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

Ali Enayat & Albert Visser

Amsterdam Workshop on Truth

March 15, 2013
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.
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 $M$ of $U$ there is an elementary extension $M^+$ satisfying $U^+$. 
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 $M$ of $U$ there is an elementary extension $M^+$ satisfying $U^+$. 

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 $U^+$. 
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 $M$ of $U$ there is an elementary extension $M^+$ satisfying $U^+$. 


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.

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.

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.

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.

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.
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 \textit{sequential}, i.e., if we can define a binary relation $\in$ satisfying Adjunctive Set Theory AS which is given by the following axioms:

\begin{enumerate}
  \item there is an empty set
  \item for any $x$ and $y$ there is some $z$ whose elements are precisely $x \cup \{y\}$ (adjunction)
\end{enumerate}

We do not demand extensionality.
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.
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):

$tarski_0(S, F) := (F(x) \rightarrow \text{Form}(x)) \land (S(x, \alpha) \rightarrow (F(x) \land \text{asn}(\alpha, x))) \land ((y \triangleleft x \land F(x)) \rightarrow F(y)).$

$tarski_1,R(S, F) :=$

$\left(F(x) \land (x = \neg R(v_0, \cdots, v_{n-1}) \land \text{asn}(\alpha, x)) \rightarrow \right.$

$(S(x, \alpha) \leftrightarrow R([\alpha]v_0 \cdots, [\alpha]v_{n-1})) \right).$
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($S,F$):

$tarski_0(S,F) := (F(x) \rightarrow \text{Form}(x)) \land (S(x,\alpha) \rightarrow (F(x) \land \text{asn}(\alpha,x))) \land ((y \triangleleft x \land F(x)) \rightarrow F(y)).$

$tarski_1,R(S,F) :=$

$\left( (F(x) \land (x = \text{``}R(v_0,\cdots,v_{n-1})\text{''}) \land \text{asn}(\alpha,x)) \rightarrow (S(x,\alpha) \leftrightarrow R([\alpha]v_0\cdots,[\alpha]v_{n-1})) \right).$
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($S, F$):
  - $\text{tarski}_0(S, F) := (F(x) \rightarrow \text{Form}(x)) \land (S(x, \alpha) \rightarrow (F(x) \land \text{asn}(\alpha, x))) \land ((y \triangleleft x \land F(x)) \rightarrow F(y)).$
  - $\text{tarski}_{1,R}(S, F) := (F(x) \land (x = \lceil R(v_0, \ldots, v_{n-1}) \rceil) \land \text{asn}(\alpha, x)) \rightarrow (S(x, \alpha) \leftrightarrow R([\alpha]_{v_0} \cdots, [\alpha]_{v_{n-1}})).$
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($S, F$):

- $\text{tarski}_0(S, F) := (F(x) \rightarrow \text{Form}(x)) \land (S(x, \alpha) \rightarrow (F(x) \land \text{asn}(\alpha, x))) \land ((y \triangleleft x \land F(x)) \rightarrow F(y))$.

- $\text{tarski}_{1,R}(S, F) := (F(x) \land (x \models R(v_0, \cdots, v_{n-1})) \land \text{asn}(\alpha, x)) \rightarrow (S(x, \alpha) \iff R([\alpha]v_0, \cdots, [\alpha]v_{n-1}))$.
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($S, F$):

\begin{itemize}
\item $\text{tarski}_0(S, F) := (F(x) \rightarrow \text{Form}(x)) \land (S(x, \alpha) \rightarrow (F(x) \land \text{asn}(\alpha, x))) \land ((y \rem x \land F(x)) \rightarrow F(y))$.
\item $\text{tarski}_{1,R}(S, F) :=$
\begin{align*}
(F(x) \land (x = \upharpoonright R(v_0, \ldots, v_{n-1})^{-1}) \land \text{asn}(\alpha, x)) \rightarrow \\
(S(x, \alpha) \leftrightarrow R([\alpha]v_0 \cdot \ldots, [\alpha]v_{n-1})).
\end{align*}
\end{itemize}
Axioms for Satisfaction 2

- $\text{tarski}_2(S, F) :=$
  \[
  (F(x) \land (x = \neg y) \land \text{asn}(\alpha, x)) \rightarrow (S(x, \alpha) \leftrightarrow \neg S(y, \alpha)).
  \]

- $\text{tarski}_3(S, F) :=$
  \[
  (F(x) \land (x = y_1 \lor y_2) \land \text{asn}(\alpha, x)) \rightarrow \\
  \left( S(x, \alpha) \leftrightarrow \left( S(y_1, \alpha \upharpoonright \text{fv}(y_1)) \lor S(y_2, \alpha \upharpoonright \text{fv}(y_2)) \right) \right).
  \]

- $\text{tarski}_4(S, F) :=$
  \[
  (F(x) \land (x = \exists v y) \land \text{asn}(\alpha, x)) \rightarrow \\
  \left( S(x, \alpha) \leftrightarrow \exists \alpha' \supseteq \alpha S(y, \alpha') \right).
  \]
**Axioms for Satisfaction 2**

- \( \text{tarski}_2(S, F) := \)
  \[
  \left( F(x) \land (x = \neg y) \land \text{asn}(\alpha, x) \right) \rightarrow (S(x, \alpha) \leftrightarrow \neg S(y, \alpha)).
  \]

- \( \text{tarski}_3(S, F) := \)
  \[
  \left( F(x) \land (x = y_1 \lor y_2) \land \text{asn}(\alpha, x) \right) \rightarrow
  \left( S(x, \alpha) \leftrightarrow \left( S(y_1, \alpha \upharpoonright \text{fv}(y_1)) \lor S(y_2, \alpha \upharpoonright \text{fv}(y_2)) \right) \right).
  \]

