}(y_1)) \vee \\ \mathsf{S}(y_2, \alpha \restriction \mathsf{fv}(y_2))\end{array}\right)\right).$$
- $\mathsf{tarski}_4(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \exists v\, y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow$$
$$\left(\begin{array}{c}\mathsf{S}(x,\alpha) \leftrightarrow \\ \exists \alpha' \supseteq \alpha\, \mathsf{S}(y,\alpha')\end{array}\right).$$

Universiteit Utrecht

Axioms for Satisfaction 2

- $\mathsf{tarski}_2(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \neg y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow (\mathsf{S}(x,\alpha) \leftrightarrow \neg \mathsf{S}(y,\alpha)).$$
- $\mathsf{tarski}_3(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner y_1 \vee y_2 \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow \left(\mathsf{S}(x,\alpha) \leftrightarrow \left(\begin{array}{c}\mathsf{S}(y_1, \alpha \restriction \mathsf{fv}(y_1)) \vee \\ \mathsf{S}(y_2, \alpha \restriction \mathsf{fv}(y_2))\end{array}\right)\right).$$
- $\mathsf{tarski}_4(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \exists v\ y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow \left(\begin{array}{c}\mathsf{S}(x,\alpha) \leftrightarrow \\ \exists \alpha' \supseteq \alpha\ \mathsf{S}(y,\alpha')\end{array}\right).$$

Universiteit Utrecht

Axioms for Satisfaction 2

- $\mathsf{tarski}_2(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \neg y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow (\mathsf{S}(x,\alpha) \leftrightarrow \neg \mathsf{S}(y,\alpha)).$$
- $\mathsf{tarski}_3(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner y_1 \vee y_2 \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow$$
$$\left(\mathsf{S}(x,\alpha) \leftrightarrow \left(\begin{array}{c}\mathsf{S}(y_1, \alpha \restriction \mathsf{fv}(y_1)) \vee \\ \mathsf{S}(y_2, \alpha \restriction \mathsf{fv}(y_2))\end{array}\right)\right).$$
- $\mathsf{tarski}_4(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \exists v\, y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow$$
$$\left(\begin{array}{c}\mathsf{S}(x,\alpha) \leftrightarrow \\ \exists \alpha' \supseteq \alpha\, \mathsf{S}(y,\alpha')\end{array}\right).$$

Universiteit Utrecht

Axioms for Satisfaction 2

- $\mathsf{tarski}_2(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \neg y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow (\mathsf{S}(x,\alpha) \leftrightarrow \neg \mathsf{S}(y,\alpha)).$$
- $\mathsf{tarski}_3(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner y_1 \vee y_2 \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow$$
$$\left(\mathsf{S}(x,\alpha) \leftrightarrow \left(\begin{array}{c}\mathsf{S}(y_1, \alpha \restriction \mathsf{fv}(y_1)) \vee \\ \mathsf{S}(y_2, \alpha \restriction \mathsf{fv}(y_2))\end{array}\right)\right).$$
- $\mathsf{tarski}_4(\mathsf{S},\mathsf{F}) :=$
$$\left(\begin{array}{c}\mathsf{F}(x) \wedge (x = \ulcorner \exists v\ y \urcorner) \wedge \\ \mathsf{asn}(\alpha, x)\end{array}\right) \rightarrow$$
$$\left(\begin{array}{c}\mathsf{S}(x,\alpha) \leftrightarrow \\ \exists \alpha' \supseteq \alpha\ \mathsf{S}(y,\alpha')\end{array}\right).$$

Universiteit Utrecht

Extensions of the Base Theory

- $B^{\mathrm{fs}} := B \cup \mathrm{Tarski}\,(\mathrm{S}, \mathrm{form})$.
- $B^{\mathrm{is}} := B \cup \bigcup \{\mathrm{Tarski}\,(\mathrm{S}, \mathrm{form}_n) : n \in \omega\} \cup \mathrm{ind}(\mathrm{S})$,

  where form_n is the collection of formulas $\mathcal{L}_{\mathrm{B}}$ with quantifier alternation depth at most n and $\mathrm{ind}(\mathrm{S})$ is the full scheme of induction on the designated numbers in the language $\mathcal{L}_B(\mathrm{S})$.
- $B^{\mathrm{fis}} := B^{\mathrm{fs}} \cup B^{\mathrm{is}}$.

Universiteit Utrecht

Extensions of the Base Theory

- $B^{\mathrm{fs}} := B \cup \mathrm{Tarski}\,(\mathrm{S}, \mathrm{form})$.
- $B^{\mathrm{is}} := B \cup \bigcup \{\mathrm{Tarski}\,(\mathrm{S}, \mathrm{form}_n) : n \in \omega\} \cup \mathrm{ind}(\mathrm{S})$,

  where form_n is the collection of formulas $\mathcal{L}_{\mathrm{B}}$ with quantifier alternation depth at most n and $\mathrm{ind}(\mathrm{S})$ is the full scheme of induction on the designated numbers in the language $\mathcal{L}_B(\mathrm{S})$.
- $B^{\mathrm{fis}} := B^{\mathrm{fs}} \cup B^{\mathrm{is}}$.

Universiteit Utrecht

Extensions of the Base Theory

- $B^{\mathrm{fs}} := B \cup \mathrm{Tarski}\,(\mathrm{S}, \mathrm{form})$.
- $B^{\mathrm{is}} := B \cup \bigcup \{\mathrm{Tarski}\,(\mathrm{S}, \mathrm{form}_n) : n \in \omega\} \cup \mathrm{ind}(\mathrm{S})$,

  where form_n is the collection of formulas $\mathcal{L}_{\mathrm{B}}$ with quantifier alternation depth at most n and $\mathrm{ind}(\mathrm{S})$ is the full scheme of induction on the designated numbers in the language $\mathcal{L}_B(\mathrm{S})$.
- $B^{\mathrm{fis}} := B^{\mathrm{fs}} \cup B^{\mathrm{is}}$.

Universiteit Utrecht

Extensions of the Base Theory

- $B^{\mathrm{fs}} := B \cup \mathrm{Tarski}\,(\mathrm{S}, \mathrm{form})$.
- $B^{\mathrm{is}} := B \cup \bigcup \{\mathrm{Tarski}\,(\mathrm{S}, \mathrm{form}_n) : n \in \omega\} \cup \mathrm{ind}(\mathrm{S})$,

  where form_n is the collection of formulas $\mathcal{L}_{\mathrm{B}}$ with quantifier alternation depth at most n and $\mathrm{ind}(\mathrm{S})$ is the full scheme of induction on the designated numbers in the language $\mathcal{L}_B(\mathrm{S})$.
- $B^{\mathrm{fis}} := B^{\mathrm{fs}} \cup B^{\mathrm{is}}$.

