The incompleteness of the theori(es) of canonical calculi

András Máté

21.11.2025

Where we are

We did define the followings:

Where we are

We did define the followings:

- A first-order language $\mathcal{L}^{1*}$ in which

Where we are

We did define the followings:

- A first-order language $\mathcal{L}^{1*}$ in which
  - name constants denote autonymously the letters of $\mathcal{A}_{cc}$ plus ϑ denotes the empty string;
  - the empty string denotes the concatenation function;
  - predicate letters denote string properties and relations like being (a code of) a calculus or the substitution relation between a calculus variable, a word and two strings.

Where we are

We did define the followings:

- A first-order language $\mathcal{L}^{1*}$ in which
  - name constants denote autonymously the letters of $\mathcal{A}_{cc}$ plus ϑ denotes the empty string;
  - the empty string denotes the concatenation function;
  - predicate letters denote string properties and relations like being (a code of) a calculus or the substitution relation between a calculus variable, a word and two strings.
- The first-order theory $\mathbf{CC}^*$ of canonical calculi in $\mathcal{L}^{1*}$.

Where we are

We did define the followings:

- A first-order language $\mathcal{L}^{1*}$ in which
  - name constants denote autonymously the letters of $\mathcal{A}_{cc}$ plus ϑ denotes the empty string;
  - the empty string denotes the concatenation function;
  - predicate letters denote string properties and relations like being (a code of) a calculus or the substitution relation between a calculus variable, a word and two strings.
- The first-order theory $\mathbf{CC}^*$ of canonical calculi in $\mathcal{L}^{1*}$.
- The canonical calculus Σ^* which generates the theorems of $\mathbf{CC}^*$.

Denotation and truth in $\mathcal{L}^{1*}$

Denotation and truth in $\mathcal{L}^{1*}$

We can give a truth definition for $\mathcal{L}^{1*}$-formulas, independently from any set-theoretical semantics.

Denotation and truth in $\mathcal{L}^{1*}$

We can give a truth definition for $\mathcal{L}^{1*}$-formulas, independently from any set-theoretical semantics.

The closed terms of $\mathcal{L}^{1*}$ are $\mathcal{A}_{cc} \cup \{\vartheta\}$-strings. They denote $\mathcal{A}_{cc}$-strings that we obtain from them by deleting the ϑs.

Denotation and truth in $\mathcal{L}^{1*}$

We can give a truth definition for $\mathcal{L}^{1*}$-formulas, independently from any set-theoretical semantics.

The closed terms of $\mathcal{L}^{1*}$ are $\mathcal{A}_{cc} \cup \{\vartheta\}$-strings. They denote $\mathcal{A}_{cc}$-strings that we obtain from them by deleting the ϑs.

- A closed atomic formula $\ulcorner s = t \urcorner$ is true iff 's' and 't' denote the same string.

Denotation and truth in $\mathcal{L}^{1*}$

We can give a truth definition for $\mathcal{L}^{1*}$-formulas, independently from any set-theoretical semantics.

The closed terms of $\mathcal{L}^{1*}$ are $\mathcal{A}_{cc} \cup \{\vartheta\}$-strings. They denote $\mathcal{A}_{cc}$-strings that we obtain from them by deleting the ϑs.

- A closed atomic formula $\ulcorner s = t \urcorner$ is true iff 's' and 't' denote the same string.
- Closed atomic formulas containing the predicates I, L, W, V, T, R, K, F, S are true iff they are true according to the intended interpretation. I.e., $\ulcorner I(s) \urcorner$ is true iff the string s is an index, $\ulcorner K(s) \urcorner$ is true iff s is a code of a calculus, $\ulcorner S(s)(t)(v)(u) \urcorner$ is true iff by substituting the word (variable-free string) v for the variable u in the string t, we get s, etc.

Denotation and truth in $\mathcal{L}^{1*}$

We can give a truth definition for $\mathcal{L}^{1*}$-formulas, independently from any set-theoretical semantics.

The closed terms of $\mathcal{L}^{1*}$ are $\mathcal{A}_{cc} \cup \{\vartheta\}$-strings. They denote $\mathcal{A}_{cc}$-strings that we obtain from them by deleting the ϑs.

