The unprovability of the consistency of **CC**

András Máté

28.11.2025

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.
Σ: the canonical calculus generating the theorems of **CC**.

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.
Σ: the canonical calculus generating the theorems of **CC**.
σ: the code of the calculus Σ, i. e. $\Sigma' = \sigma$.

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.
Σ: the canonical calculus generating the theorems of **CC**.
σ: the code of the calculus Σ, i. e. $\Sigma' = \sigma$.
Lemma: The true closed atomic formulas of $\mathcal{L}^{10}$ are provable in **CC**.

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.
Σ: the canonical calculus generating the theorems of **CC**.
σ: the code of the calculus Σ, i. e. $\Sigma' = \sigma$.
Lemma: The true closed atomic formulas of $\mathcal{L}^{10}$ are provable in **CC**.
If A is a formula of $\mathcal{L}^{10}$ with at most one free variable and $A' = a$, then the diagonalization of A is the formula $B = A^{a/x}$ with the code $B' = b$.

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.
Σ: the canonical calculus generating the theorems of **CC**.
σ: the code of the calculus Σ, i. e. $\Sigma' = \sigma$.
Lemma: The true closed atomic formulas of $\mathcal{L}^{10}$ are provable in **CC**.
If A is a formula of $\mathcal{L}^{10}$ with at most one free variable and $A' = a$, then the diagonalization of A is the formula $B = A^{a/x}$ with the code $B' = b$.
$Diag_\sigma(a, b)$ is the abbreviaton of the formula

$$D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(b\mathbf{S}'a\mathbf{S}'a'\mathbf{S}'x') \wedge D(\sigma)(b).$$

Recapitulation (notations, lemmas, interpretations)

CC: The first-order theory of canonical calculi, formulated in the language $\mathcal{L}^{10}$.
The code of any object O is the string O'.
Σ: the canonical calculus generating the theorems of **CC**.
σ: the code of the calculus Σ, i. e. $\Sigma' = \sigma$.
Lemma: The true closed atomic formulas of $\mathcal{L}^{10}$ are provable in **CC**.
If A is a formula of $\mathcal{L}^{10}$ with at most one free variable and $A' = a$, then the diagonalization of A is the formula $B = A^{a/x}$ with the code $B' = b$.
$Diag_\sigma(a, b)$ is the abbreviaton of the formula

$$D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(b\mathbf{S}'a\mathbf{S}'a'\mathbf{S}'x') \wedge D(\sigma)(b).$$

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

Recapitulation continued

.

Recapitulation continued

Be A_0 the following formula with the code a_0:

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

.

Recapitulation continued

Be A_0 the following formula with the code a_0:

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

Its diagonalization is the sentence $G(= B_0)$ with the done g:

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

.

Be A_0 the following formula with the code a_0:

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

Its diagonalization is the sentence $G (= B_0)$ with the done g:

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

If G were provable in **CC**, then **CC** would be inconsistent.

.

Recapitulation continued

Be A_0 the following formula with the code a_0:

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

Its diagonalization is the sentence $G(= B_0)$ with the done g:

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

If G were provable in **CC**, then **CC** would be inconsistent.
* If G were false, then it would be provable in **CC**.

.

Recapitulation continued

Be A_0 the following formula with the code a_0:

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

Its diagonalization is the sentence $G(= B_0)$ with the done g:

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

If G were provable in **CC**, then **CC** would be inconsistent.
* If G were false, then it would be provable in **CC**.
Therefore, G is a true but unprovable sentence of $\mathcal{L}^{10}$ (first incompleteness theorem).

The consistency sentence

The consistency sentence

The following $\mathcal{L}^{10}$-sentence expresses the consistency of **CC**:
$Cons_\sigma = \exists\mathfrak{x}(D(\sigma)(\mathbf{F}'\mathfrak{x}) \wedge \neg D(\sigma)(\mathfrak{x}))$

The consistency sentence

The following $\mathcal{L}^{10}$-sentence expresses the consistency of **CC**:
$Cons_\sigma = \exists\mathfrak{x}(D(\sigma)(\mathbf{F}'\mathfrak{x}) \wedge \neg D(\sigma)(\mathfrak{x}))$
Let us abbreviate $D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(a)$ by $Th_\sigma(a)$. The starred proposition of the previous slide can be expressed in $\mathcal{L}^{10}$ by the sentence $\neg G \supset Th(g)$.