Universiteit Utrecht

Satisfaction Classes

- Suppose $\mathcal{M} \models B$ and F is a subset of M and S is a binary relation on M.
- S is an *F-satisfaction class* if $(\mathcal{M}, S, F) \models \mathrm{Tarski(S, F)}$.
- If $F = \mathrm{Form}^{\mathcal{M}} \cap \omega$, then we say that F is the set of *standard* $\mathcal{L}_{\mathrm{B}}$-formulas of $\mathcal{M}$. In this case there is a unique F-satisfaction class on $\mathcal{M}$, which we refer to as the *Tarskian satisfaction class on $\mathcal{M}$.*
- S is a *full satisfaction class on $\mathcal{M}$* if S is an *F-satisfaction class* for $F := \mathrm{Form}^{\mathcal{M}}$. This is equivalent to $(\mathcal{M}, S, F) \models \mathrm{B}^{\mathrm{fs}}$.

Universiteit Utrecht

Satisfaction Classes

- Suppose $\mathcal{M} \models B$ and F is a subset of M and S is a binary relation on M.
- S is an *F-satisfaction class* if $(\mathcal{M}, S, F) \models \mathrm{Tarski(S, F)}$.
- If $F = \mathrm{Form}^{\mathcal{M}} \cap \omega$, then we say that F is the set of *standard* $\mathcal{L}_{\mathrm{B}}$-formulas of $\mathcal{M}$. In this case there is a unique F-satisfaction class on $\mathcal{M}$, which we refer to as the *Tarskian satisfaction class on $\mathcal{M}$.*
- S is a *full satisfaction class on $\mathcal{M}$* if S is an *F-satisfaction class* for $F := \mathrm{Form}^{\mathcal{M}}$. This is equivalent to $(\mathcal{M}, S, F) \models \mathrm{B}^{\mathrm{fs}}$.

Universiteit Utrecht

Satisfaction Classes

- Suppose $\mathcal{M} \models B$ and F is a subset of M and S is a binary relation on M.
- S is an *F-satisfaction class* if $(\mathcal{M}, S, F) \models \mathrm{Tarski(S, F)}$.
- If $F = \mathrm{Form}^{\mathcal{M}} \cap \omega$, then we say that F is the set of *standard* $\mathcal{L}_{\mathrm{B}}$-formulas of $\mathcal{M}$. In this case there is a unique F-satisfaction class on $\mathcal{M}$, which we refer to as the *Tarskian satisfaction class on $\mathcal{M}$.*
- S is a *full satisfaction class on $\mathcal{M}$* if S is an *F-satisfaction class* for $F := \mathrm{Form}^{\mathcal{M}}$. This is equivalent to $(\mathcal{M}, S, F) \models \mathrm{B}^{\mathrm{fs}}$.

Universiteit Utrecht

Satisfaction Classes

- Suppose $\mathcal{M} \models B$ and F is a subset of M and S is a binary relation on M.
- S is an *F-satisfaction class* if $(\mathcal{M}, S, F) \models \mathrm{Tarski(S, F)}$.
- If $F = \mathrm{Form}^{\mathcal{M}} \cap \omega$, then we say that F is the set of *standard* $\mathcal{L}_{\mathrm{B}}$-formulas of $\mathcal{M}$. In this case there is a unique F-satisfaction class on $\mathcal{M}$, which we refer to as the *Tarskian satisfaction class on $\mathcal{M}$.*
- S is a *full satisfaction class on $\mathcal{M}$* if S is an *F-satisfaction class* for $F := \mathrm{Form}^{\mathcal{M}}$. This is equivalent to $(\mathcal{M}, S, F) \models \mathrm{B}^{\mathrm{fs}}$.

Universiteit Utrecht

Satisfaction Classes

- Suppose $\mathcal{M} \models B$ and F is a subset of M and S is a binary relation on M.
- S is an *F-satisfaction class* if $(\mathcal{M}, S, F) \models \mathrm{Tarski(S, F)}$.
- If $F = \mathrm{Form}^{\mathcal{M}} \cap \omega$, then we say that F is the set of *standard* $\mathcal{L}_{\mathrm{B}}$-formulas of $\mathcal{M}$. In this case there is a unique F-satisfaction class on $\mathcal{M}$, which we refer to as the *Tarskian satisfaction class on $\mathcal{M}$.*
- S is a *full satisfaction class on $\mathcal{M}$* if S is an *F-satisfaction class* for $F := \mathrm{Form}^{\mathcal{M}}$. This is equivalent to $(\mathcal{M}, S, F) \models \mathrm{B}^{\mathrm{fs}}$.

Universiteit Utrecht

The Conservativity of $\mathrm{PA}^{\mathrm{FS}}$ over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. $\mathrm{PA}^{\mathrm{fs}}$ *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Conservativity of PA^{FS} over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. PA^{fs} *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Conservativity of $\mathrm{PA}^{\mathrm{FS}}$ over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. $\mathrm{PA}^{\mathrm{fs}}$ *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Conservativity of $\mathrm{PA}^{\mathrm{FS}}$ over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. $\mathrm{PA}^{\mathrm{fs}}$ *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Conservativity of $\mathsf{PA}^{\mathsf{FS}}$ over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. $\mathsf{PA}^{\mathsf{fs}}$ *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Conservativity of $\mathsf{PA}^{\mathsf{FS}}$ over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. $\mathsf{PA}^{\mathsf{fs}}$ *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Conservativity of $\mathrm{PA}^{\mathrm{FS}}$ over PA

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of* PA *carries a full satisfaction class*.
- **Theorem**. $\mathrm{PA}^{\mathrm{fs}}$ *is conservative over* PA.
- The method used by Kotlarski et al. is based on the proof-theoretic machinery of $\mathcal{M}$-logic (an infinitary logic based on an ambient nonstandard model $\mathcal{M}$).
- $\mathcal{M}$-logic is not prima facie applicable to theories that are either weaker than PA, or to theories that are formulated in infinite languages.
- **Theorem** (Lachlan (1981)). *Every non-standard model of* PA *that carries a full satisfaction class is recursively saturated*.
- Since we can construct examples of non-standard models of PA that are not recursively saturated, we see that not every model of PA has an *expansion* with a full satisfaction class.

