Hypercalculi (continuation)

András Máté

31.03.2023

Next goal: formalize derivability

Next goal: formalize derivability

Construct a (hyper)calculus $\mathbf{H}_2$ such that if the calculus $\mathbf{C}$ derives the string c then $\mathbf{H}_2$ derives a string that is the translation of $\mathbf{C} \mapsto c$.

Next goal: formalize derivability

Construct a (hyper)calculus $\mathbf{H}_2$ such that if the calculus $\mathbf{C}$ derives the string c then $\mathbf{H}_2$ derives a string that is the translation of $\mathbf{C} \mapsto c$.
We extend $\mathbf{H}_1$ (dropping the release rule 13*) to the calculus $\mathbf{H}_2$.

Next goal: formalize derivability

Construct a (hyper)calculus $\mathbf{H}_2$ such that if the calculus $\mathbf{C}$ derives the string c then $\mathbf{H}_2$ derives a string that is the translation of $\mathbf{C} \mapsto c$.

We extend $\mathbf{H}_1$ (dropping the release rule 13*) to the calculus $\mathbf{H}_2$.

Two new auxiliary letters: D for derivable and S for substitution. In more details:

Next goal: formalize derivability

Construct a (hyper)calculus $\mathbf{H}_2$ such that if the calculus $\mathbf{C}$ derives the string c then $\mathbf{H}_2$ derives a string that is the translation of $\mathbf{C} \mapsto c$.
We extend $\mathbf{H}_1$ (dropping the release rule 13*) to the calculus $\mathbf{H}_2$.
Two new auxiliary letters: D for derivable and S for substitution. In more details:

- xDy: the calculus x derives the string y

Next goal: formalize derivability

Construct a (hyper)calculus $\mathbf{H}_2$ such that if the calculus $\mathbf{C}$ derives the string c then $\mathbf{H}_2$ derives a string that is the translation of $\mathbf{C} \mapsto c$.

We extend $\mathbf{H}_1$ (dropping the release rule 13^*) to the calculus $\mathbf{H}_2$.

Two new auxiliary letters: D for derivable and S for substitution. In more details:

- xDy: the calculus x derives the string y
- $vSuSySx$: if we substitute the word y for the variable x, we get the string v from the string u. Remember that words are variable-free strings.

Next goal: formalize derivability

Construct a (hyper)calculus $\mathbf{H}_2$ such that if the calculus $\mathbf{C}$ derives the string c then $\mathbf{H}_2$ derives a string that is the translation of $\mathbf{C} \mapsto c$.

We extend $\mathbf{H}_1$ (dropping the release rule 13*) to the calculus $\mathbf{H}_2$.

Two new auxiliary letters: D for derivable and S for substitution. In more details:

- xDy: the calculus x derives the string y
- $vSuSySx$: if we substitute the word y for the variable x, we get the string v from the string u. Remember that words are variable-free strings.

In the above description of the intended meaning, I have dropped the phrase 'translation of'. But never forget that we speak here not about the letters, variables, etc. of our hypercalculus, but about the strings translating the letters etc. of the original calculus.

$\mathbf{H}_2$: the definition of substitution

$\mathbf{H}_2$: the definition of substitution

Substitution needs an inductive definition, too. Base: The substitution of the variable x by the word y makes y from x (rule 18.) and leaves any other character – letters (14.), the arrow (15.), other variables (16.-17) – unchanged. Inductive rule: If the substitution makes v from u and w from z, then from their concatenation uz it makes the concatenation of the results vw.

$\mathbf{H}_2$: the definition of substitution

Substitution needs an inductive definition, too. Base: The substitution of the variable x by the word y makes y from x (rule 18.) and leaves any other character – letters (14.), the arrow (15.), other variables (16.-17) – unchanged. Inductive rule: If the substitution makes v from u and w from z, then from their concatenation uz it makes the concatenation of the results vw.

14. $Lu \to uSuSySx$
15. $\gg S \gg SySx$
16. $Vx \to Iz \to x\beta zSx\beta zSySx$
17. $Vx \to Iz \to xSxSySx\beta z$
18. $Vx \to Wy \to ySxSySx$
19. $vSuSySx \to wSzSySx \to vwSuzSySx$

$\mathbf{H}_2$: the definition of derivability

$\mathbf{H}_2$: the definition of derivability

Base: every calculus derives its rules. (In details: an one-rule calculus derives the rule, and longer calculi derive their last, first and middle rules.) Inductive rules are substitution and detachment.

$\mathbf{H}_2$: the definition of derivability

Base: every calculus derives its rules. (In details: an one-rule calculus derives the rule, and longer calculi derive their last, first and middle rules.) Inductive rules are substitution and detachment.

20. $Rx \rightarrow xDx$
21. $Rx \rightarrow Ky \rightarrow y * xDx$
22. $Rx \rightarrow Ky \rightarrow x * yDx$
23. $Rx \rightarrow Ky \rightarrow Kz \rightarrow y * x * zDx$
24. $zDu \rightarrow vSuSySx \rightarrow zDv$
25. $xDy \rightarrow xDy \gg z \rightarrow xDz$

$\mathbf{H}_2$: the definition of derivability

Base: every calculus derives its rules. (In details: an one-rule calculus derives the rule, and longer calculi derive their last, first and middle rules.) Inductive rules are substitution and detachment.