- A closed atomic formula $\ulcorner s = t \urcorner$ is true iff 's' and 't' denote the same string.
- Closed atomic formulas containing the predicates I, L, W, V, T, R, K, F, S are true iff they are true according to the intended interpretation. I.e., $\ulcorner I(s) \urcorner$ is true iff the string s is an index, $\ulcorner K(s) \urcorner$ is true iff s is a code of a calculus, $\ulcorner S(s)(t)(v)(u) \urcorner$ is true iff by substituting the word (variable-free string) v for the variable u in the string t, we get s, etc.

These stipulations are effective, so the reference to the intended interpretation is not problematic.

Truth definition (continuation)

Truth definition (continuation)

- $\ulcorner D(s)(t) \urcorner$ is true iff $\mathbf{H}_3 \mapsto s'Dt'$ where s' and t' are the strings denoted by s resp. t, i.e. the calculus encoded by s derives the string t.

Truth definition (continuation)

- $\ulcorner D(s)(t) \urcorner$ is true iff $\mathbf{H}_3 \mapsto s' D t'$ where s' and t' are the strings denoted by s resp. t, i.e. the calculus encoded by s derives the string t.
- $\ulcorner A(s) \urcorner$ is true iff $\mathbf{H}_3 \mapsto A s'$, i. e. s is an autonomous numeral.

Truth definition (continuation)

- $\ulcorner D(s)(t) \urcorner$ is true iff $\mathbf{H}_3 \mapsto s'Dt'$ where s' and t' are the strings denoted by s resp. t, i.e. the calculus encoded by s derives the string t.
- $\ulcorner A(s) \urcorner$ is true iff $\mathbf{H}_3 \mapsto As'$, i. e. s is an autonomous numeral.

We could use the method of these stipulations at the first two items, too. But these last ones are not effective (just this is our point).

Truth definition (continuation)

- $\ulcorner D(s)(t) \urcorner$ is true iff $\mathbf{H}_3 \mapsto s'Dt'$ where s' and t' are the strings denoted by s resp. t, i.e. the calculus encoded by s derives the string t.
- $\ulcorner A(s) \urcorner$ is true iff $\mathbf{H}_3 \mapsto As'$, i. e. s is an autonomous numeral.

We could use the method of these stipulations at the first two items, too. But these last ones are not effective (just this is our point).

- The evaluation of negation and conditional goes as usual.

Truth definition (continuation)

- $\ulcorner D(s)(t) \urcorner$ is true iff $\mathbf{H}_3 \mapsto s' D t'$ where s' and t' are the strings denoted by s resp. t, i.e. the calculus encoded by s derives the string t.
- $\ulcorner A(s) \urcorner$ is true iff $\mathbf{H}_3 \mapsto As'$, i. e. s is an autonomous numeral.

We could use the method of these stipulations at the first two items, too. But these last ones are not effective (just this is our point).

- The evaluation of negation and conditional goes as usual.
- Truth condition for universal quantification is given by *substitution* (that's why we don't need set theory): $\ulcorner \forall x A \urcorner$ is false iff for some t $\mathcal{A}_{cc}$-string, $[A]^{t/x}$ is false, and true in the other case. (Remark: x is a metalanguage variable here running on the $\mathcal{L}^{1*}$-variables $\mathfrak{x}_n$.)

Truth definition (continuation)

- $\ulcorner D(s)(t) \urcorner$ is true iff $\mathbf{H}_3 \mapsto s'Dt'$ where s' and t' are the strings denoted by s resp. t, i.e. the calculus encoded by s derives the string t.
- $\ulcorner A(s) \urcorner$ is true iff $\mathbf{H}_3 \mapsto As'$, i. e. s is an autonomous numeral.

We could use the method of these stipulations at the first two items, too. But these last ones are not effective (just this is our point).

- The evaluation of negation and conditional goes as usual.
- Truth condition for universal quantification is given by *substitution* (that's why we don't need set theory): $\ulcorner \forall x A \urcorner$ is false iff for some t $\mathcal{A}_{cc}$-string, $[A]^{t/x}$ is false, and true in the other case. (Remark: x is a metalanguage variable here running on the $\mathcal{L}^{1*}$-variables $\mathfrak{x}_n$.)
- Open formulas of $\mathcal{L}^{1*}$ are true iff their universal closure is true.

The consistency of **CC***

Theorem: All the theorems of **CC*** are true according to the above truth definition.

Theorem: All the theorems of $\mathbf{CC}^*$ are true according to the above truth definition.