Universiteit Utrecht

The Core Construction 1

- **Core Lemma.** *Let $\mathcal{N}_0 \models B$ and suppose S_0 is an F_0-satisfaction class, where $F_0 \subseteq F_1 := \mathrm{Form}^{\mathcal{N}_0}$. Then, there is an elementary extension $\mathcal{N}_1$ of $\mathcal{N}_0$ that carries an F_1-satisfaction class S_1 such that $S_0 = S_1 \cap (F_0 \times N_0)$.*

- **Proof.** Let $\mathcal{L}_{\mathrm{B}}^{+}(\mathcal{N}_0)$ be the language obtained by enriching $\mathcal{L}_B$ with constant symbols for each member of N_0, and new *unary* predicates U_c for each $c \in \mathrm{Form}^{\mathcal{N}_0}$.

Universiteit Utrecht

The Core Construction 1

- **Core Lemma.** *Let $\mathcal{N}_0 \models B$ and suppose S_0 is an F_0-satisfaction class, where $F_0 \subseteq F_1 := \mathrm{Form}^{\mathcal{N}_0}$. Then, there is an elementary extension $\mathcal{N}_1$ of $\mathcal{N}_0$ that carries an F_1-satisfaction class S_1 such that $S_0 = S_1 \cap (F_0 \times N_0)$.*

- **Proof.** Let $\mathcal{L}_B^+(\mathcal{N}_0)$ be the language obtained by enriching $\mathcal{L}_B$ with constant symbols for each member of N_0, and new *unary* predicates U_c for each $c \in \mathrm{Form}^{\mathcal{N}_0}$.

Universiteit Utrecht

The Core Construction 1

- **Core Lemma.** *Let $\mathcal{N}_0 \models B$ and suppose S_0 is an F_0-satisfaction class, where $F_0 \subseteq F_1 := \mathrm{Form}^{\mathcal{N}_0}$. Then, there is an elementary extension $\mathcal{N}_1$ of $\mathcal{N}_0$ that carries an F_1-satisfaction class S_1 such that $S_0 = S_1 \cap (F_0 \times N_0)$.*

- **Proof.** Let $\mathcal{L}_{\mathrm{B}}^{+}(\mathcal{N}_0)$ be the language obtained by enriching $\mathcal{L}_B$ with constant symbols for each member of N_0, and new *unary* predicates U_c for each $c \in \mathrm{Form}^{\mathcal{N}_0}$.

Universiteit Utrecht

The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner$, then

$$\theta_c := \forall \alpha \left(\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).$$

- If $\mathcal{N} \models c = \ulcorner \neg d \urcorner$ then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge \neg \mathsf{U}_d(\alpha)).$$

- If $\mathcal{N} \models c = \ulcorner d_1 \vee d_2 \urcorner$, then $\theta_c :=$

$$\forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge (\mathsf{U}_{d_1}(\alpha \restriction \mathsf{FV}(d_1)) \vee \mathsf{U}_{d_2}(\alpha \restriction \mathsf{FV}(d_2)))).$$

- If $\mathcal{N} \models c = \ulcorner \exists v \; b \urcorner$, then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \; \mathsf{U}_b(\alpha')).$$

Universiteit Utrecht

The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner$, then

$$\theta_c := \forall \alpha \left(\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).$$

- If $\mathcal{N} \models c = \ulcorner \neg d \urcorner$ then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge \neg \mathsf{U}_d(\alpha)).$$

- If $\mathcal{N} \models c = \ulcorner d_1 \vee d_2 \urcorner$, then $\theta_c :=$

$$\forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge (\mathsf{U}_{d_1}(\alpha \restriction \mathsf{FV}(d_1)) \vee \mathsf{U}_{d_2}(\alpha \restriction \mathsf{FV}(d_2)))).$$

- If $\mathcal{N} \models c = \ulcorner \exists v \; b \urcorner$, then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \; \mathsf{U}_b(\alpha')).$$

Universiteit Utrecht

The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner$, then

$$\theta_c := \forall \alpha \left(\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).$$

- If $\mathcal{N} \models c = \ulcorner \neg d \urcorner$ then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge \neg \mathsf{U}_d(\alpha)).$$

- If $\mathcal{N} \models c = \ulcorner d_1 \vee d_2 \urcorner$, then $\theta_c :=$

$$\forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge (\mathsf{U}_{d_1}(\alpha \restriction \mathsf{FV}(d_1)) \vee \mathsf{U}_{d_2}(\alpha \restriction \mathsf{FV}(d_2)))).$$

- If $\mathcal{N} \models c = \ulcorner \exists v \; b \urcorner$, then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \; \mathsf{U}_b(\alpha')).$$

Universiteit Utrecht

The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner$, then

$$\theta_c := \forall \alpha \left(\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).$$

- If $\mathcal{N} \models c = \ulcorner \neg d \urcorner$ then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge \neg \mathsf{U}_d(\alpha)).$$

- If $\mathcal{N} \models c = \ulcorner d_1 \vee d_2 \urcorner$, then $\theta_c :=$

$$\forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge (\mathsf{U}_{d_1}(\alpha \restriction \mathsf{FV}(d_1)) \vee \mathsf{U}_{d_2}(\alpha \restriction \mathsf{FV}(d_2)))).$$

- If $\mathcal{N} \models c = \ulcorner \exists v \; b \urcorner$, then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \; \mathsf{U}_b(\alpha')).$$

Universiteit Utrecht

The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \ulcorner R(v_0, \cdots, v_{n-1}) \urcorner$, then

$$\theta_c := \forall \alpha \left(\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).$$

- If $\mathcal{N} \models c = \ulcorner \neg d \urcorner$ then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge \neg \mathsf{U}_d(\alpha)).$$

- If $\mathcal{N} \models c = \ulcorner d_1 \vee d_2 \urcorner$, then $\theta_c :=$

$$\forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \mathsf{asn}(\alpha, c) \wedge (\mathsf{U}_{d_1}(\alpha \restriction \mathsf{FV}(d_1)) \vee \mathsf{U}_{d_2}(\alpha \restriction \mathsf{FV}(d_2)))).$$

- If $\mathcal{N} \models c = \ulcorner \exists v \; b \urcorner$, then

$$\theta_c := \forall \alpha \, (\mathsf{U}_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \; \mathsf{U}_b(\alpha')).$$

Universiteit Utrecht

The Core Construction 3

- Let

$$\Gamma := \{\mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \in S_0\} \cup \{\neg\, \mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \notin S_0\}$$

  and define $\mathrm{Th}^+(\mathcal{N}_0) := \mathrm{Th}(\mathcal{N}_0, c)_{c \in N_0} \cup \Theta \cup \Gamma$.
- We show that $\mathrm{Th}^+(\mathcal{N}_0)$ is consistent by demonstrating that each finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ is interpretable in $(\mathcal{N}_0, S_0)$.
- Suppose T_0 is a finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ and let C consist of the c that appear in T_0. If C is empty everything is easy. So, we assume $C \neq \emptyset$.
- We construct subsets $\{U_c : c \in C\}$ of N_0 such that the following two conditions hold when U_c, is interpreted by U_c:
  1. $(\mathcal{N}_0, U_c)_{c \in C} \models \{\theta_c : c \in C\}$ and
  2. if $c \in C \cap F_0$, then $\alpha \in U_c$ iff $(c,\alpha) \in S_0$.