The consistency sentence

The following $\mathcal{L}^{10}$-sentence expresses the consistency of **CC**:
$Cons_\sigma = \exists\mathfrak{x}(D(\sigma)(\mathbf{F}'\mathfrak{x}) \wedge \neg D(\sigma)(\mathfrak{x}))$
Let us abbreviate $D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(a)$ by $Th_\sigma(a)$. The starred proposition of the previous slide can be expressed in $\mathcal{L}^{10}$ by the sentence $\neg G \supset Th(g)$.
The metalanguage argument for the starred proposition can be formalized as a deduction in **CC** (but we need SUD).
I. e., $\mathbf{CC} \vdash \neg G \supset Th(g)$ (Step 1.).

The consistency sentence

The following $\mathcal{L}^{10}$-sentence expresses the consistency of **CC**:
$Cons_\sigma = \exists\mathfrak{x}(D(\sigma)(\mathbf{F}'\mathfrak{x}) \wedge \neg D(\sigma)(\mathfrak{x}))$
Let us abbreviate $D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(a)$ by $Th_\sigma(a)$. The starred proposition of the previous slide can be expressed in $\mathcal{L}^{10}$ by the sentence $\neg G \supset Th(g)$.
The metalanguage argument for the starred proposition can be formalized as a deduction in **CC** (but we need SUD).
I. e., $\mathbf{CC} \vdash \neg G \supset Th(g)$ (Step 1.).
Be $C_0 = Diag_\sigma(a_0, g)$ with the code c_0. We know that $\mathbf{CC} \vdash G$ iff $\mathbf{CC} \vdash C_0$. This biconditional can be proven within **CC** again, i.e. $\mathbf{CC} \vdash B_0 \leftrightarrow C_0$.

The consistency sentence

The following $\mathcal{L}^{10}$-sentence expresses the consistency of **CC**:
$Cons_\sigma = \exists\mathfrak{x}(D(\sigma)(\mathbf{F}'\mathfrak{x}) \wedge \neg D(\sigma)(\mathfrak{x}))$
Let us abbreviate $D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(a)$ by $Th_\sigma(a)$. The starred proposition of the previous slide can be expressed in $\mathcal{L}^{10}$ by the sentence $\neg G \supset Th(g)$.
The metalanguage argument for the starred proposition can be formalized as a deduction in **CC** (but we need SUD).
I. e., $\mathbf{CC} \vdash \neg G \supset Th(g)$ (Step 1.).
Be $C_0 = Diag_\sigma(a_0, g)$ with the code c_0. We know that $\mathbf{CC} \vdash G$ iff $\mathbf{CC} \vdash C_0$. This biconditional can be proven within **CC** again, i.e. $\mathbf{CC} \vdash B_0 \leftrightarrow C_0$.
Using the definition of Th_σ and the previous lemmas, we get $\mathbf{CC} \vdash Th(b_0) \supset Th(c_0)$.

The consistency sentence

The following $\mathcal{L}^{10}$-sentence expresses the consistency of **CC**:
$Cons_\sigma = \exists\mathfrak{x}(D(\sigma)(\mathbf{F}'\mathfrak{x}) \wedge \neg D(\sigma)(\mathfrak{x}))$
Let us abbreviate $D(\sigma)(\mathbf{F}'a) \wedge D(\sigma)(a)$ by $Th_\sigma(a)$. The starred proposition of the previous slide can be expressed in $\mathcal{L}^{10}$ by the sentence $\neg G \supset Th(g)$.
The metalanguage argument for the starred proposition can be formalized as a deduction in **CC** (but we need SUD).
I. e., $\mathbf{CC} \vdash \neg G \supset Th(g)$ (Step 1.).
Be $C_0 = Diag_\sigma(a_0, g)$ with the code c_0. We know that $\mathbf{CC} \vdash G$ iff $\mathbf{CC} \vdash C_0$. This biconditional can be proven within **CC** again, i.e. $\mathbf{CC} \vdash B_0 \leftrightarrow C_0$.
Using the definition of Th_σ and the previous lemmas, we get $\mathbf{CC} \vdash Th(b_0) \supset Th(c_0)$.
Using the result of Step 1., we get
(Step 2.) $\mathbf{CC} \vdash \neg G \supset Th(c_0)$

The end of our proof

The end of our proof

We know that if $\mathbf{CC} \vdash C_0$, then $\mathbf{CC} \vdash G$, and if $\mathbf{CC} \vdash G$, then $\mathbf{CC} \vdash \neg C_0$.