The proof goes by induction following the inductive definition of $\Gamma \vdash A$ (previous presentation): the axioms of $\mathbf{CC}^*$ are true (simple calculation), the basic formulas of first-order logic (of $\mathcal{L}^{1*}$) are true and detachment preserves truth.

The consistency of $\mathbf{CC}^*$

Theorem: All the theorems of $\mathbf{CC}^*$ are true according to the above truth definition.

The proof goes by induction following the inductive definition of $\Gamma \vdash A$ (previous presentation): the axioms of $\mathbf{CC}^*$ are true (simple calculation), the basic formulas of first-order logic (of $\mathcal{L}^{1*}$) are true and detachment preserves truth.

Corollary: $\mathbf{CC}^*$ is consistent. Because there are false sentences of $\mathcal{L}^{1*}$ (e.g., '$\alpha = \beta$'), and according to the theorem, they are not provable.

The consistency of $\mathbf{CC}^*$

Theorem: All the theorems of $\mathbf{CC}^*$ are true according to the above truth definition.

The proof goes by induction following the inductive definition of $\Gamma \vdash A$ (previous presentation): the axioms of $\mathbf{CC}^*$ are true (simple calculation), the basic formulas of first-order logic (of $\mathcal{L}^{1*}$) are true and detachment preserves truth.

Corollary: $\mathbf{CC}^*$ is consistent. Because there are false sentences of $\mathcal{L}^{1*}$ (e.g., '$\alpha = \beta$'), and according to the theorem, they are not provable.

Theorem: If $\mathbf{H}_3 \mapsto f$, then $Tr(f)$ is provable in $\mathbf{CC}^*$.
The proof goes by induction following the inductive definition of strings derivable in $\mathbf{H}_3$.

Undecidability

Theorem: $\mathbf{CC}^*$ is not decidable.
Suppose we have an algorithm to decide which sentences of $\mathcal{L}^{1*}$ are theorems of $\mathbf{CC}^*$. In this case, we could decide which sentences of the form $A(c)$ (where c is a numeral) are theorems. But this would mean that we could decide which numerals are autonomous - in contradiction to our earlier result that the class of autonomous numerals is not decidable.

Theorem: $\mathbf{CC}^*$ is not decidable.
Suppose we have an algorithm to decide which sentences of $\mathcal{L}^{1*}$ are theorems of $\mathbf{CC}^*$. In this case, we could decide which sentences of the form $A(c)$ (where c is a numeral) are theorems. But this would mean that we could decide which numerals are autonomous - in contradiction to our earlier result that the class of autonomous numerals is not decidable.

Theorem(Church-Turing-Markov): First-order logic is not decidable.
I. e., there is no algorithm for every first-order language that decides about every formula whether it is a logical truth (consequence of the empty set of formulas) or not.
E.g., for $\mathcal{L}^{1*}$ there is no such algorithm. Because otherwise we had an algorithm to decide which formulas of the form $\mathbf{Ax} \supset A(c)$ are logical truths (where $\mathbf{Ax}$ is the conjunction of all axioms of $\mathbf{CC}^*$ and c is a numeral). This would imply the decidability of the class of autonomous numerals again.

Negation completeness

Negation completeness

A theory is omniscient (knows everything about its subject) if it can decide every question that can be formulated in the language of the theory.

Negation completeness

A theory is omniscient (knows everything about its subject) if it can decide every question that can be formulated in the language of the theory.

Formal counterpart: a formal theory is negation complete iff for every sentence A of its language, either A or $\neg A$ is a theorem. Otherwise, it's called (negation) incomplete.

Negation completeness

A theory is omniscient (knows everything about its subject) if it can decide every question that can be formulated in the language of the theory.

Formal counterpart: a formal theory is negation complete iff for every sentence A of its language, either A or $\neg A$ is a theorem. Otherwise, it's called (negation) incomplete.

Inconsistent theories are negation complete but uninteresting.

Negation completeness

A theory is omniscient (knows everything about its subject) if it can decide every question that can be formulated in the language of the theory.

Formal counterpart: a formal theory is negation complete iff for every sentence A of its language, either A or $\neg A$ is a theorem. Otherwise, it's called (negation) incomplete.

Inconsistent theories are negation complete but uninteresting.

A plausible idea: if an axiomatic theory proves to be negation incomplete, then it was too weak. Let us extend the set of axioms by some true sentences.