Universiteit Utrecht

The Core Construction 3

- Let

$$\Gamma := \{\mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \in S_0\} \cup \{\neg\, \mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \notin S_0\}$$

  and define $\mathrm{Th}^+(\mathcal{N}_0) := \mathrm{Th}(\mathcal{N}_0, c)_{c \in N_0} \cup \Theta \cup \Gamma$.
- We show that $\mathrm{Th}^+(\mathcal{N}_0)$ is consistent by demonstrating that each finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ is interpretable in $(\mathcal{N}_0, S_0)$.
- Suppose T_0 is a finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ and let C consist of the c that appear in T_0. If C is empty everything is easy. So, we assume $C \neq \emptyset$.
- We construct subsets $\{U_c : c \in C\}$ of N_0 such that the following two conditions hold when U_c, is interpreted by U_c:
  1. $(\mathcal{N}_0, U_c)_{c \in C} \models \{\theta_c : c \in C\}$ and
  2. if $c \in C \cap F_0$, then $\alpha \in U_c$ iff $(c,\alpha) \in S_0$.

Universiteit Utrecht

The Core Construction 3

- Let

$$\Gamma := \{\mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \in S_0\} \cup \{\neg\, \mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \notin S_0\}$$

  and define $\mathrm{Th}^+(\mathcal{N}_0) := \mathrm{Th}(\mathcal{N}_0, c)_{c \in N_0} \cup \Theta \cup \Gamma$.
- We show that $\mathrm{Th}^+(\mathcal{N}_0)$ is consistent by demonstrating that each finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ is interpretable in $(\mathcal{N}_0, S_0)$.
- Suppose T_0 is a finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ and let C consist of the c that appear in T_0. If C is empty everything is easy. So, we assume $C \neq \emptyset$.
- We construct subsets $\{U_c : c \in C\}$ of N_0 such that the following two conditions hold when U_c, is interpreted by U_c:
  1. $(\mathcal{N}_0, U_c)_{c \in C} \models \{\theta_c : c \in C\}$ and
  2. if $c \in C \cap F_0$, then $\alpha \in U_c$ iff $(c,\alpha) \in S_0$.

Universiteit Utrecht

The Core Construction 3

- Let

$$\Gamma := \{\mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \in S_0\} \cup \{\neg\, \mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \notin S_0\}$$

  and define $\mathrm{Th}^+(\mathcal{N}_0) := \mathrm{Th}(\mathcal{N}_0, c)_{c \in N_0} \cup \Theta \cup \Gamma$.
- We show that $\mathrm{Th}^+(\mathcal{N}_0)$ is consistent by demonstrating that each finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ is interpretable in $(\mathcal{N}_0, S_0)$.
- Suppose T_0 is a finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ and let C consist of the c that appear in T_0. If C is empty everything is easy. So, we assume $C \neq \emptyset$.
- We construct subsets $\{U_c : c \in C\}$ of N_0 such that the following two conditions hold when U_c, is interpreted by U_c:
  1. $(\mathcal{N}_0, U_c)_{c \in C} \models \{\theta_c : c \in C\}$ and
  2. if $c \in C \cap F_0$, then $\alpha \in U_c$ iff $(c,\alpha) \in S_0$.

Universiteit Utrecht

The Core Construction 3

- Let

$$\Gamma := \{\mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \in S_0\} \cup \{\neg\, \mathsf{U}_c(\alpha) : c \in F_0 \text{ and } (c,\alpha) \notin S_0\}$$

  and define $\mathrm{Th}^+(\mathcal{N}_0) := \mathrm{Th}(\mathcal{N}_0, c)_{c \in N_0} \cup \Theta \cup \Gamma$.
- We show that $\mathrm{Th}^+(\mathcal{N}_0)$ is consistent by demonstrating that each finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ is interpretable in $(\mathcal{N}_0, S_0)$.
- Suppose T_0 is a finite subset of $\mathrm{Th}^+(\mathcal{N}_0)$ and let C consist of the c that appear in T_0. If C is empty everything is easy. So, we assume $C \neq \emptyset$.
- We construct subsets $\{U_c : c \in C\}$ of N_0 such that the following two conditions hold when U_c, is interpreted by U_c:
  1. $(\mathcal{N}_0, U_c)_{c \in C} \models \{\theta_c : c \in C\}$ and
  2. if $c \in C \cap F_0$, then $\alpha \in U_c$ iff $(c,\alpha) \in S_0$.

Universiteit Utrecht

The Core Construction 4

- We shall construct $\{U_c : c \in C\}$ *in stages*, beginning with the simplest formulas in C, and working our way up using the Tarski rules for more complex ones.
- $c \lhd d$ expresses: c is a direct subformula of d.
- Define $\lhd^*$ on C by:

$$c \lhd^* d \text{ iff } (c \lhd d)^{\mathcal{N}_0} \text{ and } \theta_d \in T_0.$$

- The finiteness of C implies that $(C, \lhd^*)$ is well-founded, which in turn helps us define a useful measure of complexity for $c \in C$ using the following recursive definition:

$$\mathrm{rank}_C(x) := \sup\{\mathrm{rank}_C(y) + 1 : x \in C \text{ and } (y \lhd^* x)^{\mathcal{N}_0}\}.$$

Universiteit Utrecht

The Core Construction 4

- We shall construct $\{U_c : c \in C\}$ *in stages*, beginning with the simplest formulas in C, and working our way up using the Tarski rules for more complex ones.
- $c \lhd d$ expresses: c is a direct subformula of d.
- Define $\lhd^*$ on C by:

$$c \lhd^* d \text{ iff } (c \lhd d)^{\mathcal{N}_0} \text{ and } \theta_d \in T_0.$$

- The finiteness of C implies that $(C, \lhd^*)$ is well-founded, which in turn helps us define a useful measure of complexity for $c \in C$ using the following recursive definition:

$$\mathrm{rank}_C(x) := \sup\{\mathrm{rank}_C(y) + 1 : x \in C \text{ and } (y \lhd^* x)^{\mathcal{N}_0}\}.$$

Universiteit Utrecht

The Core Construction 4

- We shall construct $\{U_c : c \in C\}$ *in stages*, beginning with the simplest formulas in C, and working our way up using the Tarski rules for more complex ones.
- $c \lhd d$ expresses: c is a direct subformula of d.
- Define $\lhd^*$ on C by:

$$c \lhd^* d \text{ iff } (c \lhd d)^{\mathcal{N}_0} \text{ and } \theta_d \in T_0.$$