- \( \text{tarski}_4(S, F) := \)
  \[
  \left( F(x) \land (x = \exists v y) \land \text{asn}(\alpha, x) \right) \rightarrow
  \left( S(x, \alpha) \leftrightarrow \exists \alpha' \supseteq \alpha S(y, \alpha') \right).
  \]
Axioms for Satisfaction 2

- \( \text{tarski}_2(S, F) := \)
  \[
  \left( F(x) \land (x = \neg y) \land \right. \\
  \left. \text{asn}(\alpha, x) \right) \rightarrow (S(x, \alpha) \leftrightarrow \neg S(y, \alpha)).
  \]

- \( \text{tarski}_3(S, F) := \)
  \[
  \left( F(x) \land (x = y_1 \lor y_2) \land \right. \\
  \left. \text{asn}(\alpha, x) \right) \rightarrow \\
  \left( S(x, \alpha) \leftrightarrow \left( S(y_1, \alpha \upharpoonright \text{fv}(y_1)) \lor \\ S(y_2, \alpha \upharpoonright \text{fv}(y_2)) \right) \right).
  \]

- \( \text{tarski}_4(S, F) := \)
  \[
  \left( F(x) \land (x = \exists v y) \land \right. \\
  \left. \text{asn}(\alpha, x) \right) \rightarrow \\
  \left( S(x, \alpha) \leftrightarrow \left( \exists \alpha' \supseteq \alpha S(y, \alpha') \right) \right).
  \]
Axioms for Satisfaction 2

\[ \text{tarski}_2(S, F) := \left( F(x) \land (x = \neg y) \land \text{asn}(\alpha, x) \right) \rightarrow (S(x, \alpha) \leftrightarrow \neg S(y, \alpha)). \]

\[ \text{tarski}_3(S, F) := \left( F(x) \land (x = y_1 \lor y_2) \land \text{asn}(\alpha, x) \right) \rightarrow \left( S(x, \alpha) \leftrightarrow \left( S(y_1, \alpha \upharpoonright \text{fv}(y_1)) \lor S(y_2, \alpha \upharpoonright \text{fv}(y_2)) \right) \right). \]

\[ \text{tarski}_4(S, F) := \left( F(x) \land (x = \exists y) \land \text{asn}(\alpha, x) \right) \rightarrow \left( S(x, \alpha) \leftrightarrow \left( \exists \alpha' \supseteq \alpha S(y, \alpha') \right) \right). \]
Extensions of the Base Theory

- $B^\text{fs} := B \cup \text{Tarski (S, form)}$.

- $B^\text{is} := B \cup \bigcup \{\text{Tarski (S, form}_n\} : n \in \omega\} \cup \text{ind(S)}$,
  where form$_n$ is the collection of formulas $\mathcal{L}_B$ with quantifier alternation depth at most $n$ and ind(S) is the full scheme of induction on the designated numbers in the language $\mathcal{L}_B(S)$.

- $B^\text{fis} := B^\text{fs} \cup B^\text{is}$. 
Extensions of the Base Theory

- \( B^{\text{fs}} := B \cup \text{Tarski}(S, \text{form}) \).  

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

where \( \text{form}_n \) is the collection of formulas \( \mathcal{L}_B \) with quantifier alternation depth at most \( n \) and \( \text{ind}(S) \) is the full scheme of induction on the designated numbers in the language \( \mathcal{L}_B(S) \).

- \( B^{\text{fis}} := B^{\text{fs}} \cup B^{\text{is}} \).
Extensions of the Base Theory

\begin{itemize}
  \item \( B^{fs} := B \cup \text{Tarski}(S, \text{form}) \). \\
  \item \( B^{is} := B \cup \bigcup \{ \text{Tarski}(S, \text{form}_n) : n \in \omega \} \cup \text{ind}(S) \), \\
  \text{where form}_n \text{ is the collection of formulas } \mathcal{L}_B \text{ with quantifier} \\
  \text{alternation depth at most } n \text{ and ind}(S) \text{ is the full scheme of} \\
  \text{induction on the designated numbers in the language } \mathcal{L}_B(S). \\
  \item \( B^{fis} := B^{fs} \cup B^{is} \). 
\end{itemize}
Extensions of the Base Theory

- $B^{fs} := B \cup \text{Tarski}(S, \text{form})$.

- $B^{is} := B \cup \bigcup \left\{ \text{Tarski}(S, \text{form}_n) : n \in \omega \right\} \cup \text{ind}(S)$, where $\text{form}_n$ is the collection of formulas $\mathcal{L}_B$ with quantifier alternation depth at most $n$ and $\text{ind}(S)$ is the full scheme of induction on the designated numbers in the language $\mathcal{L}_B(S)$.

- $B^{fis} := B^{fs} \cup B^{is}$. 
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 $(\mathcal{M}, S, F) \models \text{Tarski}(S, F)$.

- If $F = \text{Form}^\mathcal{M} \cap \omega$, then we say that $F$ is the set of standard $\mathcal{L}_B$-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 $F := \text{Form}^\mathcal{M}$. This is equivalent to $(\mathcal{M}, S, F) \models B^{fs}$. 
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 \( (\mathcal{M}, S, F) \models \text{Tarski}(S, F) \).

► If \( F = \text{Form}^M \cap \omega \), then we say that \( F \) is the set of standard \( \mathcal{L}_B \)-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 \( F := \text{Form}^M \). This is equivalent to \( (\mathcal{M}, S, F) \models B^{fs} \).
Satisfaction Classes

- Suppose $\mathcal{M} \models B$ and $F$ is a subset of $M$ and $S$ is a binary relation on $\mathcal{M}$.

