Abstract Gödelian languages: continuation

András Máté

23.02.2024

An abstract Tarski-like theorem

Theorem T

i. $\tilde{T}^*$ is not expressible.

ii. If G_1 holds, then $\tilde{T}$ is not expressible.

iii. If G_1 and G_2 holds, then T is not expressible.

An abstract Tarski-like theorem

Theorem T

i. $\tilde{T}^*$ is not expressible.
ii. If G_1 holds, then $\tilde{T}$ is not expressible.
iii. If G_1 and G_2 holds, then T is not expressible.

Proof of i. by D: If it is expressible, then $\tilde{T}$ has a Gödel sentence. This sentence belongs to $\mathcal{T}$ iff its Gödel number belongs to $\tilde{T}$. I.e., it is true iff it is false.

Theorem T
i. $\tilde{T}^*$ is not expressible.
ii. If G_1 holds, then $\tilde{T}$ is not expressible.
iii. If G_1 and G_2 holds, then T is not expressible.

Proof of i. by D: If it is expressible, then $\tilde{T}$ has a Gödel sentence. This sentence belongs to $\mathcal{T}$ iff its Gödel number belongs to $\tilde{T}$. I.e., it is true iff it is false.

Proofs of ii. and iii. by *modus tollens* from i. and the conditions.

Consistency, decidability and completeness

Consistency, decidability and completeness

$\mathcal{L}$ is consistent iff $\mathcal{P}$ and $\mathcal{R}$ are disjoint; inconsistent otherwise.

Consistency, decidability and completeness

$\mathcal{L}$ is consistent iff $\mathcal{P}$ and $\mathcal{R}$ are disjoint; inconsistent otherwise.

Correct systems are always consistent, but not the other way.

Consistency, decidability and completeness

$\mathcal{L}$ is consistent iff $\mathcal{P}$ and $\mathcal{R}$ are disjoint; inconsistent otherwise.

Correct systems are always consistent, but not the other way.

Sentence S is decidable iff it is either provable or refutable; undecidable otherwise.

$\mathcal{L}$ is consistent iff $\mathcal{P}$ and $\mathcal{R}$ are disjoint; inconsistent otherwise.

Correct systems are always consistent, but not the other way.

Sentence S is decidable iff it is either provable or refutable; undecidable otherwise.

$\mathcal{L}$ is complete iff every sentence is decidable; incomplete otherwise.

Two GT-like theorems

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Two GT-like theorems

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Two GT-like theorems

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Proof for Theorem 1.^{o}:
If K_k expresses R^*, then $K_k(k)$ is true iff $k \in R^*$

Two GT-like theorems

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Proof for Theorem 1.^{o}:
If K_k expresses R^*, then $K_k(k)$ is true iff $k \in R^*$ iff $d(k) \in R$

Two GT-like theorems

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Proof for Theorem 1.^{o}:
If K_k expresses R^*, then $K_k(k)$ is true iff $k \in R^*$ iff $d(k) \in R$
But the Gödel-number of $K_k(k)$ is just $d(k)$.

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Proof for Theorem 1.^{o}:
If K_k expresses R^*, then $K_k(k)$ is true iff $k \in R^*$ iff $d(k) \in R$
But the Gödel-number of $K_k(k)$ is just $d(k)$.
Therefore, $K_k(k)$ is true iff it is refutable.

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Proof for Theorem 1.^{o}:
If K_k expresses R^*, then $K_k(k)$ is true iff $k \in R^*$ iff $d(k) \in R$
But the Gödel-number of $K_k(k)$ is just $d(k)$.
Therefore, $K_k(k)$ is true iff it is refutable. By correctness, it is false but not refutable.

Two GT-like theorems

Theorem 1.
If $\mathcal{L}$ is correct and $\tilde{P}^*$ is expressible, then $\mathcal{L}$ is incomplete.

Theorem 1.^{o} (Dual of Theorem 1.)
If $\mathcal{L}$ is correct and R^* is expressible, then $\mathcal{L}$ is incomplete.

Proof for Theorem 1.^{o}:
If K_k expresses R^*, then $K_k(k)$ is true iff $k \in R^*$ iff $d(k) \in R$
But the Gödel-number of $K_k(k)$ is just $d(k)$.
Therefore, $K_k(k)$ is true iff it is refutable. By correctness, it is false but not refutable.