- The finiteness of C implies that $(C, \lhd^*)$ is well-founded, which in turn helps us define a useful measure of complexity for $c \in C$ using the following recursive definition:

$$\mathrm{rank}_C(x) := \sup\{\mathrm{rank}_C(y) + 1 : x \in C \text{ and } (y \lhd^* x)^{\mathcal{N}_0}\}.$$

Universiteit Utrecht

The Core Construction 4

- We shall construct $\{U_c : c \in C\}$ *in stages*, beginning with the simplest formulas in C, and working our way up using the Tarski rules for more complex ones.
- $c \lhd d$ expresses: c is a direct subformula of d.
- Define $\lhd^*$ on C by:

$$c \lhd^* d \text{ iff } (c \lhd d)^{\mathcal{N}_0} \text{ and } \theta_d \in T_0.$$

- The finiteness of C implies that $(C, \lhd^*)$ is well-founded, which in turn helps us define a useful measure of complexity for $c \in C$ using the following recursive definition:

$$\mathrm{rank}_C(x) := \sup\{\mathrm{rank}_C(y) + 1 : x \in C \text{ and } (y \lhd^* x)^{\mathcal{N}_0}\}.$$

Universiteit Utrecht

The Core Construction 4

- We shall construct $\{U_c : c \in C\}$ *in stages*, beginning with the simplest formulas in C, and working our way up using the Tarski rules for more complex ones.
- $c \lhd d$ expresses: c is a direct subformula of d.
- Define $\lhd^*$ on C by:

$$c \lhd^* d \text{ iff } (c \lhd d)^{\mathcal{N}_0} \text{ and } \theta_d \in T_0.$$

- The finiteness of C implies that $(C, \lhd^*)$ is well-founded, which in turn helps us define a useful measure of complexity for $c \in C$ using the following recursive definition:

$$\mathrm{rank}_C(x) := \sup\{\mathrm{rank}_C(y) + 1 : x \in C \text{ and } (y \lhd^* x)^{\mathcal{N}_0}\}.$$

Universiteit Utrecht

The Core Construction 5

- Note that $\mathrm{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \lhd^* c)^{\mathcal{N}_0}$.
- Next, let $C_i := \{x \in C : \mathrm{rank}_C(x)\} \leq i\}$.
- Since C is finite and non-empty, we have $C_0 \neq \emptyset$.
- $c \in C_0$ iff ($c \in C$ and C does not contain the code of any subformula of the formula coded by c).
- If $c \in C_{i+1}$, then the codes of every immediate subformula of the formula coded by c are in C_i. This observation ensures that the following recursive clauses yield a well-defined U_c for each $c \in C$.

Universiteit Utrecht

The Core Construction 5

- Note that $\mathrm{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \lhd^* c)^{\mathcal{N}_0}$.
- Next, let $C_i := \{x \in C : \mathrm{rank}_C(x)\} \leq i\}$.
- Since C is finite and non-empty, we have $C_0 \neq \emptyset$.
- $c \in C_0$ iff ($c \in C$ and C does not contain the code of any subformula of the formula coded by c).
- If $c \in C_{i+1}$, then the codes of every immediate subformula of the formula coded by c are in C_i. This observation ensures that the following recursive clauses yield a well-defined U_c for each $c \in C$.

Universiteit Utrecht

The Core Construction 5

- Note that $\mathrm{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \lhd^* c)^{\mathcal{N}_0}$.
- Next, let $C_i := \{x \in C : \mathrm{rank}_C(x)\} \leq i\}$.
- Since C is finite and non-empty, we have $C_0 \neq \emptyset$.
- $c \in C_0$ iff ($c \in C$ and C does not contain the code of any subformula of the formula coded by c).
- If $c \in C_{i+1}$, then the codes of every immediate subformula of the formula coded by c are in C_i. This observation ensures that the following recursive clauses yield a well-defined U_c for each $c \in C$.

Universiteit Utrecht

The Core Construction 5

- Note that $\mathrm{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \lhd^* c)^{\mathcal{N}_0}$.
- Next, let $C_i := \{x \in C : \mathrm{rank}_C(x)\} \leq i\}$.
- Since C is finite and non-empty, we have $C_0 \neq \emptyset$.
- $c \in C_0$ iff ($c \in C$ and C does not contain the code of any subformula of the formula coded by c).
- If $c \in C_{i+1}$, then the codes of every immediate subformula of the formula coded by c are in C_i. This observation ensures that the following recursive clauses yield a well-defined U_c for each $c \in C$.

Universiteit Utrecht

The Core Construction 5

- Note that $\mathrm{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \lhd^* c)^{\mathcal{N}_0}$.
- Next, let $C_i := \{x \in C : \mathrm{rank}_C(x)\} \leq i\}$.
- Since C is finite and non-empty, we have $C_0 \neq \emptyset$.
- $c \in C_0$ iff ($c \in C$ and C does not contain the code of any subformula of the formula coded by c).
- If $c \in C_{i+1}$, then the codes of every immediate subformula of the formula coded by c are in C_i. This observation ensures that the following recursive clauses yield a well-defined U_c for each $c \in C$.

Universiteit Utrecht

The Core Construction 5

- Note that $\mathrm{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \lhd^* c)^{\mathcal{N}_0}$.
- Next, let $C_i := \{x \in C : \mathrm{rank}_C(x)\} \leq i\}$.
- Since C is finite and non-empty, we have $C_0 \neq \emptyset$.
- $c \in C_0$ iff ($c \in C$ and C does not contain the code of any subformula of the formula coded by c).
- If $c \in C_{i+1}$, then the codes of every immediate subformula of the formula coded by c are in C_i. This observation ensures that the following recursive clauses yield a well-defined U_c for each $c \in C$.