- $S$ is an $F$-satisfaction class if $(\mathcal{M}, S, F) \models \text{Tarski}(S, F)$.

- If $F = \text{Form}^M \cap \omega$, then we say that $F$ is the set of standard $\mathcal{L}_B$-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 $F := \text{Form}^M$. This is equivalent to $(\mathcal{M}, S, F) \models B^{fs}$. 

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 $(\mathcal{M}, S, F) \models \text{Tarski}(S, F)$.

- If $F = \text{Form}^\mathcal{M} \cap \omega$, then we say that $F$ is the set of standard $\mathcal{L}_B$-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 $F := \text{Form}^\mathcal{M}$. This is equivalent to $(\mathcal{M}, S, F) \models B^{fs}$. 
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 $(\mathcal{M}, S, F) \models \text{Tarski}(S, F)$.

- If $F = \text{Form}^\mathcal{M} \cap \omega$, then we say that $F$ is the set of standard $\mathcal{L}_B$-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 $F := \text{Form}^\mathcal{M}$. This is equivalent to $(\mathcal{M}, S, F) \models B^{fs}$. 
The Conservativity of $\text{PA}^{\text{FS}}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of $\text{PA}$ carries a full satisfaction class.*

- **Theorem.** $\text{PA}^{\text{fs}}$ is conservative over $\text{PA}$.

- 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 $\text{PA}$, or to theories that are formulated in infinite languages.

- **Theorem** (Lachlan (1981)). *Every non-standard model of $\text{PA}$ that carries a full satisfaction class is recursively saturated.*

- Since we can construct examples of non-standard models of $\text{PA}$ that are not recursively saturated, we see that not every model of $\text{PA}$ has an *expansion* with a full satisfaction class.
The Conservativity of $\text{PA}^{\text{FS}}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of $\text{PA}$ carries a full satisfaction class.*

- **Theorem.** $\text{PA}^{\text{FS}}$ is conservative over $\text{PA}$.

- 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 $\text{PA}$, or to theories that are formulated in infinite languages.

- **Theorem** (Lachlan (1981)). *Every non-standard model of $\text{PA}$ that carries a full satisfaction class is recursively saturated.*

- Since we can construct examples of non-standard models of $\text{PA}$ that are not recursively saturated, we see that not every model of $\text{PA}$ has an *expansion* with a full satisfaction class.
The Conservativity of $\text{PA}^{\text{FS}}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of $\text{PA}$ carries a full satisfaction class.*

- **Theorem.** $\text{PA}^{\text{fs}}$ *is conservative over* $\text{PA}$.

- 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 $\text{PA}$, or to theories that are formulated in infinite languages.

- **Theorem** (Lachlan (1981)). *Every non-standard model of $\text{PA}$ that carries a full satisfaction class is recursively saturated.*

- Since we can construct examples of non-standard models of $\text{PA}$ that are not recursively saturated, we see that not every model of $\text{PA}$ has an *expansion* with a full satisfaction class.
The Conservativity of $\text{PA}^{\text{FS}}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of $\text{PA}$ carries a full satisfaction class.***

- **Theorem.** $\text{PA}^{\text{fs}}$ is conservative over $\text{PA}$.

- 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 $\text{PA}$, or to theories that are formulated in infinite languages.

- **Theorem** (Lachlan (1981)). *Every non-standard model of $\text{PA}$ that carries a full satisfaction class is recursively saturated.*

- Since we can construct examples of non-standard models of $\text{PA}$ that are not recursively saturated, we see that not every model of $\text{PA}$ has an expansion with a full satisfaction class.
The Conservativity of $\text{PA}^\text{FS}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of PA carries a full satisfaction class.*

- **Theorem.** $\text{PA}^\text{fs}$ *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.
The Conservativity of $\text{PA}^{\text{FS}}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of PA carries a full satisfaction class.*

- **Theorem**. $\text{PA}^{\text{fs}}$ is conservative over $\text{PA}$.

- 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.
The Conservativity of $\text{PA}^{\text{FS}}$ over $\text{PA}$

- **Theorem** (Kotlarski-Krajewski-Lachlan (1981)). *Every countable recursively saturated model of $\text{PA}$ carries a full satisfaction class.*

- **Theorem.** $\text{PA}^{\text{fs}}$ is conservative over $\text{PA}$.

- 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 $\text{PA}$, or to theories that are formulated in infinite languages.

- **Theorem** (Lachlan (1981)). *Every non-standard model of $\text{PA}$ that carries a full satisfaction class is recursively saturated.*

- Since we can construct examples of non-standard models of $\text{PA}$ that are not recursively saturated, we see that not every model of $\text{PA}$ has an *expansion* with a full satisfaction class.
Core Lemma. Let $\mathcal{N}_0 \models B$ and suppose $S_0$ is an $F_0$-satisfaction class, where $F_0 \subseteq F_1 := \text{Form}^{\mathcal{N}_0}$. Then, there is an elementary extension $\mathcal{N}_1$ of $\mathcal{N}_0$ that carries an $F_1$-satisfaction class $S_1$ such that $S_0 = S_1 \cap (F_0 \times \mathcal{N}_0)$.

Proof. Let $\mathcal{L}_B^+(\mathcal{N}_0)$ be the language obtained by enriching $\mathcal{L}_B$ with constant symbols for each member of $\mathcal{N}_0$, and new unary predicates $U_c$ for each $c \in \text{Form}^{\mathcal{N}_0}$. 
Core Lemma. Let $\mathcal{N}_0 \models B$ and suppose $S_0$ is an $F_0$-satisfaction class, where $F_0 \subseteq F_1 := \text{Form}^{\mathcal{N}_0}$. Then, there is an elementary extension $\mathcal{N}_1$ of $\mathcal{N}_0$ that carries an $F_1$-satisfaction class $S_1$ such that $S_0 = S_1 \cap (F_0 \times \mathcal{N}_0)$.