20. $Rx \to xDx$
21. $Rx \to Ky \to y * xDx$
22. $Rx \to Ky \to x * yDx$
23. $Rx \to Ky \to Kz \to y * x * zDx$
24. $zDu \to vSuSySx \to zDv$
25. $xDy \to xDy \gg z \to xDz$

The calculus $\mathbf{H}_2$ consisting of the rules 1-25 derives Ka, Wb and aDb iff a is the translation of some calculus $\mathbf{C}$, b is the translation of a word c of the alphabet of $\mathbf{C}$ and $\mathbf{C}$ derives c. We can't give suitable release rules here.

$\mathbf{H}_2$ (over an alphabet $\mathcal{A}_{cc}$ plus 9 auxiliary letters) derives strings with the intended meanings "a is a calculus", "b is a string of the alphabet of a", "a derives b". (a and b are *translations, codes* of a calculus resp. word in $\mathcal{A}_{cc}$.)

The calculus $\mathbf{H}_3$

$\mathbf{H}_2$ (over an alphabet $\mathcal{A}_{cc}$ plus 9 auxiliary letters) derives strings with the intended meanings "a is a calculus", "b is a string of the alphabet of a", "a derives b". (a and b are *translations, codes* of a calculus resp. word in $\mathcal{A}_{cc}$.)

The calculus $\mathbf{H}_3$ is an extension of $\mathbf{H}_2$. It renders numerals to every $\mathcal{A}_{cc}$-string. (This is in effect a Gödel numbering.)
Numerals: strings consisting of α-s only.

$\mathbf{H}_2$ (over an alphabet $\mathcal{A}_{cc}$ plus 9 auxiliary letters) derives strings with the intended meanings "a is a calculus", "b is a string of the alphabet of a", "a derives b". (a and b are *translations, codes* of a calculus resp. word in $\mathcal{A}_{cc}$.)

The calculus $\mathbf{H}_3$ is an extension of $\mathbf{H}_2$. It renders numerals to every $\mathcal{A}_{cc}$-string. (This is in effect a Gödel numbering.)
Numerals: strings consisting of α-s only.

First step: introduce a lexicographic ordering of $\mathcal{A}_{cc}$-strings.
New auxiliary letter: F for the relation 'follows'.
I. e., xFy should mean that the string y follows x in the lexicographic ordering.
Base: α follows the empty word.
Inductive rules define the follower of a string according to its last letter.

Lexicographic ordering

26. $F\alpha$
27. $x\alpha Fx\beta$
28. $x\beta Fx\xi$
29. $x\xi Fx \gg$
30. $x \gg Fx*$
31. $xFy \to x * Fy\alpha$

Lexicographic ordering

26. $F\alpha$
27. $x\alpha Fx\beta$
28. $x\beta Fx\xi$
29. $x\xi Fx \gg$
30. $x \gg Fx*$
31. $xFy \rightarrow x * Fy\alpha$

From the language radix axioms it follows that:
Every $\mathcal{A}_{cc}$-string has one and only one follower;
Except of the empty string, each string is the follower of one and only one string.
The empty string is not a follower of anything.
I. e., strings with the empty string as 0 and this follower-relation as the successor-function fulfil axioms of primitive Peano arithmetics without mathematical induction.

Gödel numbering of $\mathcal{A}_{cc}$-strings

Gödel numbering of $\mathcal{A}_{cc}$-strings

Now we can add the (Gödel-)numbering to our calculus on the trivial way.
G is a a new auxiliary letter, intended meaning of xGy: y is the ordinal number of x in the lexicographic ordering.

Gödel numbering of $\mathcal{A}_{cc}$-strings

Now we can add the (Gödel-)numbering to our calculus on the trivial way.
G is a a new auxiliary letter, intended meaning of xGy: y is the ordinal number of x in the lexicographic ordering.
Basis: the ordinal number of the empty string is the empty string itself.
Inductive rule: to get the number of the follower of a string x we need to add an α to the number of x.

Now we can add the (Gödel-)numbering to our calculus on the trivial way.
G is a a new auxiliary letter, intended meaning of xGy: y is the ordinal number of x in the lexicographic ordering.
Basis: the ordinal number of the empty string is the empty string itself.
Inductive rule: to get the number of the follower of a string x we need to add an α to the number of x.

32. G
33. $xFy \to xGz \to yGz\alpha$

Now we can add the (Gödel-)numbering to our calculus on the trivial way.
G is a a new auxiliary letter, intended meaning of xGy: y is the ordinal number of x in the lexicographic ordering.
Basis: the ordinal number of the empty string is the empty string itself.
Inductive rule: to get the number of the follower of a string x we need to add an α to the number of x.

32. G

33. $xFy \to xGz \to yGz\alpha$

Our hypercalculus $\mathbf{H}_3$ now consists of the rules 1-33. and it suffices to prove at least one important incompleteness result.