Universiteit Utrecht

The Core Construction 6

- If $c \in C_0$ then $U_c := \begin{cases} \{\alpha : (c, \alpha) \in S_0\}, \text{ if } c \in F_0; \\ U_c := \emptyset, \text{ if } c \notin F_0. \end{cases}$
- If $c \in C_{i+1} \setminus C_i$ and $\ulcorner c = \neg d \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \notin U_d\}.$$

- If $c \in C_{i+1} \setminus C_i$ and $c = \ulcorner a \vee b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \restriction \mathsf{fv}(a) \in U_a \text{ or } \alpha \restriction \mathsf{fv}(b) \in U_b\}.$$

- If $c \in C_{i+1} \setminus C_i$ and $c = \ulcorner \exists v\ b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \exists \alpha' \supseteq \alpha\ \alpha' \in U_b\}.$$

Universiteit Utrecht

The Core Construction 6

- If $c \in C_0$ then $U_c := \begin{cases} \{\alpha : (c, \alpha) \in S_0\}, \text{ if } c \in F_0; \\ U_c := \emptyset, \text{ if } c \notin F_0. \end{cases}$
- If $c \in C_{i+1} \setminus C_i$ and $\ulcorner c = \neg d \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \notin U_d\}.$$

- If $c \in C_{i+1} \setminus C_i$ and $c = \ulcorner a \vee b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \restriction \mathsf{fv}(a) \in U_a \text{ or } \alpha \restriction \mathsf{fv}(b) \in U_b\}.$$

- If $c \in C_{i+1} \setminus C_i$ and $c = \ulcorner \exists v\ b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \exists \alpha' \supseteq \alpha\ \alpha' \in U_b\}.$$

Universiteit Utrecht

The Core Construction 6

- If $c \in C_0$ then $U_c := \begin{cases} \{\alpha : (c, \alpha) \in S_0\}, \text{ if } c \in F_0; \\ U_c := \emptyset, \text{ if } c \notin F_0. \end{cases}$
- If $c \in C_{i+1} \backslash C_i$ and $\ulcorner c = \neg d \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \notin U_d\}.$$

- If $c \in C_{i+1} \backslash C_i$ and $c = \ulcorner a \vee b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \restriction \mathsf{fv}(a) \in U_a \text{ or } \alpha \restriction \mathsf{fv}(b) \in U_b\}.$$

- If $c \in C_{i+1} \backslash C_i$ and $c = \ulcorner \exists v\ b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \exists \alpha' \supseteq \alpha\ \alpha' \in U_b\}.$$

Universiteit Utrecht

The Core Construction 6

- If $c \in C_0$ then $U_c := \begin{cases} \{\alpha : (c, \alpha) \in S_0\}, \text{ if } c \in F_0; \\ U_c := \emptyset, \text{ if } c \notin F_0. \end{cases}$
- If $c \in C_{i+1} \backslash C_i$ and $\ulcorner c = \neg d \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \notin U_d\}.$$

- If $c \in C_{i+1} \backslash C_i$ and $c = \ulcorner a \vee b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \restriction \mathsf{fv}(a) \in U_a \text{ or } \alpha \restriction \mathsf{fv}(b) \in U_b\}.$$

- If $c \in C_{i+1} \backslash C_i$ and $c = \ulcorner \exists v\ b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \exists \alpha' \supseteq \alpha\ \alpha' \in U_b\}.$$

Universiteit Utrecht

The Core Construction 6

- If $c \in C_0$ then $U_c := \begin{cases} \{\alpha : (c, \alpha) \in S_0\}, \text{ if } c \in F_0; \\ U_c := \emptyset, \text{ if } c \notin F_0. \end{cases}$
- If $c \in C_{i+1} \backslash C_i$ and $\ulcorner c = \neg d \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \notin U_d\}.$$

- If $c \in C_{i+1} \backslash C_i$ and $c = \ulcorner a \vee b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \alpha \restriction \mathsf{fv}(a) \in U_a \text{ or } \alpha \restriction \mathsf{fv}(b) \in U_b\}.$$

- If $c \in C_{i+1} \backslash C_i$ and $c = \ulcorner \exists v\ b \urcorner$, then

$$U_c := \{\alpha \in \mathsf{asn}_c : \exists \alpha' \supseteq \alpha\ \alpha' \in U_b\}.$$

Universiteit Utrecht

The Core Construction 7

- **Core Theorem.** *Let $\mathcal{M}_0$ be a model of B of any cardinality. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that admits a full satisfaction class.*

- **Proof:** Let F_0 be the set of *atomic* $\mathcal{N}$-formulas and let S_0 be the obvious satisfaction predicate for F_0. Then by the Lemma there is an elementary extension $\mathcal{M}_1$ of $\mathcal{M}_0$ that carries a $\mathrm{Form}_B^{\mathcal{M}_0}$ satisfaction class. Thanks to the Core Lemma, this argument can be carried out countably many times to yield two sequences $\langle \mathcal{M}_i : i \in \omega \rangle$ and $\langle S_i : i \in \omega \rangle$ that satisfy the following two properties:
  1. $\mathcal{M}_i \prec \mathcal{M}_{i+1}$;
  2. S_{i+1} is a $\mathrm{Form}^{\mathcal{M}_i}$-satisfaction class on $\mathcal{M}_{i+1}$ for each $i \in \omega$.

- We take: $\mathcal{M} := \bigcup_{i \in \omega} \mathcal{M}_i$, and $S := \bigcup_{i \in \omega} S_i$.

Universiteit Utrecht

The Core Construction 7

- **Core Theorem.** *Let $\mathcal{M}_0$ be a model of B of any cardinality. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that admits a full satisfaction class.*

- **Proof:** Let F_0 be the set of *atomic* $\mathcal{N}$-formulas and let S_0 be the obvious satisfaction predicate for F_0. Then by the Lemma there is an elementary extension $\mathcal{M}_1$ of $\mathcal{M}_0$ that carries a $\mathrm{Form}_B^{\mathcal{M}_0}$ satisfaction class. Thanks to the Core Lemma, this argument can be carried out countably many times to yield two sequences $\langle \mathcal{M}_i : i \in \omega \rangle$ and $\langle S_i : i \in \omega \rangle$ that satisfy the following two properties:
  1. $\mathcal{M}_i \prec \mathcal{M}_{i+1}$;
  2. S_{i+1} is a $\mathrm{Form}^{\mathcal{M}_i}$-satisfaction class on $\mathcal{M}_{i+1}$ for each $i \in \omega$.

- We take: $\mathcal{M} := \bigcup_{i \in \omega} \mathcal{M}_i$, and $S := \bigcup_{i \in \omega} S_i$.

Universiteit Utrecht

The Core Construction 7

- **Core Theorem.** *Let $\mathcal{M}_0$ be a model of B of any cardinality. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that admits a full satisfaction class.*

- **Proof:** Let F_0 be the set of *atomic* $\mathcal{N}$-formulas and let S_0 be the obvious satisfaction predicate for F_0. Then by the Lemma there is an elementary extension $\mathcal{M}_1$ of $\mathcal{M}_0$ that carries a $\mathrm{Form}_B^{\mathcal{M}_0}$ satisfaction class. Thanks to the Core Lemma, this argument can be carried out countably many times to yield two sequences $\langle \mathcal{M}_i : i \in \omega \rangle$ and $\langle S_i : i \in \omega \rangle$ that satisfy the following two properties:
  1. $\mathcal{M}_i \prec \mathcal{M}_{i+1}$;
  2. S_{i+1} is a $\mathrm{Form}^{\mathcal{M}_i}$-satisfaction class on $\mathcal{M}_{i+1}$ for each $i \in \omega$.