Proof. Let $\mathcal{L}^+_{B}(\mathcal{N}_0)$ be the language obtained by enriching $\mathcal{L}_B$ with constant symbols for each member of $\mathcal{N}_0$, and new unary predicates $U_c$ for each $c \in \text{Form}^{\mathcal{N}_0}$. 
Core Lemma. Let $\mathcal{N}_0 \models B$ and suppose $S_0$ is an $F_0$-satisfaction class, where $F_0 \subseteq F_1 := \text{Form}^{\mathcal{N}_0}$. Then, there is an elementary extension $\mathcal{N}_1$ of $\mathcal{N}_0$ that carries an $F_1$-satisfaction class $S_1$ such that $S_0 = S_1 \cap (F_0 \times \mathcal{N}_0)$.

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


The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \neg \top$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land R([\alpha]_{v_0}, \ldots, [\alpha]_{v_{n-1}}) \right).
  \]

- If $\mathcal{N} \models c = \neg \top$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land \neg U_d(\alpha) \right).
  \]

- If $\mathcal{N} \models c = d_1 \lor d_2$, then $\theta_c :=$
  \[
  \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land (U_{d_1}(\alpha \upharpoonright \text{FV}(d_1)) \lor U_{d_2}(\alpha \upharpoonright \text{FV}(d_2)))) \right).
  \]

- If $\mathcal{N} \models c = \exists \nu b$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \exists \alpha' \supset \alpha \ U_b(\alpha') \right).
  \]
The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \lnot R(v_0, \cdots, v_{n-1}) \land$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).
  \]

- If $\mathcal{N} \models c = \lnot d \land$ then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land \lnot U_d(\alpha) \right).
  \]

- If $\mathcal{N} \models c = d_1 \lor d_2 \land$, then $\theta_c :=$
  \[
  \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land (U_{d_1}(\alpha \upharpoonright \text{FV}(d_1)) \lor U_{d_2}(\alpha \upharpoonright \text{FV}(d_2))) \right).
  \]

- If $\mathcal{N} \models c = \exists v \ b \land$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \ U_b(\alpha') \right).
  \]
The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \neg R(v_0, \ldots, v_{n-1})$, then
  \[ \theta_c := \forall \alpha (U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land R([\alpha]_{v_0}, \ldots, [\alpha]_{v_{n-1}})) . \]

- If $\mathcal{N} \models c = \neg d$, then
  \[ \theta_c := \forall \alpha (U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land \neg U_d(\alpha)) . \]

- If $\mathcal{N} \models c = d_1 \lor d_2$, then
  \[ \theta_c := \forall \alpha (U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land (U_{d_1}(\alpha \uparrow \text{FV}(d_1)) \lor U_{d_2}(\alpha \uparrow \text{FV}(d_2)))) . \]

- If $\mathcal{N} \models c = \exists v \ b$, then
  \[ \theta_c := \forall \alpha (U_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \ U_b(\alpha')) . \]
The Core Construction 2

- If $R \in \mathcal{L}_B$ and $\mathcal{N}_0 \models c = \lnot R(v_0, \cdots, v_{n-1})$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land R([\alpha]_{v_0}, \cdots, [\alpha]_{v_{n-1}}) \right).
  \]

- If $\mathcal{N} \models c = \lnot d$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land \lnot U_d(\alpha) \right).
  \]

- If $\mathcal{N} \models c = d_1 \lor d_2$, then $\theta_c :=$
  \[
  \forall \alpha \left( U_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land (U_{d_1}(\alpha \upharpoonright \text{FV}(d_1)) \lor U_{d_2}(\alpha \upharpoonright \text{FV}(d_2))) \right).
  \]

- If $\mathcal{N} \models c = \exists v b$, then
  \[
  \theta_c := \forall \alpha \left( U_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \ U_b(\alpha') \right).
  \]
The Core Construction 2

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

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

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

  $$\theta_c := \forall \alpha \left( \text{U}_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land \neg \text{U}_d(\alpha) \right).$$

- If $\mathcal{N} \models c = \neg (d_1 \lor d_2)$, then $\theta_c :=$

  $$\forall \alpha \left( \text{U}_c(\alpha) \leftrightarrow \text{asn}(\alpha, c) \land \left( \text{U}_{d_1}(\alpha \upharpoonright \text{FV}(d_1)) \lor \text{U}_{d_2}(\alpha \upharpoonright \text{FV}(d_2)) \right) \right).$$

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

  $$\theta_c := \forall \alpha \left( \text{U}_c(\alpha) \leftrightarrow \exists \alpha' \supseteq \alpha \ \text{U}_b(\alpha) \right).$$
Let

\[ \Gamma := \{ U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \in S_0 \} \cup \{ \neg U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \notin S_0 \} \]

and define \( \text{Th}^+(\mathcal{N}_0) := \text{Th}(\mathcal{N}_0, c_{c \in N_0} \cup \Theta \cup \Gamma) \).

We show that \( \text{Th}^+(\mathcal{N}_0) \) is consistent by demonstrating that each finite subset of \( \text{Th}^+(\mathcal{N}_0) \) is interpretable in \( (\mathcal{N}_0, S_0) \).