Negation completeness

A theory is omniscient (knows everything about its subject) if it can decide every question that can be formulated in the language of the theory.

Formal counterpart: a formal theory is negation complete iff for every sentence A of its language, either A or $\neg A$ is a theorem. Otherwise, it's called (negation) incomplete.

Inconsistent theories are negation complete but uninteresting.

A plausible idea: if an axiomatic theory proves to be negation incomplete, then it was too weak. Let us extend the set of axioms by some true sentences.

There are very weak but negation complete theories, e.g. the theory of two-member world.

Negation completeness

A theory is omniscient (knows everything about its subject) if it can decide every question that can be formulated in the language of the theory.

Formal counterpart: a formal theory is negation complete iff for every sentence A of its language, either A or $\neg A$ is a theorem. Otherwise, it's called (negation) incomplete.

Inconsistent theories are negation complete but uninteresting.

A plausible idea: if an axiomatic theory proves to be negation incomplete, then it was too weak. Let us extend the set of axioms by some true sentences.

There are very weak but negation complete theories, e.g. the theory of two-member world.

The interesting case is when a theory is incomplete because it is too strong, and therefore the incompleteness cannot be remedied by extending the theory.

The calculus Σ^* and its own code σ^*

The calculus Σ^* and its own code σ^*

If $D(a)(b)$ is provable in $\mathbf{CC}^*$, it means that the calculus *translated by* the string a derives the string translated by the string b. Method of translation: see 3rd October presentation.

The calculus Σ^* and its own code σ^*

If $D(a)(b)$ is provable in $\mathbf{CC}^*$, it means that the calculus *translated by* the string a derives the string translated by the string b. Method of translation: see 3rd October presentation. The theorems were generated by the canonical calculus Σ^* (last week). The alphabet of Σ^* can be encoded again in the two-letter alphabet $\mathcal{A}_1 = \{\alpha, \beta\}$ and the whole calculus will be encoded/translated as an $\mathcal{A}_{cc}$-string σ^*.
$(\mathcal{A}_{cc} = \{\alpha,\ \beta,\ \xi,\ \gg,\ *\})$.

The calculus Σ^* and its own code σ^*

If $D(a)(b)$ is provable in $\mathbf{CC}^*$, it means that the calculus *translated by* the string a derives the string translated by the string b. Method of translation: see 3rd October presentation. The theorems were generated by the canonical calculus Σ^* (last week). The alphabet of Σ^* can be encoded again in the two-letter alphabet $\mathcal{A}_1 = \{\alpha, \beta\}$ and the whole calculus will be encoded/translated as an $\mathcal{A}_{cc}$-string σ^*.
$(\mathcal{A}_{cc} = \{\alpha,\ \beta,\ \xi,\ \gg,\ *\})$.
$\mathbf{CC}^*$ proves among others propositions of the form $D(\sigma^*)(b)$ which means that $\mathbf{CC}^*$ proves a proposition encoded by the string b. This fact gives us the possibility to *diagonalize* the theory.

Circularity is not always vicious, but some caution is needed

Circularity is not always vicious, but some caution is needed

Remember: the theorems of $\mathbf{CC}^*$ are generated by the calculus Σ^*. Its auxiliary letters partly overlap in meaning with the auxiliary letters of $\mathbf{H}_3$ and therefore with the non-logical constants of $\mathcal{L}^{1*}$ Because of this, we will use the same letter (V for variable, T for term, F for formula, etc.). But to avoid ambiguity, the auxiliary letters of Σ^* written in boldface.

Circularity is not always vicious, but some caution is needed

Remember: the theorems of $\mathbf{CC}^*$ are generated by the calculus Σ^*. Its auxiliary letters partly overlap in meaning with the auxiliary letters of $\mathbf{H}_3$ and therefore with the non-logical constants of $\mathcal{L}^{1*}$ Because of this, we will use the same letter (V for variable, T for term, F for formula, etc.). But to avoid ambiguity, the auxiliary letters of Σ^* written in boldface.
To produce the code σ^* of Σ^* (and other codes of the expressions occurring in it), we need to translate each letter of its alphabet into $\mathcal{A}_1^\circ$. The translation of any letter C will be denoted by C'. Therefore we will see S as expressing the four-argument substitution relation in $\mathcal{L}^{1*}$, $\mathbf{S}$ as an auxiliary letter expressing substitution in Σ^* and $\mathbf{S}'$ as the code of $\mathbf{S}$.