- We take: $\mathcal{M} := \bigcup_{i \in \omega} \mathcal{M}_i$, and $S := \bigcup_{i \in \omega} S_i$.

Universiteit Utrecht

The Core Construction 7

- **Core Theorem.** *Let $\mathcal{M}_0$ be a model of B of any cardinality. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that admits a full satisfaction class.*

- **Proof:** Let F_0 be the set of *atomic* $\mathcal{N}$-formulas and let S_0 be the obvious satisfaction predicate for F_0. Then by the Lemma there is an elementary extension $\mathcal{M}_1$ of $\mathcal{M}_0$ that carries a $\mathrm{Form}_B^{\mathcal{M}_0}$ satisfaction class. Thanks to the Core Lemma, this argument can be carried out countably many times to yield two sequences $\langle \mathcal{M}_i : i \in \omega \rangle$ and $\langle S_i : i \in \omega \rangle$ that satisfy the following two properties:
  1. $\mathcal{M}_i \prec \mathcal{M}_{i+1}$;
  2. S_{i+1} is a $\mathrm{Form}^{\mathcal{M}_i}$-satisfaction class on $\mathcal{M}_{i+1}$ for each $i \in \omega$.

- We take: $\mathcal{M} := \bigcup_{i \in \omega} \mathcal{M}_i$, and $S := \bigcup_{i \in \omega} S_i$.

Universiteit Utrecht

The Core Construction 8

- **Corollary.** *B^{fs} is a conservative extension of B, for every base theory B.*

Universiteit Utrecht

The Core Construction 8

- **Corollary.** *B^{fs} is a conservative extension of B, for every base theory B.*

Universiteit Utrecht

The Arithmetization of the Core Construction

- **Theorem.** *The following statement* $(*)$ *is provable within* WKL_0 :

  $(*)$ *Every consistent base theory B has a model* $\mathcal{M}$ that carries a full satisfaction class S and which has the property that the Tarskian satisfaction relation of $(\mathcal{M}, S)$ is coded by some $X \subseteq \omega$.

- **Theorem.** $\mathsf{PRA} \vdash$ *"B^{fs} is conservative over B" for every r.e. base theory B.*

Universiteit Utrecht

The Arithmetization of the Core Construction

- **Theorem.** *The following statement* $(*)$ *is provable within* WKL_0 :

  $(*)$ *Every consistent base theory B has a model* $\mathcal{M}$ that carries a full satisfaction class S and which has the property that the Tarskian satisfaction relation of $(\mathcal{M}, S)$ is coded by some $X \subseteq \omega$.

- **Theorem.** $\mathsf{PRA} \vdash$ *"B^{fs} is conservative over B" for every r.e. base theory B.*

Universiteit Utrecht

The Arithmetization of the Core Construction

- **Theorem.** *The following statement* $(*)$ *is provable within* WKL_0 :

  $(*)$ *Every consistent base theory B has a model* $\mathcal{M}$ that carries a full satisfaction class S and which has the property that the Tarskian satisfaction relation of $(\mathcal{M}, S)$ is coded by some $X \subseteq \omega$.

- **Theorem.** $\mathsf{PRA} \vdash$ *"B^{fs} is conservative over B" for every r.e. base theory B.*

Universiteit Utrecht

Pathological Satisfaction Classes

- **Definition.** For any standard formula σ of $\mathcal{L}_B$, and for each $a \in \mathbb{N}^{\mathcal{M}}$, where $\mathcal{M}$ is a model of B, the 'formula' σ_a is defined by internal recursion in $\mathcal{M}_0$ via $\sigma_0 := \sigma$; and $\sigma_{n+1} := \sigma_n \vee \sigma_n$.
- **Theorem.** *Let $\sigma := \exists v_0 \, (v_0 = v_0)$ and let $\mathcal{M}_0$ be a model of B of any cardinality. Then, $\mathcal{M}_0$ has an elementary extension $\mathcal{M}$* that carries a full satisfaction class S such that

$$\{a \in \mathbb{N}^{\mathcal{M}} : \sigma_a \text{ is } S\text{-valid}\} = \omega.$$

Universiteit Utrecht

Pathological Satisfaction Classes

- **Definition.** For any standard formula σ of $\mathcal{L}_B$, and for each $a \in \mathbb{N}^{\mathcal{M}}$, where $\mathcal{M}$ is a model of B, the 'formula' σ_a is defined by internal recursion in $\mathcal{M}_0$ via $\sigma_0 := \sigma$; and $\sigma_{n+1} := \sigma_n \vee \sigma_n$.
- **Theorem.** *Let $\sigma := \exists v_0 \, (v_0 = v_0)$ and let $\mathcal{M}_0$ be a model of B of any cardinality. Then, $\mathcal{M}_0$ has an elementary extension $\mathcal{M}$* that carries a full satisfaction class S such that

$$\{a \in \mathbb{N}^{\mathcal{M}} : \sigma_a \text{ is } S\text{-valid}\} = \omega.$$

Universiteit Utrecht

Pathological Satisfaction Classes

- **Definition.** For any standard formula σ of $\mathcal{L}_B$, and for each $a \in \mathbb{N}^{\mathcal{M}}$, where $\mathcal{M}$ is a model of B, the 'formula' σ_a is defined by internal recursion in $\mathcal{M}_0$ via $\sigma_0 := \sigma$; and $\sigma_{n+1} := \sigma_n \vee \sigma_n$.
- **Theorem.** *Let* $\sigma := \exists v_0 \, (v_0 = v_0)$ *and let* $\mathcal{M}_0$ *be a model of* B *of any cardinality. Then,* $\mathcal{M}_0$ *has an elementary extension* $\mathcal{M}$ that carries a full satisfaction class S such that

$$\{a \in \mathbb{N}^{\mathcal{M}} : \sigma_a \text{ is } S\text{-valid}\} = \omega.$$

Universiteit Utrecht

Desirable Satisfaction Classes

- **Theorem.** *Let $\mathcal{M}_0 \models B$, where B is a base theory. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that carries full satisfaction classes S_1, S_2, and S_3 such that:*
  a. *S_1 is schematically correct;*
  b. *S_2 is both existentially and disjunctively correct;*
  c. *S_3 is both extensional and alphabetically correct.*

- *Moreover, if B is an inductive base theory, then $\mathcal{M}$ carries a full satisfaction class S_4 such that: S_4 is $\Sigma_{B,\infty}$-correct.*