Suppose \( T_0 \) is a finite subset of \( \text{Th}^+(\mathcal{N}_0) \) 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 \( (c, \alpha) \in S_0 \).
The Core Construction 3

Let

\[ \Gamma := \{ U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \in S_0 \} \cup \{ \neg U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \not\in S_0 \} \]

and define \( \text{Th}^+(N_0) := \text{Th}(N_0, c_{c \in N_0} \cup \Theta \cup \Gamma) \).

We show that \( \text{Th}^+(N_0) \) is consistent by demonstrating that each finite subset of \( \text{Th}^+(N_0) \) is interpretable in \( (N_0, S_0) \).

Suppose \( T_0 \) is a finite subset of \( \text{Th}^+(N_0) \) 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. \( (N_0, U_c)_{c \in C} \models \{ \theta_c : c \in C \} \) and
2. if \( c \in C \cap F_0 \), then \( \alpha \in U_c \) iff \( (c, \alpha) \in S_0 \).
The Core Construction 3

Let

\[ \Gamma := \{ U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \in S_0 \} \cup \{ \neg U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \notin S_0 \} \]

and define \( \text{Th}^+(N_0) := \text{Th}(N_0, c_{c \in N_0}) \cup \Theta \cup \Gamma \).

We show that \( \text{Th}^+(N_0) \) is consistent by demonstrating that each finite subset of \( \text{Th}^+(N_0) \) is interpretable in \( (N_0, S_0) \).

Suppose \( T_0 \) is a finite subset of \( \text{Th}^+(N_0) \) 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. \( (N_0, U_c)_{c \in C} \models \{ \theta_c : c \in C \} \)
2. if \( c \in C \cap F_0 \), then \( \alpha \in U_c \) iff \( (c, \alpha) \in S_0 \).
The Core Construction 3

Let

\[ \Gamma := \{ U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \in S_0 \} \cup \{ \neg U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \not\in S_0 \} \]

and define \( \text{Th}^+(N_0) := \text{Th}(N_0, c_{c\in N_0} \cup \Theta \cup \Gamma) \).

We show that \( \text{Th}^+(N_0) \) is consistent by demonstrating that each finite subset of \( \text{Th}^+(N_0) \) is interpretable in \((N_0, S_0)\).

Suppose \( T_0 \) is a finite subset of \( \text{Th}^+(N_0) \) 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. \( (N_0, U_c)_{c\in C} \models \{ \theta_c : c \in C \} \) and
2. if \( c \in C \cap F_0 \), then \( \alpha \in U_c \) iff \( (c, \alpha) \in S_0 \).
The Core Construction 3

Let

\[ \Gamma := \{ U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \in S_0 \} \cup \{ \neg U_c(\alpha) : c \in F_0 \text{ and } (c, \alpha) \notin S_0 \} \]

and define \( \text{Th}^+(N_0) := \text{Th}(N_0, c)_{c \in N_0} \cup \Theta \cup \Gamma \).

We show that \( \text{Th}^+(N_0) \) is consistent by demonstrating that each finite subset of \( \text{Th}^+(N_0) \) is interpretable in \( (N_0, S_0) \).

Suppose \( T_0 \) is a finite subset of \( \text{Th}^+(N_0) \) 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. \( (N_0, U_c)_{c \in C} \models \{ \theta_c : c \in C \} \) and
2. if \( c \in C \cap F_0 \), then \( \alpha \in U_c \) iff \( (c, \alpha) \in S_0 \).
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.

\( c \rhd d \) expresses: \( c \) is a direct subformula of \( d \).

Define \( \lhd^* \) on \( C \) by:

\[
\begin{align*}
 c \lhd^* d \iff (c \lhd d)^{\mathbb{N}_0} \text{ and } \theta_d \in T_0.
\end{align*}
\]

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:

\[
\begin{align*}
\text{rank}_C(x) := \sup\{\text{rank}_C(y) + 1 : x \in C \text{ and } (y \lhd^* x)^{\mathbb{N}_0}\}.
\end{align*}
\]
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.
- \( c \triangleleft d \) expresses: \( c \) is a direct subformula of \( d \).
- Define \( \triangleleft^* \) on \( C \) by:

  \[
  c \triangleleft^* d \iff (c \triangleleft d)^{N_0} \text{ and } \theta_d \in T_0.
  \]

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

  \[
  \text{rank}_C(x) := \sup \{ \text{rank}_C(y) + 1 : x \in C \text{ and } (y \triangleleft^* x)^{N_0} \}.
  \]
We shall construct \( \{ U_c : c \in C \} \) \textit{in stages}, beginning with the simplest formulas in \( C \), and working our way up using the Tarski rules for more complex ones.

\( c \triangleleft d \) expresses: \( c \) is a direct subformula of \( d \).

Define \( \triangleleft^* \) on \( C \) by:

\[
c \triangleleft^* d \iff (c \triangleleft d)^{N_0} \text{ and } \theta_d \in T_0.
\]

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

\[
\text{rank}_C(x) := \sup\{\text{rank}_C(y) + 1 : x \in C \text{ and } (y \triangleleft^* x)^{N_0}\}.
\]
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.

\( c \triangleleft d \) expresses: \( c \) is a direct subformula of \( d \).

Define \( \triangleleft^* \) on \( C \) by:

\[
\text{c} \triangleleft^* \text{d} \iff (\text{c} \triangleleft \text{d})^N_0 \text{ and } \theta_d \in T_0.
\]

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

\[
\text{rank}_C(x) := \sup \{ \text{rank}_C(y) + 1 : x \in C \text{ and } (y \triangleleft^* x)^N_0 \}.
\]
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.

\( c \triangleleft d \) expresses: \( c \) is a direct subformula of \( d \).