The theory **CC**

The theory **CC**

CC comes from the theory **CC*** by deleting some predicates from the language $\mathcal{L}^{1*}$ and the axioms belonging to them from the axioms and adding one more auxiliary axiom called SUD (next slide).

CC comes from the theory **CC*** by deleting some predicates from the language $\mathcal{L}^{1*}$ and the axioms belonging to them from the axioms and adding one more auxiliary axiom called SUD (next slide).

In some details:
The language $\mathcal{L}^{10}$ of **CC** is the same as $\mathcal{L}^{1*}$ except of that it does not contain the two-place predicates F and G and the one-place predicate A.

CC comes from the theory $\mathbf{CC}^*$ by deleting some predicates from the language $\mathcal{L}^{1*}$ and the axioms belonging to them from the axioms and adding one more auxiliary axiom called SUD (next slide).

In some details:
The language $\mathcal{L}^{10}$ of **CC** is the same as $\mathcal{L}^{1*}$ except of that it does not contain the two-place predicates F and G and the one-place predicate A.

The class of axioms Γ_0 of **CC** comes from the axioms of $\mathbf{CC}^*$ by omitting the last nine axioms corresponding the rules 26.-34. of $\mathbf{H}_3$ (i.e, it contains the axioms that translate the rules of $\mathbf{H}_2$ but not the further rules of $\mathbf{H}_3$ governing the predicates omitted) and by adding SUD.

CC and σ

We can apply the previous truth definition. We will see that SUD is true according to this definition, too. Therefore, the theorems of **CC** are all true and the theory is consistent.

CC and σ

We can apply the previous truth definition. We will see that SUD is true according to this definition, too. Therefore, the theorems of **CC** are all true and the theory is consistent. The calculus producing the theorems of **CC** is Σ and its code in $\mathcal{L}^{10}$ is σ. The truth definiton for the formulas of $\mathcal{L}^{10}$ is the same as the truth definition for them in $\mathcal{L}^{1*}$. The notational conventions introduced previously also remain valid.

We can apply the previous truth definition. We will see that SUD is true according to this definition, too. Therefore, the theorems of **CC** are all true and the theory is consistent. The calculus producing the theorems of **CC** is Σ and its code in $\mathcal{L}^{10}$ is σ. The truth definiton for the formulas of $\mathcal{L}^{10}$ is the same as the truth definition for them in $\mathcal{L}^{1*}$. The notational conventions introduced previously also remain valid.

The axioms are just the axioms of **CC*** minus the axioms concerning the omitted predicates plus the axiom SUD (*S*ubstitution *U*niquely *D*eterminded):

$$\forall\mathfrak{x}_1\forall\mathfrak{x}_2\forall\mathfrak{x}_3\forall\mathfrak{x}_4$$
$$(D(\sigma)(\mathfrak{x}_3\mathbf{S}'\mathfrak{x}_2\mathbf{S}'\mathfrak{x}_1\mathbf{S}'\mathfrak{x}) \supset D(\sigma)(\mathfrak{x}_4\mathbf{S}'\mathfrak{x}_2\mathbf{S}'\mathfrak{x}_1\mathbf{S}'\mathfrak{x}) \supset \mathfrak{x}_3 = \mathfrak{x}_4)$$

CC and σ

We can apply the previous truth definition. We will see that SUD is true according to this definition, too. Therefore, the theorems of **CC** are all true and the theory is consistent. The calculus producing the theorems of **CC** is Σ and its code in $\mathcal{L}^{10}$ is σ. The truth definiton for the formulas of $\mathcal{L}^{10}$ is the same as the truth definition for them in $\mathcal{L}^{1*}$. The notational conventions introduced previously also remain valid.

The axioms are just the axioms of **CC*** minus the axioms concerning the omitted predicates plus the axiom SUD (Substitution Uniquely Determinded):

$$\forall \mathfrak{x}_1 \forall \mathfrak{x}_2 \forall \mathfrak{x}_3 \forall \mathfrak{x}_4$$
$$(D(\sigma)(\mathfrak{x}_3 \mathbf{S}' \mathfrak{x}_2 \mathbf{S}' \mathfrak{x}_1 \mathbf{S}' \mathfrak{x}) \supset D(\sigma)(\mathfrak{x}_4 \mathbf{S}' \mathfrak{x}_2 \mathbf{S}' \mathfrak{x}_1 \mathbf{S}' \mathfrak{x}) \supset \mathfrak{x}_3 = \mathfrak{x}_4)$$

It follows from the truth definition above that this axiom is true.