The end of our proof

We know that if $\mathbf{CC} \vdash C_0$, then $\mathbf{CC} \vdash G$, and if $\mathbf{CC} \vdash G$, then $\mathbf{CC} \vdash \neg C_0$.

It follows that $\mathbf{CC} \vdash Th(c_0) \supset Th(\neg' c_0)$.

The end of our proof

We know that if $\mathbf{CC} \vdash C_0$, then $\mathbf{CC} \vdash G$, and if $\mathbf{CC} \vdash G$, then $\mathbf{CC} \vdash \neg C_0$.

It follows that $\mathbf{CC} \vdash Th(c_0) \supset Th(\neg' c_0)$.

Therefore, using Step 2. and propositional logic:
(Step 3.) $\mathbf{CC} \vdash \neg G \supset (Th(c_0) \wedge Th(\neg' c_0))$

The end of our proof

We know that if $\mathbf{CC} \vdash C_0$, then $\mathbf{CC} \vdash G$, and if $\mathbf{CC} \vdash G$, then $\mathbf{CC} \vdash \neg C_0$.

It follows that $\mathbf{CC} \vdash Th(c_0) \supset Th(\neg' c_0)$.

Therefore, using Step 2. and propositional logic:
(Step 3.) $\mathbf{CC} \vdash \neg G \supset (Th(c_0) \wedge Th(\neg' c_0))$
By first-order logic,
(Step 4.) $\mathbf{CC} \vdash (Th(c_0) \wedge Th(\neg' c_0)) \supset \neg Cons_\sigma$.

The end of our proof

We know that if $\mathbf{CC} \vdash C_0$, then $\mathbf{CC} \vdash G$, and if $\mathbf{CC} \vdash G$, then $\mathbf{CC} \vdash \neg C_0$.

It follows that $\mathbf{CC} \vdash Th(c_0) \supset Th(\neg' c_0)$.

Therefore, using Step 2. and propositional logic:
(Step 3.) $\mathbf{CC} \vdash \neg G \supset (Th(c_0) \wedge Th(\neg' c_0))$
By first-order logic,
(Step 4.) $\mathbf{CC} \vdash (Th(c_0) \wedge Th(\neg' c_0)) \supset \neg Cons_\sigma$.
Therefore, $\mathbf{CC} \vdash \neg G \supset \neg Cons_\sigma$.
and consequently,
$\mathbf{CC} \vdash Cons_\sigma \supset G$.

The end of our proof

We know that if $\mathbf{CC} \vdash C_0$, then $\mathbf{CC} \vdash G$, and if $\mathbf{CC} \vdash G$, then $\mathbf{CC} \vdash \neg C_0$.

It follows that $\mathbf{CC} \vdash Th(c_0) \supset Th(\neg' c_0)$.

Therefore, using Step 2. and propositional logic:
(Step 3.) $\mathbf{CC} \vdash \neg G \supset (Th(c_0) \wedge Th(\neg' c_0))$
By first-order logic,
(Step 4.) $\mathbf{CC} \vdash (Th(c_0) \wedge Th(\neg' c_0)) \supset \neg Cons_\sigma$.
Therefore, $\mathbf{CC} \vdash \neg G \supset \neg Cons_\sigma$.
and consequently,
$\mathbf{CC} \vdash Cons_\sigma \supset G$.
It means that if $Cons_\sigma$ were provable, then G, the Gödel sentence would be provable, too. But from the first incompleteness theorem we know that the Gödel sentence is not provable, and therefore $Cons_\sigma$ can't be provable, either. Q.e.d.