Define \( \triangleleft^* \) on \( C \) by:

\[
\text{\( c \triangleleft^* d \iff (c \triangleleft d)^{N_0} \text{ and } \theta_d \in T_0. \)}
\]

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

\[
\text{rank}_C(x) := \sup\{ \text{rank}_C(y) + 1 : x \in C \text{ and } (y \triangleleft^* x)^{N_0} \}.
\]
The Core Construction 5

- Note that \( \text{rank}_G(c) = 0 \) precisely when there is no \( x \in C \) such that \( (x \triangleleft^* c)^{N_0} \).
- Next, let \( C_i := \{ x \in C : \text{rank}_C(x) \leq i \} \).
- 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 \( c \in C \).
Note that \( \text{rank}_G(c) = 0 \) precisely when there is no \( x \in C \) such that \( (x \triangleleft^* c) \backslash \mathcal{N}_0 \).

Next, let \( C_i := \{ x \in C : \text{rank}_C(x) \leq i \} \).

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 \( c \in C \).
Note that \( \text{rank}_G(c) = 0 \) precisely when there is no \( x \in C \) such that \( (x \triangleleft^* c)^N \setminus N_0 \).

Next, let \( C_i := \{ x \in C : \text{rank}_C(x) \leq i \} \).

Since \( C \) is finite and non-empty, we have \( C_0 \neq \emptyset \).

c \in C_0 \iff (c \in C \text{ and } C \text{ 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 \( c \in C \).
Note that $\text{rank}_G(c) = 0$ precisely when there is no $x \in C$ such that $(x \triangleleft^* c) \cap N_0$.

Next, let $C_i := \{ x \in C : \text{rank}_C(x) \leq i \}$.

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 $c \in C$. 

Note that \( \text{rank}_G(c) = 0 \) precisely when there is no \( x \in C \) such that \( (x \triangleleft^* c)^{\mathcal{N}_0} \).

Next, let \( C_i := \{x \in C : \text{rank}_C(x) \leq i\} \).

Since \( C \) is finite and non-empty, we have \( C_0 \neq \emptyset \).

\( c \in C_0 \) iff \( (c \in C \text{ and } C \text{ does not contain the code of any subformula of the formula coded by } c) \).

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 \( c \in C \).
Note that \( \operatorname{rank}_G(c) = 0 \) precisely when there is no \( x \in C \) such that \( (x \triangleleft^* c) \downarrow \mathbb{N}_0 \).

Next, let \( C_i := \{ x \in C : \operatorname{rank}_C(x) \leq i \} \).

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 \( c \in C \).
The Core Construction 6

- If \( c \in C_0 \) then \( U_c := \begin{cases} \{ \alpha : (c, \alpha) \in S_0 \}, & \text{if } c \in F_0; \\ \emptyset, & \text{if } c \notin F_0. \end{cases} \)

- If \( c \in C_{i+1} \setminus C_i \) and \( \neg c = \neg d \), then
  \[
  U_c := \{ \alpha \in \text{asn}_c : \alpha \notin U_d \}.
  \]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = a \lor b \), then
  \[
  U_c := \{ \alpha \in \text{asn}_c : \alpha \upharpoonright \text{fv}(a) \in U_a \text{ or } \alpha \upharpoonright \text{fv}(b) \in U_b \}.
  \]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = \exists v \ b \), then
  \[
  U_c := \{ \alpha \in \text{asn}_c : \exists \alpha' \supseteq \alpha \alpha' \in U_b \}.
  \]
The Core Construction 6

- If \( c \in C_0 \) then 
  \[
  U_c := \begin{cases} 
    \{ \alpha : (c, \alpha) \in S_0 \}, & \text{if } c \in F_0; \\
    \emptyset, & \text{if } c \notin F_0.
  \end{cases}
  \]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = \neg d \), then 
  \[
  U_c := \{ \alpha \in \text{asn}_c : \alpha \notin U_d \}.
  \]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = a \lor b \), then 
  \[
  U_c := \{ \alpha \in \text{asn}_c : \alpha \upharpoonright \text{fv}(a) \in U_a \text{ or } \alpha \upharpoonright \text{fv}(b) \in U_b \}.
  \]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = \exists v \ b \), then 
  \[
  U_c := \{ \alpha \in \text{asn}_c : \exists \alpha' \supseteq \alpha \alpha' \in U_b \}.
  \]
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 $\neg c = \neg d \neg$, then

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

▶ If $c \in C_{i+1} \setminus C_i$ and $c = \neg a \lor b \neg$, then

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

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

$$U_c := \{ \alpha \in \text{asn}_c : \exists \alpha' \supseteq \alpha \alpha' \in U_b \}.$$
The Core Construction 6

- If \( c \in C_0 \) then \( U_c := \begin{cases} \{ \alpha : (c, \alpha) \in S_0 \}, & \text{if } c \in F_0; \\ \emptyset, & \text{if } c \notin F_0. \end{cases} \)

- If \( c \in C_{i+1} \setminus C_i \) and \( \neg c = \neg \neg d \), then

\[
U_c := \{ \alpha \in \text{asn}_c : \alpha \notin U_d \}. 
\]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = \neg a \lor b \), then

\[
U_c := \{ \alpha \in \text{asn}_c : \alpha \upharpoonright \text{fv}(a) \in U_a \text{ or } \alpha \upharpoonright \text{fv}(b) \in U_b \} .
\]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = \neg \exists v b \), then

\[
U_c := \{ \alpha \in \text{asn}_c : \exists \alpha' \supseteq \alpha \alpha' \in U_b \}. 
\]
The Core Construction 6