Corollary: If G_1 holds and R is expressible, then $\mathcal{L}$ is incomplete.

Some exercises

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H <u>represents</u> A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H represents A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H represents A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H represents A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$

$$H_h(h) \in \mathcal{P} \leftrightarrow h \in A$$

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H <u>represents</u> A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$

$$d(h) \in P \leftrightarrow H_h(h) \in \mathcal{P} \leftrightarrow h \in A$$

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H represents A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$
$h \in P^* \leftrightarrow d(h) \in P \leftrightarrow H_h(h) \in \mathcal{P} \leftrightarrow h \in A$

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H represents A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$
$h \in P^* \leftrightarrow d(h) \in P \leftrightarrow H_h(h) \in \mathcal{P} \leftrightarrow h \in A$
Therefore, $h \notin P^*$, $h \notin A$,

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H <u>represents</u> A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$
$h \in P^* \leftrightarrow d(h) \in P \leftrightarrow H_h(h) \in \mathcal{P} \leftrightarrow h \in A$
Therefore, $h \notin P^*$, $h \notin A$, $h \notin R^*$

Some exercises

Are the sets $(\tilde{A})^*$ and $(\tilde{A^*})$ always the same?

The predicate H <u>represents</u> A iff for every n,

$$H(n) \in \mathcal{P} \leftrightarrow n \in A$$

Prove that if R^* is representable and $\mathcal{L}$ is consistent, then $\mathcal{L}$ is incomplete.

Prove the following: If there is a representable set that contains R^* and disjoint from P^*, then $\mathcal{L}$ is incomplete.
Be A represented by H_h, $A \cap P^* = \emptyset$, $R^* \subseteq A$
$h \in P^* \leftrightarrow d(h) \in P \leftrightarrow H_h(h) \in \mathcal{P} \leftrightarrow h \in A$
Therefore, $h \notin P^*$, $h \notin A$, $h \notin R^*$
Consequently, $d(h)$, the Gödel-number of $H_h(h)$ is
$\notin P$ and $\notin R$.

Two exercises for homework

Two exercises for homework

Be $\mathcal{L}$ a correct system where P^* is expressible. Suppose the following condition holds:
For any predicate H there is a predicate H' s.t. for every n, $H'(n)$ is provable iff $H(n)$ is refutable.
Prove that $\mathcal{L}$ is incomplete.

Two exercises for homework

Be $\mathcal{L}$ a correct system where P^* is expressible. Suppose the following condition holds:
For any predicate H there is a predicate H' s.t. for every n, $H'(n)$ is provable iff $H(n)$ is refutable.
Prove that $\mathcal{L}$ is incomplete.

Suppose that the following conditions hold in $\mathcal{L}$:

- The predicate E_7 expresses P.
- If E_n is a predicate that names A then E_{3n} expresses $\tilde{A}$.
- If E_n is a predicate that names A then E_{3n+1} expresses A^*.

Two exercises for homework

Be $\mathcal{L}$ a correct system where P^* is expressible. Suppose the following condition holds:
For any predicate H there is a predicate H' s.t. for every n, $H'(n)$ is provable iff $H(n)$ is refutable.
Prove that $\mathcal{L}$ is incomplete.

Suppose that the following conditions hold in $\mathcal{L}$:

- The predicate E_7 expresses P.
- If E_n is a predicate that names A then E_{3n} expresses $\tilde{A}$.
- If E_n is a predicate that names A then E_{3n+1} expresses A^*.

1. Find numbers a and b s.t. $E_a(b)$ is true but not provable. Find the two solution for that both numbers are less than 100.

Two exercises for homework

Be $\mathcal{L}$ a correct system where P^* is expressible. Suppose the following condition holds:
For any predicate H there is a predicate H' s.t. for every n, $H'(n)$ is provable iff $H(n)$ is refutable.
Prove that $\mathcal{L}$ is incomplete.

Suppose that the following conditions hold in $\mathcal{L}$:

- The predicate E_7 expresses P.
- If E_n is a predicate that names A then E_{3n} expresses $\tilde{A}$.
- If E_n is a predicate that names A then E_{3n+1} expresses A^*.