Diagonalization in **CC** (preparatory steps)

We have seen that all the theorems of **CC*** are true. Consequently, the theorems of **CC** are also. The converse of this latter statement – that every true closed formula is provable – would be the completeness statement for **CC**. We will prove the falsity of this statement roughly by the standard Gödelian methods. At first, we show that the simplest true propositions are provable.

Diagonalization in **CC** (preparatory steps)

We have seen that all the theorems of $\mathbf{CC}^*$ are true. Consequently, the theorems of **CC** are also. The converse of this latter statement – that every true closed formula is provable – would be the completeness statement for **CC**. We will prove the falsity of this statement roughly by the standard Gödelian methods. At first, we show that the simplest true propositions are provable.

Lemma 1.: The true *closed atomic formulas* of $\mathcal{L}^{1*}$ resp. $\mathcal{L}^{10}$ are all provable in $\mathbf{CC}^*$ resp. in **CC**.

We have seen that all the theorems of $\mathbf{CC}^*$ are true. Consequently, the theorems of **CC** are also. The converse of this latter statement – that every true closed formula is provable – would be the completeness statement for **CC**. We will prove the falsity of this statement roughly by the standard Gödelian methods. At first, we show that the simplest true propositions are provable.

Lemma 1.: The true *closed atomic formulas* of $\mathcal{L}^{1*}$ resp. $\mathcal{L}^{10}$ are all provable in $\mathbf{CC}^*$ resp. in **CC**.

See the truth definition for $\mathcal{L}^{1*}$.

Diagonalization in **CC** (preparatory steps)

We have seen that all the theorems of **CC*** are true. Consequently, the theorems of **CC** are also. The converse of this latter statement – that every true closed formula is provable – would be the completeness statement for **CC**. We will prove the falsity of this statement roughly by the standard Gödelian methods. At first, we show that the simplest true propositions are provable.

Lemma 1.: The true *closed atomic formulas* of $\mathcal{L}^{1*}$ resp. $\mathcal{L}^{10}$ are all provable in **CC*** resp. in **CC**.

See the truth definition for $\mathcal{L}^{1*}$.

Lemma 2.: If a string f is derivable in Σ, then $\sigma D f'$ is derivable in $\mathbf{H}_2$. Therefore, $D(\sigma)(f')$ is a true atomic formula of $\mathcal{L}^{10}$. According to Lemma 1., it is a theorem of **CC**.

The diagonal formula

The diagonal formula

Be A a formula with at most one variable x of the language $\mathcal{L}^{10}$, $A' = a$, $B = A^{a/x}$ and $B' = b$.

The diagonal formula

Be A a formula with at most one variable x of the language $\mathcal{L}^{10}$, $A' = a$, $B = A^{a/x}$ and $B' = b$.

Let us assume first that B is provable in **CC**. In this case, Σ derives the words $\mathbf{F}A$, $B\mathbf{S}A\mathbf{S}a\mathbf{S}x$, B.

The diagonal formula

Be A a formula with at most one variable x of the language $\mathcal{L}^{10}$, $A' = a$, $B = A^{a/x}$ and $B' = b$.

Let us assume first that B is provable in **CC**. In this case, Σ derives the words $\mathbf{F}A$, $B\mathbf{S}A\mathbf{S}a\mathbf{S}x$, B.

Therefore by Lemma 2., the following atomic formulas are theorems of **CC**: $D(\sigma)(\mathbf{F}'a)$, $D(\sigma)(b\mathbf{S}'a\mathbf{S}'a'\mathbf{S}'x')$, $D(\sigma)(b)$.

The diagonal formula

Be A a formula with at most one variable x of the language $\mathcal{L}^{10}$, $A' = a$, $B = A^{a/x}$ and $B' = b$.

Let us assume first that B is provable in **CC**. In this case, Σ derives the words $\mathbf{F}A$, $B\mathbf{S}A\mathbf{S}a\mathbf{S}x$, B.

Therefore by Lemma 2., the following atomic formulas are theorems of **CC**: $D(\sigma)(\mathbf{F}'a)$, $D(\sigma)(b\mathbf{S}'a\mathbf{S}'a'\mathbf{S}'x')$, $D(\sigma)(b)$.