- If \( c \in C_0 \) then \( U_c := \begin{cases} \{ \alpha : (c, \alpha) \in S_0 \}, & \text{if } c \in F_0; \\ \emptyset, & \text{if } c \notin F_0. \end{cases} \)

- If \( c \in C_{i+1} \setminus C_i \) and \( \neg c = \neg d \), then

\[
U_c := \{ \alpha \in \text{asn}_c : \alpha \notin U_d \}.
\]

- If \( c \in C_{i+1} \setminus C_i \) and \( c = a \lor b \), then

\[
U_c := \{ \alpha \in \text{asn}_c : \alpha \upharpoonright \text{fv}(a) \in U_a \text{ or } \alpha \upharpoonright \text{fv}(b) \in U_b \}.
\]

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

\[
U_c := \{ \alpha \in \text{asn}_c : \exists \alpha' \supseteq \alpha \alpha' \in U_b \}.
\]
The Core Construction 7

- **Core Theorem.** Let $M_0$ be a model of $B$ of any cardinality. There is an elementary extension $M$ of $M_0$ 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 $M_1$ of $M_0$ that carries a $\text{Form}_{B}^{M_0}$ satisfaction class. Thanks to the Core Lemma, this argument can be carried out countably many times to yield two sequences $\langle M_i : i \in \omega \rangle$ and $\langle S_i : i \in \omega \rangle$ that satisfy the following two properties:
  1. $M_i \prec M_{i+1}$;
  2. $S_{i+1}$ is a $\text{Form}_{M_i}$-satisfaction class on $M_{i+1}$ for each $i \in \omega$.

- We take: $M := \bigcup_{i \in \omega} M_i$, and $S := \bigcup_{i \in \omega} S_i$. 
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 $\text{Form}_{\mathcal{M}_0}^B$ 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 $\text{Form}_{\mathcal{M}_i}^\mathcal{M}$-satisfaction class on $\mathcal{M}_{i+1}$ for each $i \in \omega$.

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

Core Theorem. Let $M_0$ be a model of $B$ of any cardinality. There is an elementary extension $M$ of $M_0$ that admits a full satisfaction class.

Proof: Let $F_0$ be the set of atomic $N$-formulas and let $S_0$ be the obvious satisfaction predicate for $F_0$. Then by the Lemma there is an elementary extension $M_1$ of $M_0$ that carries a Form$_B^{M_0}$ satisfaction class. Thanks to the Core Lemma, this argument can be carried out countably many times to yield two sequences $\langle M_i : i \in \omega \rangle$ and $\langle S_i : i \in \omega \rangle$ that satisfy the following two properties:

1. $M_i \prec M_{i+1}$;

2. $S_{i+1}$ is a Form$_{M_i}$-satisfaction class on $M_{i+1}$ for each $i \in \omega$.

We take: $M := \bigcup_{i \in \omega} M_i$, and $S := \bigcup_{i \in \omega} S_i$. 
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 $\text{Form}^{\mathcal{M}_0}_B$ 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 $\text{Form}^{\mathcal{M}_i}$-satisfaction class on $\mathcal{M}_{i+1}$ for each $i \in \omega$.

We take: $\mathcal{M} := \bigcup_{i \in \omega} \mathcal{M}_i$, and $S := \bigcup_{i \in \omega} S_i$. 
**Corollary.** $B^s$ is a conservative extension of $B$, for every base theory $B$. 
Corollary. $B^{fs}$ is a conservative extension of $B$, for every base theory $B$. 
Theorem. The following statement (*) is provable within WKL₀:

(*) 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. PRA ⊢ “$B^{fs}$ is conservative over B” for every r.e. base theory B.
The Arithmetization of the Core Construction

Theorem. The following statement (*) is provable within WKL₀:

(*) 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. PRA ⊢ “$B^{fs}$ is conservative over $B$” for every r.e. base theory $B$. 
The Arithmetization of the Core Construction

Theorem. The following statement (*) is provable within WKL₀:

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

Theorem. PRA ⊢ “$B^{fs}$ is conservative over B” for every r.e. base theory B.
Definition. For any standard formula $\sigma$ of $\mathcal{L}_B$, and for each $a \in \mathbb{N}^M$, where $M$ is a model of $B$, the ‘formula’ $\sigma_a$ is defined by internal recursion in $M_0$ via $\sigma_0 := \sigma$; and $\sigma_{n+1} := \sigma_n \lor \sigma_n$.

Theorem. Let $\sigma := \exists v_0 (v_0 = v_0)$ and let $M_0$ be a model of $B$ of any cardinality. Then, $M_0$ has an elementary extension $M$ that carries a full satisfaction class $S$ such that

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

- **Definition.** For any standard formula $\sigma$ of $\mathcal{L}_B$, and for each $a \in \mathbb{N}^\mathcal{M}$, where $\mathcal{M}$ is a model of $B$, the ‘formula’ $\sigma_a$ is defined by internal recursion in $\mathcal{M}_0$ via $\sigma_0 := \sigma$; and $\sigma_{n+1} := \sigma_n \lor \sigma_n$.

- **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

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

- **Definition.** For any standard formula $\sigma$ of $\mathcal{L}_B$, and for each $a \in \mathbb{N}^\mathcal{M}$, where $\mathcal{M}$ is a model of $B$, the ‘formula’ $\sigma_a$ is defined by internal recursion in $\mathcal{M}_0$ via $\sigma_0 := \sigma$; and $\sigma_{n+1} := \sigma_n \lor \sigma_n$.

- **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

$$\{ a \in \mathbb{N}^\mathcal{M} : \sigma_a \text{ is } S\text{-valid} \} = \omega.$$
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.
Theorem. Let $M_0 \models B$, where $B$ is a base theory. There is an elementary extension $M$ of $M_0$ that carries full satisfaction classes $S_1$, $S_2$, and $S_3$ such that:

- $S_1$ is schematically correct;
- $S_2$ is both existentially and disjunctively correct;
- $S_3$ is both extensional and alphabetically correct.