1. Find numbers a and b s.t. $E_a(b)$ is true but not provable. Find the two solution for that both numbers are less than 100.
2. Prove that there are infinitely many such pairs $(a,\ b)$.

Two exercises for homework

Be $\mathcal{L}$ a correct system where P^* is expressible. Suppose the following condition holds:
For any predicate H there is a predicate H' s.t. for every n, $H'(n)$ is provable iff $H(n)$ is refutable.
Prove that $\mathcal{L}$ is incomplete.

Suppose that the following conditions hold in $\mathcal{L}$:

- The predicate E_7 expresses P.
- If E_n is a predicate that names A then E_{3n} expresses $\tilde{A}$.
- If E_n is a predicate that names A then E_{3n+1} expresses A^*.

1. Find numbers a and b s.t. $E_a(b)$ is true but not provable. Find the two solution for that both numbers are less than 100.
2. Prove that there are infinitely many such pairs $(a,\ b)$.
3. Suppose that E_{10} is a predicate. Find $(c,\ d)$ s.t. $E_c(d)$ is a Gödel sentence of the set named by E_{10}.

The language $\mathcal{L}_E$: Basic symbols

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0,\ {}'\ (,\ ),\ f,\ {}^{\prime},\ v,\ \neg,\ \rightarrow,\ \forall,\ =,\ \leq,\ \sharp$$

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0, \; {}' \; (, \;), \; f, \; \prime, \; v, \; \neg, \; \to, \; \forall, \; =, \; \leq, \; \sharp$$

Numerals:

$$0, \; 0', \; 0'', \; 0''', \ldots$$

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0,\ {}'\,(,\ ),\ f,\ {}_{'},\ v,\ \neg,\ \rightarrow,\ \forall,\ =,\ \leq,\ \sharp$$

Numerals:

$$0,\ 0',\ 0'',\ 0''',\dots$$

$f_{'}$, $f_{''}$, $f_{'''}$: function symbols with the following intended meanings and abbreviations:

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0,\ '\,(,\ ),\ f,\ {}_{\prime},\ v,\ \neg,\ \rightarrow,\ \forall,\ =,\ \leq,\ \sharp$$

Numerals:

$$0,\ 0',\ 0'',\ 0''',\ldots$$

$f_{\prime}$, $f_{\prime\prime}$, $f_{\prime\prime\prime}$: function symbols with the following intended meanings and abbreviations:

$f_{\prime}(xy):\ x+y$

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0,\ {}'\,(,\ ),\ f,\ {}_{\prime},\ v,\ \neg,\ \rightarrow,\ \forall,\ =,\ \leq,\ \sharp$$

Numerals:

$$0,\ 0',\ 0'',\ 0''',\ldots$$

$f_{\prime}$, $f_{\prime\prime}$, $f_{\prime\prime\prime}$: function symbols with the following intended meanings and abbreviations:

$f_{\prime}(xy):\ x+y$

$f_{\prime\prime}(xy):\ x\cdot y$

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0,\ {}'\ (,\ \ ),\ f,\ {}_{\prime},\ v,\ \neg,\ \rightarrow,\ \forall,\ =,\ \leq,\ \sharp$$

Numerals:

$$0,\ 0',\ 0'',\ 0''',\ldots$$

$f_{\prime}$, $f_{\prime\prime}$, $f_{\prime\prime\prime}$: function symbols with the following intended meanings and abbreviations:

$f_{\prime}(xy):\ x+y$

$f_{\prime\prime}(xy):\ x\cdot y$

$f_{\prime\prime\prime}(xy): x\mathbf{E}y(: x^y)$

The language $\mathcal{L}_E$: Basic symbols

Alphabet:

$$0,\ {}'\ (,\ \ ),\ f,\ {}_{\prime},\ v,\ \neg,\ \rightarrow,\ \forall,\ =,\ \leq,\ \sharp$$

Numerals:

$$0,\ 0',\ 0'',\ 0''',\ldots$$

$f_{\prime}$, $f_{\prime\prime}$, $f_{\prime\prime\prime}$: function symbols with the following intended meanings and abbreviations:

$f_{\prime}(xy):\ x+y$

$f_{\prime\prime}(xy):\ x\cdot y$

$f_{\prime\prime\prime}(xy): x\mathbf{E}y(: x^y)$

Variables: $v_1,\ v_2,\ v_3\ldots$, as abbreviations for:

$$(v_{\prime}),\ (v_{\prime\prime}),\ (v_{\prime\prime\prime})\ldots$$

Syntax

Terms:

Syntax

Terms:

1. Variables and numerals are terms.

Syntax

Terms:

1. Variables and numerals are terms.
2. $t_1{}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms:

1. Variables and numerals are terms.
2. ${t_1}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Syntax

Terms:

1. Variables and numerals are terms.
2. ${t_1}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

Syntax

Terms:

1. Variables and numerals are terms.
2. ${t_1}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.

Syntax

Terms:

1. Variables and numerals are terms.
2. $t_1{}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.
$\neg F$, $(F \to G)$ are formulas and for any i, $\forall v_i F$ is a formula, too.

Terms:

1. Variables and numerals are terms.
2. $t_1{}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.
$\neg F$, $(F \to G)$ are formulas and for any i, $\forall v_i F$ is a formula, too.

Free and bounded occurrences of a variable v_i:

Syntax

Terms:

1. Variables and numerals are terms.
2. t_1', $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.
$\neg F$, $(F \to G)$ are formulas and for any i, $\forall v_i F$ is a formula, too.

Free and bounded occurrences of a variable v_i:

- In terms and atomic formulas, every occurrence is free.

Syntax

Terms:

1. Variables and numerals are terms.
2. $t_1{}'$, $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.
$\neg F$, $(F \to G)$ are formulas and for any i, $\forall v_i F$ is a formula, too.

Free and bounded occurrences of a variable v_i:

- In terms and atomic formulas, every occurrence is free.
- In formulas $\neg F$ and $(F \to G)$ free occurrences are the same as in F resp. in F and G.

Syntax

Terms:

1. Variables and numerals are terms.
2. t_1', $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.
$\neg F$, $(F \to G)$ are formulas and for any i, $\forall v_i F$ is a formula, too.

Free and bounded occurrences of a variable v_i:

- In terms and atomic formulas, every occurrence is free.
- In formulas $\neg F$ and $(F \to G)$ free occurrences are the same as in F resp. in F and G.
- In $\forall v_i F$, v_i has no free occurrences and all of its occurrences are called bounded.

Syntax

Terms:

1. Variables and numerals are terms.
2. t_1', $(t_1 + t_2)$, $(t_1 \cdot t_2)$, $(t_1 \mathbf{E} t_2)$ are terms.

Terms containing no variables are called closed terms or constants.

Formulas:

$t_1 = t_2$ and $t_1 \leq t_2$ are atomic formulas.
$\neg F$, $(F \to G)$ are formulas and for any i, $\forall v_i F$ is a formula, too.

Free and bounded occurrences of a variable v_i:

- In terms and atomic formulas, every occurrence is free.
- In formulas $\neg F$ and $(F \to G)$ free occurrences are the same as in F resp. in F and G.
- In $\forall v_i F$, v_i has no free occurrences and all of its occurrences are called bounded.
- If $j \neq i$, every free occurrence of v_i in F remains free in $\forall v_j F$.

Syntax, continuation

Syntax, continuation

Formulas with no free occurrence of any variable are called sentences or closed formulas.

Syntax, continuation

Formulas with no free occurrence of any variable are called sentences or closed formulas.

Abbreviations (metalanguage names) for numerals: $\bar{n}$ for $0^{''\cdots'}$ if the number of the commas is n.

Syntax, continuation

Formulas with no free occurrence of any variable are called <u>sentences</u> or <u>closed formulas</u>.

Abbreviations (metalanguage names) for numerals: $\bar{n}$ for $0^{''\cdots'}$ if the number of the commas is n.

<u>Substitution</u> (numeral for variable):

Syntax, continuation

Formulas with no free occurrence of any variable are called sentences or closed formulas.

Abbreviations (metalanguage names) for numerals: $\bar{n}$ for $0''^{\cdots\prime}$ if the number of the commas is n.

Substitution (numeral for variable):

If $F(v_i)$ is a formula with the single variable with free occurrences (single free variable) v_i, then $F(\bar{n})$ is the sentence where all the free occurrences of v_i are substituted with $\bar{n}$.

Syntax, continuation

Formulas with no free occurrence of any variable are called sentences or closed formulas.