Universiteit Utrecht

Desirable Satisfaction Classes

- **Theorem.** *Let $\mathcal{M}_0 \models B$, where B is a base theory. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that carries full satisfaction classes S_1, S_2, and S_3 such that:*
  a. *S_1 is schematically correct*;
  b. *S_2 is both existentially and disjunctively correct*;
  c. *S_3 is both extensional and alphabetically correct.*

- *Moreover, if B is an inductive base theory, then $\mathcal{M}$ carries a full satisfaction class S_4 such that: S_4 is $\Sigma_{B,\infty}$-correct.*

Universiteit Utrecht

Desirable Satisfaction Classes

- **Theorem.** *Let $\mathcal{M}_0 \models B$, where B is a base theory. There is an elementary extension $\mathcal{M}$ of $\mathcal{M}_0$ that carries full satisfaction classes S_1, S_2, and S_3 such that:*
  a. *S_1 is schematically correct*;
  b. *S_2 is both existentially and disjunctively correct*;
  c. *S_3 is both extensional and alphabetically correct.*

- *Moreover, if B is an inductive base theory, then $\mathcal{M}$ carries a full satisfaction class S_4 such that: S_4 is $\Sigma_{B,\infty}$-correct.*

Universiteit Utrecht

Interpretability Issues 1

- Let ACA be the strengthening of ACA_0 with the full scheme of induction. It has been long known that ACA and $\mathsf{PA}^{\mathsf{fis}}$ are 'proof-theoretically equivalent'. The result below provides a more precise relationship between the two theories.
- **Theorem.** *There is a sentence σ in the language of ACA_0 such that $\mathsf{PA}^{\mathsf{fis}}$ and $\mathsf{ACA} + \sigma$ are bi-interpretable.*
- **Theorem.** *B^{is} and B^{fs} are both interpretable in B for every inductive recursively axiomatizable base theory B.*

Universiteit Utrecht

Interpretability Issues 1

- Let ACA be the strengthening of ACA_0 with the full scheme of induction. It has been long known that ACA and $\mathsf{PA}^{\mathsf{fis}}$ are 'proof-theoretically equivalent'. The result below provides a more precise relationship between the two theories.
- **Theorem.** *There is a sentence σ in the language of ACA_0 such that $\mathsf{PA}^{\mathsf{fis}}$ and $\mathsf{ACA} + \sigma$ are bi-interpretable.*
- **Theorem.** *B^{is} and B^{fs} are both interpretable in B for every inductive recursively axiomatizable base theory B.*

Universiteit Utrecht

Interpretability Issues 1

- Let ACA be the strengthening of ACA_0 with the full scheme of induction. It has been long known that ACA and $\mathsf{PA}^{\mathsf{fis}}$ are 'proof-theoretically equivalent'. The result below provides a more precise relationship between the two theories.
- **Theorem.** *There is a sentence σ in the language of* ACA_0 *such that* $\mathsf{PA}^{\mathsf{fis}}$ *and* $\mathsf{ACA} + \sigma$ *are bi-interpretable.*
- **Theorem.** *B^{is} and B^{fs} are both interpretable in* B *for every inductive recursively axiomatizable base theory B.*

Universiteit Utrecht

Interpretability Issues 1

- Let ACA be the strengthening of ACA_0 with the full scheme of induction. It has been long known that ACA and $\mathsf{PA}^{\mathsf{fis}}$ are 'proof-theoretically equivalent'. The result below provides a more precise relationship between the two theories.
- **Theorem.** *There is a sentence σ in the language of* ACA_0 *such that* $\mathsf{PA}^{\mathsf{fis}}$ *and* $\mathsf{ACA} + \sigma$ *are bi-interpretable.*
- **Theorem.** B^{is} *and* B^{fs} *are both interpretable in* B *for every inductive recursively axiomatizable base theory B.*

Universiteit Utrecht

Interpretability Issues 2

- **Theorem** (Interpretability among PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$, and ACA_0).
  a. *The theories* PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$ *are mutually interpretable.*
  b. *Each of the theories* PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$ *is interpretable in* ACA_0*, but none of them interprets* ACA_0.
  c. *No pair of the theories* PA, $\mathsf{PA}^{\mathsf{fs}}$, $\mathsf{PA}^{\mathsf{is}}$, ACA_0 *are bi-interpretable.*
- *If B is a consistent finitely axiomatizable base theory, then neither* B^{is} *nor* B^{fs} *is interpretable in B.*

Universiteit Utrecht

Interpretability Issues 2

- **Theorem** (Interpretability among PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$, and ACA_0).
  a. *The theories* PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$ *are mutually interpretable.*
  b. *Each of the theories* PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$ *is interpretable in* ACA_0*, but none of them interprets* ACA_0.
  c. *No pair of the theories* PA, $\mathsf{PA}^{\mathsf{fs}}$, $\mathsf{PA}^{\mathsf{is}}$, ACA_0 *are bi-interpretable.*
- *If B is a consistent finitely axiomatizable base theory, then neither* B^{is} *nor* B^{fs} *is interpretable in B.*

Universiteit Utrecht

Interpretability Issues 2

- **Theorem** (Interpretability among PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$, and ACA_0).
  a. *The theories* PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$ *are mutually interpretable.*
  b. *Each of the theories* PA, $\mathsf{PA}^{\mathsf{is}}$, $\mathsf{PA}^{\mathsf{fs}}$ *is interpretable in* ACA_0*, but none of them interprets* ACA_0.
  c. *No pair of the theories* PA, $\mathsf{PA}^{\mathsf{fs}}$, $\mathsf{PA}^{\mathsf{is}}$, ACA_0 *are bi-interpretable.*
- *If B is a consistent finitely axiomatizable base theory, then neither* B^{is} *nor* B^{fs} *is interpretable in B.*

Universiteit Utrecht

Some Open Questions

- Can the conservativity of PA^{fs} over PA be verified in EFA (Exponential Function Arithmetic)?
- Is PA^{fs} + "S contains all propositional tautologies" conservative over PA?
  Cezary Cieśliński has shown that PA^{fs} + "S is closed under derivability in propositional logic" is *not* conservative over PA

Universiteit Utrecht

Some Open Questions

- Can the conservativity of PA^{fs} over PA be verified in EFA (Exponential Function Arithmetic)?
- Is PA^{fs} + "S contains all propositional tautologies" conservative over PA?
  Cezary Cieśliński has shown that PA^{fs} + "S is closed under derivability in propositional logic" is *not* conservative over PA

Universiteit Utrecht

Some Open Questions

- Can the conservativity of PA^{fs} over PA be verified in EFA (Exponential Function Arithmetic)?
- Is PA^{fs} + "S contains all propositional tautologies" conservative over PA?
  Cezary Cieśliński has shown that PA^{fs} + "S is closed under derivability in propositional logic" is *not* conservative over PA

Universiteit Utrecht