Let us abbreviate their conjunction by $Diag_\sigma(a, b)$. If B was a theorem in **CC**, then this diagonal formula is a theorem, too.

Be A a formula with at most one variable x of the language $\mathcal{L}^{10}$, $A' = a$, $B = A^{a/x}$ and $B' = b$.

Let us assume first that B is provable in **CC**. In this case, Σ derives the words $\mathbf{F}A$, $B\mathbf{S}A\mathbf{S}a\mathbf{S}x$, B.

Therefore by Lemma 2., the following atomic formulas are theorems of **CC**: $D(\sigma)(\mathbf{F}'a)$, $D(\sigma)(b\mathbf{S}'a\mathbf{S}'a'\mathbf{S}'x')$, $D(\sigma)(b)$.

Let us abbreviate their conjunction by $Diag_\sigma(a,b)$. If B was a theorem in **CC**, then this diagonal formula is a theorem, too.

Let us now assume that $Diag_\sigma(a,b)$ is a theorem. Then each conjunct is a theorem, too, so they are true according our truth definition. The third conjunct says that the calculus with the code σ derives the string with the code b, i.e., B is a theorem of **CC**.

Be A a formula with at most one variable x of the language $\mathcal{L}^{10}$, $A' = a$, $B = A^{a/x}$ and $B' = b$.

Let us assume first that B is provable in **CC**. In this case, Σ derives the words $\mathbf{F}A$, $B\mathbf{S}A\mathbf{S}a\mathbf{S}x$, B.

Therefore by Lemma 2., the following atomic formulas are theorems of **CC**: $D(\sigma)(\mathbf{F}'a)$, $D(\sigma)(b\mathbf{S}'a\mathbf{S}'a'\mathbf{S}'x')$, $D(\sigma)(b)$.

Let us abbreviate their conjunction by $Diag_\sigma(a, b)$. If B was a theorem in **CC**, then this diagonal formula is a theorem, too.

Let us now assume that $Diag_\sigma(a, b)$ is a theorem. Then each conjunct is a theorem, too, so they are true according our truth definition. The third conjunct says that the calculus with the code σ derives the string with the code b, i.e., B is a theorem of **CC**.

Now we have proven

Lemma 3. B is a theorem of **CC** iff $Diag_\sigma(a, b)$ is a theorem.

The Gödel sentence and its unprovability

The Gödel sentence and its unprovability

Be A_0 the following formula with the code a_0:

$$\forall \mathfrak{x}_1 \neg Diag_\sigma(\mathfrak{x}, \mathfrak{x}_1).$$

Be A_0 the following formula with the code a_0:

$$\forall \mathfrak{x}_1 \neg Diag_\sigma(\mathfrak{x}, \mathfrak{x}_1).$$

Let us diagonalize it and call the diagonalized formula G with the code g:

$$G = \forall \mathfrak{x}_1 \neg Diag_\sigma(a_0, \mathfrak{x}_1).$$

The Gödel sentence and its unprovability

Be A_0 the following formula with the code a_0:

$$\forall \mathfrak{x}_1 \neg Diag_\sigma(\mathfrak{x}, \mathfrak{x}_1).$$

Let us diagonalize it and call the diagonalized formula G with the code g:

$$G = \forall \mathfrak{x}_1 \neg Diag_\sigma(a_0, \mathfrak{x}_1).$$

According to Lemma 3., G is a theorem of **CC** iff $Diag_\sigma(a_0, g)$ is a theorem.

The Gödel sentence and its unprovability

Be A_0 the following formula with the code a_0:

$$\forall \mathfrak{x}_1 \neg Diag_\sigma(\mathfrak{x}, \mathfrak{x}_1).$$

Let us diagonalize it and call the diagonalized formula G with the code g:

$$G = \forall \mathfrak{x}_1 \neg Diag_\sigma(a_0, \mathfrak{x}_1).$$

According to Lemma 3., G is a theorem of **CC** iff $Diag_\sigma(a_0, g)$ is a theorem.

But from G follows $\neg Diag_\sigma(a_0, g)$. Therefore, if G is a theorem, then **CC** is inconsistent. Hence, G is not a theorem.

The truth of the Gödel sentence

The truth of the Gödel sentence

Suppose that G is false. Then there is a b_0 such that the closed atomic formula $Diag_\sigma(a_0, b_0)$ is true and hence provable.