Abbreviations (metalanguage names) for numerals: $\bar{n}$ for $0^{''\cdots'}$ if the number of the commas is n.

Substitution (numeral for variable):

If $F(v_i)$ is a formula with the single variable with free occurrences (single free variable) v_i, then $F(\bar{n})$ is the sentence where all the free occurrences of v_i are substituted with $\bar{n}$.
If $F(v_{i_1}, v_{i_2}, \dots v_{i_n})$ is a formula all the free variables of which are $v_{i_1}, v_{i_2}, v_{i_n}$, then $F(\bar{k}_1, \bar{k}_2, \dots \bar{k}_n)$ is the sentence where every v_{i_j} is substituted with $\bar{k}_j$.

Syntax, continuation

Formulas with no free occurrence of any variable are called sentences or closed formulas.

Abbreviations (metalanguage names) for numerals: $\bar{n}$ for $0^{''\cdots'}$ if the number of the commas is n.

Substitution (numeral for variable):

If $F(v_i)$ is a formula with the single variable with free occurrences (single free variable) v_i, then $F(\bar{n})$ is the sentence where all the free occurrences of v_i are substituted with $\bar{n}$.
If $F(v_{i_1}, v_{i_2}, \ldots v_{i_n})$ is a formula all the free variables of which are $v_{i_1}, v_{i_2}, v_{i_n}$, then $F(\bar{k}_1, \bar{k}_2, \ldots \bar{k}_n)$ is the sentence where every v_{i_j} is substituted with $\bar{k}_j$.

$F(v_{i_1}, v_{i_2}, \ldots v_{i_n})$ is regular iff every v_{i_j} is v_j.

Additions to the syntax

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

We can prove theorems for every formula by induction on the degrees.

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

We can prove theorems for every formula by induction on the degrees.

We use the logical symbols $\vee$, $\wedge$, $\leftrightarrow$, $\exists$ as abbreviations on the usual way. Further abbreviations:

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

We can prove theorems for every formula by induction on the degrees.

We use the logical symbols $\vee$, $\wedge$, $\leftrightarrow$, $\exists$ as abbreviations on the usual way. Further abbreviations:

$t_1 \neq t_2$ for $\neg t_1 = t_2$;

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

We can prove theorems for every formula by induction on the degrees.

We use the logical symbols $\vee$, $\wedge$, $\leftrightarrow$, $\exists$ as abbreviations on the usual way. Further abbreviations:

$t_1 \neq t_2$ for $\neg t_1 = t_2$;

$t_1 < t_2$ for $t_1 \leq t_2 \wedge t_1 \neq t_2$;

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

We can prove theorems for every formula by induction on the degrees.

We use the logical symbols $\vee$, $\wedge$, $\leftrightarrow$, $\exists$ as abbreviations on the usual way. Further abbreviations:

$t_1 \neq t_2$ for $\neg t_1 = t_2$;

$t_1 < t_2$ for $t_1 \leq t_2 \wedge t_1 \neq t_2$;

$t_1^{t_2}$ for $t_1 \mathbf{E} t_2$;

Additions to the syntax

The degree of a formula is the number of logical constant occurrences contained.

We can prove theorems for every formula by induction on the degrees.

We use the logical symbols $\vee$, $\wedge$, $\leftrightarrow$, $\exists$ as abbreviations on the usual way. Further abbreviations:

$t_1 \neq t_2$ for $\neg t_1 = t_2$;

$t_1 < t_2$ for $t_1 \leq t_2 \wedge t_1 \neq t_2$;

$t_1^{t_2}$ for $t_1 \mathbf{E} t_2$;

$(\forall v_i \leq t)F$ for $\forall v_i(v_i \leq t \to F)$ and limited existential quantification on the similar way.