Moreover, if $B$ is an inductive base theory, then $M$ carries a full satisfaction class $S_4$ such that: $S_4$ is $\Sigma_{B,\infty}$-correct.
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.
Let ACA be the strengthening of ACA$_0$ with the full scheme of induction. It has been long known that ACA and PA$^{\text{fis}}$ are ‘proof-theoretically equivalent’. The result below provides a more precise relationship between the two theories.

**Theorem.** There is a sentence $\sigma$ in the language of ACA$_0$ such that PA$^{\text{fis}}$ and ACA $+ \sigma$ are bi-interpretable.

**Theorem.** $B^{\text{is}}$ and $B^{\text{fs}}$ are both interpretable in B for every inductive recursively axiomatizable base theory B.
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 PA$^{\text{fis}}$ are ‘proof-theoretically equivalent’. The result below provides a more precise relationship between the two theories.

- **Theorem.** There is a sentence $\sigma$ in the language of ACA$_0$ such that PA$^{\text{fis}}$ and ACA + $\sigma$ are bi-interpretable.

- **Theorem.** $B^{\text{is}}$ and $B^{\text{fs}}$ are both interpretable in $B$ for every inductive recursively axiomatizable base theory $B$. 
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 PA$^\text{fis}$ are ‘proof-theoretically equivalent’. The result below provides a more precise relationship between the two theories.

**Theorem.** There is a sentence $\sigma$ in the language of ACA$_0$ such that PA$^\text{fis}$ and ACA + $\sigma$ are bi-interpretable.

**Theorem.** $B^\text{is}$ and $B^\text{fs}$ are both interpretable in $B$ for every inductive recursively axiomatizable base theory $B$. 
Let ACA be the strengthening of ACA$_0$ with the full scheme of induction. It has been long known that ACA and PA$^{\text{fis}}$ are ‘proof-theoretically equivalent’. The result below provides a more precise relationship between the two theories.

**Theorem.** *There is a sentence $\sigma$ in the language of ACA$_0$ such that PA$^{\text{fis}}$ and ACA + $\sigma$ are bi-interpretable.*

**Theorem.** *$B^{\text{is}}$ and $B^{\text{fis}}$ are both interpretable in $B$ for every inductive recursively axiomatizable base theory $B$.***
Theorem (Interpretability among PA, PA_{is}, PA_{fs}, and ACA_0).

a. The theories PA, PA_{is}, PA_{fs} are mutually interpretable.

b. Each of the theories PA, PA_{is}, PA_{fs} is interpretable in ACA_0, but none of them interprets ACA_0.

c. No pair of the theories PA, PA_{fs}, PA_{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.
Interpretability Issues 2

- **Theorem** (Interpretability among PA, PA$^{is}$, PA$^{fs}$, and ACA$_0$).
  
  a. *The theories PA, PA$^{is}$, PA$^{fs}$ are mutually interpretable.*
  
  b. *Each of the theories PA, PA$^{is}$, PA$^{fs}$ is interpretable in ACA$_0$, but none of them interprets ACA$_0$.*
  
  c. *No pair of the theories PA, PA$^{fs}$, PA$^{is}$, ACA$_0$ are bi-interpretable.*

- If $B$ is a consistent finitely axiomatizable base theory, then neither $B^{is}$ nor $B^{fs}$ is interpretable in $B$. 
Theorem (Interpretability among PA, PA$^{\text{is}}$, PA$^{\text{fs}}$, and ACA$_0$).

a. The theories PA, PA$^{\text{is}}$, PA$^{\text{fs}}$ are mutually interpretable.

b. Each of the theories PA, PA$^{\text{is}}$, PA$^{\text{fs}}$ is interpretable in ACA$_0$, but none of them interprets ACA$_0$.

c. No pair of the theories PA, PA$^{\text{fs}}$, PA$^{\text{is}}$, ACA$_0$ are bi-interpretable.

If B is a consistent finitely axiomatizable base theory, then neither B$^{\text{is}}$ nor B$^{\text{fs}}$ is interpretable in B.
Some Open Questions

▶ Can the conservativity of $\text{PA}^{\text{fs}}$ over $\text{PA}$ be verified in $\text{EFA}$ (Exponential Function Arithmetic)?

▶ Is $\text{PA}^{\text{fs}} + \text{“S contains all propositional tautologies”}$ conservative over $\text{PA}$?

Cezary Cieśliński has shown that $\text{PA}^{\text{fs}} + \text{“S is closed under derivability in propositional logic”}$ is not conservative over $\text{PA}$.
Some Open Questions

- Can the conservativity of \( \text{PA}^{\text{fs}} \) over PA be verified in EFA (Exponential Function Arithmetic)?

- Is \( \text{PA}^{\text{fs}} + \text{“S contains all propositional tautologies”} \) conservative over PA?
  
  Cezary Cieśliński has shown that \( \text{PA}^{\text{fs}} + \text{“S is closed under derivability in propositional logic”} \) is not conservative over PA.
Some Open Questions

▶ Can the conservativity of \( \text{PA}^f \) over \( \text{PA} \) be verified in EFA (Exponential Function Arithmetic)?

▶ Is \( \text{PA}^f + \) “\( S \) contains all propositional tautologies” conservative over \( \text{PA} \)?

Cezary Cieśliński has shown that \( \text{PA}^f + \) “\( S \) is closed under derivability in propositional logic” is not conservative over \( \text{PA} \).