Suppose that G is false. Then there is a b_0 such that the closed atomic formula $Diag_\sigma(a_0, b_0)$ is true and hence provable.

Therefore the conjuncts $D(\sigma)(\mathbf{F}'a_0)$, $D(\sigma)(b_0\mathbf{S}'a_0\mathbf{S}'[a_0]'\mathbf{S}'x')$, $D(\sigma)(b_0)$ are all true.

Suppose that G is false. Then there is a b_0 such that the closed atomic formula $Diag_\sigma(a_0, b_0)$ is true and hence provable.

Therefore the conjuncts
$D(\sigma)(\mathbf{F}'a_0)$, $D(\sigma)(b_0\mathbf{S}'a_0\mathbf{S}'[a_0]'\mathbf{S}'x')$, $D(\sigma)(b_0)$ are all true.

From the second conjunct follows that b_0 cannot be different from g because the result of substituting the code a_0 into the formula with the code a_0 is the formula with the code g.

Suppose that G is false. Then there is a b_0 such that the closed atomic formula $Diag_\sigma(a_0, b_0)$ is true and hence provable.

Therefore the conjuncts $D(\sigma)(\mathbf{F}'a_0)$, $D(\sigma)(b_0\mathbf{S}'a_0\mathbf{S}'[a_0]'\mathbf{S}'x')$, $D(\sigma)(b_0)$ are all true.

From the second conjunct follows that b_0 cannot be different from g because the result of substituting the code a_0 into the formula with the code a_0 is the formula with the code g.

Therefore, $D(\sigma)(g)$ is true. But it means that the formula with the code g – i.e., G itself – is derivable in the calculus σ.

The truth of the Gödel sentence

Suppose that G is false. Then there is a b_0 such that the closed atomic formula $Diag_\sigma(a_0, b_0)$ is true and hence provable.

Therefore the conjuncts
$D(\sigma)(\mathbf{F}'a_0),\ D(\sigma)(b_0\mathbf{S}'a_0\mathbf{S}'[a_0]'\mathbf{S}'x'), D(\sigma)(b_0)$ are all true.

From the second conjunct follows that b_0 cannot be different from g because the result of substituting the code a_0 into the formula with the code a_0 is the formula with the code g.

Therefore, $D(\sigma)(g)$ is true. But it means that the formula with the code g – i.e., G itself – is derivable in the calculus σ.

To sum up: G is not a theorem, but if it were false, then it would be provable. Therefore, it is a true but unprovable sentence.

The truth of the Gödel sentence

Suppose that G is false. Then there is a b_0 such that the closed atomic formula $Diag_\sigma(a_0, b_0)$ is true and hence provable.

Therefore the conjuncts
$D(\sigma)(\mathbf{F}'a_0)$, $D(\sigma)(b_0\mathbf{S}'a_0\mathbf{S}'[a_0]'\mathbf{S}'x')$, $D(\sigma)(b_0)$ are all true.

From the second conjunct follows that b_0 cannot be different from g because the result of substituting the code a_0 into the formula with the code a_0 is the formula with the code g.

Therefore, $D(\sigma)(g)$ is true. But it means that the formula with the code g – i.e., G itself – is derivable in the calculus σ.

To sum up: G is not a theorem, but if it were false, then it would be provable. Therefore, it is a true but unprovable sentence.

$\neg G$ is not provable because it is false. Therefore, $\mathbf{C}C$ is not negation complete, q.e.d.

Generalization

Generalization

Theorem: Be T a first-order theory such that

i. all the theorems of **CC** are provable in T;

ii. the class of the theorems of T is definable by some canonical calculus K;

iii. no false formula of **CC** is provable in T.

Then T is incomplete. There is a sentence in the language of T which is true but not provable.

Generalization

Theorem: Be T a first-order theory such that

i. all the theorems of **CC** are provable in T;
ii. the class of the theorems of T is definable by some canonical calculus K;
iii. no false formula of **CC** is provable in T.

Then T is incomplete. There is a sentence in the language of T which is true but not provable.

Be $K' = k$. If K derives a string f, then $D(k)(f')$ is provable in T (because it is provable in **CC**). So we have an analogue of Lemma 2. Then we can introduce $Diag_k(a/x, b)$ exactly as we have introduced $Diag_\sigma$. We can prove Lemma 3. for theorems of T instead of **CC**, and produce a Gödel sentence for T.