Finishing the language $\mathcal{L}_E$

Finishing the language $\mathcal{L}_E$

Substitution (variable for variable)

Substitution (variable for variable)

Let $F(v_1)$ be a formula with one free variable. $F(v_i)$ $(i \neq 1)$ is the following formula:

Finishing the language $\mathcal{L}_E$

Substitution (variable for variable)

Let $F(v_1)$ be a formula with one free variable. $F(v_i)$ $(i \neq 1)$ is the following formula:

- If v_i has no bounded occurrences in $F(v_1)$, the formula where every free occurrence of v_1 is substituted by v_i;

Finishing the language $\mathcal{L}_E$

Substitution (variable for variable)

Let $F(v_1)$ be a formula with one free variable. $F(v_i)$ $(i \neq 1)$ is the following formula:

- If v_i has no bounded occurrences in $F(v_1)$, the formula where every free occurrence of v_1 is substituted by v_i;
- If v_i does occur in $F(v_1)$, then take the smallest j s.t. v_j does not occur in $F(v_1)$, substitute every (bounded) occurrence of v_i with v_j, and then proceed with the resulting formula as in the first clause.

Finishing the language $\mathcal{L}_E$

Substitution (variable for variable)

Let $F(v_1)$ be a formula with one free variable. $F(v_i)$ $(i \neq 1)$ is the following formula:

- If v_i has no bounded occurrences in $F(v_1)$, the formula where every free occurrence of v_1 is substituted by v_i;
- If v_i does occur in $F(v_1)$, then take the smallest j s.t. v_j does not occur in $F(v_1)$, substitute every (bounded) occurrence of v_i with v_j, and then proceed with the resulting formula as in the first clause.

With n-variable regular formulas $F(v_1, v_2, \ldots v_n)$ and substituting variables $v_{i_1}, v_{i_2}, \ldots v_{i_n}$, the procedure is similar.

Denotation and truth in $\mathcal{L}_E$

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.
If c denotes n, then c' denotes $n + 1$; $(c_1 + c_2)$ denotes the sum of of the numbers denoted by c_1 and c_2, etc.

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.
If c denotes n, then c' denotes $n+1$; $(c_1 + c_2)$ denotes the sum of of the numbers denoted by c_1 and c_2, etc.

We have a similarly trivial definition of truth for the **sentences** of $\mathcal{L}_E$.

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.
If c denotes n, then c' denotes $n+1$; $(c_1 + c_2)$ denotes the sum of of the numbers denoted by c_1 and c_2, etc.

We have a similarly trivial definition of truth for the **sentences** of $\mathcal{L}_E$.
$c_1 = c_2$ is true iff c_1 and c_2 denote the same number; $c_1 \leq c_2$ is true iff c_1 denotes a number less or equal than the number denoted by c_2.

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.
If c denotes n, then c' denotes $n+1$; $(c_1 + c_2)$ denotes the sum of of the numbers denoted by c_1 and c_2, etc.

We have a similarly trivial definition of truth for the **sentences** of $\mathcal{L}_E$.
$c_1 = c_2$ is true iff c_1 and c_2 denote the same number; $c_1 \leq c_2$ is true iff c_1 denotes a number less or equal than the number denoted by c_2.
Logical constants work on the usual way. E.g.:

Denotation and truth in $\mathcal{L}_E$

We can define the denotation of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.
If c denotes n, then c' denotes $n+1$; $(c_1 + c_2)$ denotes the sum of of the numbers denoted by c_1 and c_2, etc.

We have a similarly trivial definition of truth for the **sentences** of $\mathcal{L}_E$.
$c_1 = c_2$ is true iff c_1 and c_2 denote the same number; $c_1 \leq c_2$ is true iff c_1 denotes a number less or equal than the number denoted by c_2.
Logical constants work on the usual way. E.g.:
$\forall v_i F$ is true iff for every number n, $F(\bar{n})$ is true.

Denotation and truth in $\mathcal{L}_E$

We can define the <u>denotation</u> of a constant term of $\mathcal{L}_E$ on the trivial way:
The numeral $\bar{n}$ denotes the number n.
If c denotes n, then c' denotes $n+1$; $(c_1 + c_2)$ denotes the sum of of the numbers denoted by c_1 and c_2, etc.

We have a similarly trivial <u>definition of truth</u> for the **sentences** of $\mathcal{L}_E$.
$c_1 = c_2$ is true iff c_1 and c_2 denote the same number; $c_1 \leq c_2$ is true iff c_1 denotes a number less or equal than the number denoted by c_2.
Logical constants work on the usual way. E.g.:
$\forall v_i F$ is true iff for every number n, $F(\bar{n})$ is true.
$F(\bar{n})$ is of lower degree than $\forall v_i F$, therefore induction guarantees that this definition works.