Autonomous numerals

Autonomous numerals

Be **C** an arbitrary calculus.

An $\mathcal{A}_{cc}$-word a is the translation of **C** into our language; $\mathbf{H}_3$ derives Ka.

There is a numeral c s.t. $\mathbf{H}_3$ derives aGc, i. e. the Gödel number of **C** is c.

Autonomous numerals

Be **C** an arbitrary calculus.
An $\mathcal{A}_{cc}$-word a is the translation of **C** into our language; $\mathbf{H}_3$ derives Ka.
There is a numeral c s.t. $\mathbf{H}_3$ derives aGc, i. e. the Gödel number of **C** is c.

Does **C** derive a string whose translation is c?
Be **C** a calculus with this property (deriving its own Gödel number).
Then $\mathbf{H}_3$ derives aDc, too.
Let us call such c-s autonomous numbers.

Autonomous numerals

Be **C** an arbitrary calculus.
An $\mathcal{A}_{cc}$-word a is the translation of **C** into our language; $\mathbf{H}_3$ derives Ka.
There is a numeral c s.t. $\mathbf{H}_3$ derives aGc, i. e. the Gödel number of **C** is c.

Does **C** derive a string whose translation is c?
Be **C** a calculus with this property (deriving its own Gödel number).
Then $\mathbf{H}_3$ derives aDc, too.
Let us call such c-s autonomous numbers.
Let us extend $\mathbf{H}_3$ to define autonomous numbers.
New auxiliary letter: A with the intended meaning "autonomous". New rule:

Be **C** an arbitrary calculus.
An $\mathcal{A}_{cc}$-word a is the translation of **C** into our language; $\mathbf{H}_3$ derives Ka.
There is a numeral c s.t. $\mathbf{H}_3$ derives aGc, i. e. the Gödel number of **C** is c.

Does **C** derive a string whose translation is c?
Be **C** a calculus with this property (deriving its own Gödel number).
Then $\mathbf{H}_3$ derives aDc, too.
Let us call such c-s autonomous numbers.
Let us extend $\mathbf{H}_3$ to define autonomous numbers.
New auxiliary letter: A with the intended meaning "autonomous". New rule:

$$34. \quad xDy \to xGy \to Ay$$

Our Gödel-like theorem

Our Gödel-like theorem

The numbers are the strings of the one-letter alphabet $\mathcal{A}_0 = \{\alpha\}$, so their class is $\mathcal{A}_0^\circ$ and it can be defined inductively. The class of autonomous numerals, in class theoretic notation:

$$\boldsymbol{Aut} = \{x : x \in \mathcal{A}_0^\circ \wedge \mathbf{H}_3 \mapsto Ax\}$$

By adding a release rule deleting A to $\mathbf{H}_3$, we gain a definition of $\boldsymbol{Aut}$ by a canonical calculus.

Our Gödel-like theorem

The numbers are the strings of the one-letter alphabet $\mathcal{A}_0 = \{\alpha\}$, so their class is $\mathcal{A}_0^\circ$ and it can be defined inductively. The class of autonomous numerals, in class theoretic notation:

$$\boldsymbol{Aut} = \{x : x \in \mathcal{A}_0^\circ \wedge \mathbf{H}_3 \mapsto Ax\}$$

By adding a release rule deleting A to $\mathbf{H}_3$, we gain a definition of $\boldsymbol{Aut}$ by a canonical calculus.

We prove that the string class $\mathcal{A}_0^\circ - \boldsymbol{Aut}$ (the class of non-autonomous numerals) cannot be defined inductively.

The numbers are the strings of the one-letter alphabet $\mathcal{A}_0 = \{\alpha\}$, so their class is $\mathcal{A}_0^\circ$ and it can be defined inductively. The class of autonomous numerals, in class theoretic notation:

$$\boldsymbol{Aut} = \{x : x \in \mathcal{A}_0^\circ \wedge \mathbf{H}_3 \mapsto Ax\}$$

By adding a release rule deleting A to $\mathbf{H}_3$, we gain a definition of $\boldsymbol{Aut}$ by a canonical calculus.

We prove that the string class $\mathcal{A}_0^\circ - \boldsymbol{Aut}$ (the class of non-autonomous numerals) cannot be defined inductively.

Theorem: There is no canonical calculus $\mathbf{C}$ over some $B \supseteq \mathcal{A}_{cc}$ s.t. for any string x,

$$\mathbf{C} \mapsto x \Leftrightarrow x \in \mathcal{A}_0^\circ - \boldsymbol{Aut}$$

Proof of the theorem

Proof of the theorem

Let us assume toward contradiction that we have a calculus $\mathbf{C}$ with the Gödel number g s.t for every non-autonomous numeral c, $\mathbf{C} \mapsto c$, and there is no autonomous numeral d for that $\mathbf{C} \mapsto d$.

Proof of the theorem

Let us assume toward contradiction that we have a calculus **C** with the Gödel number g s.t for every non-autonomous numeral c, $\mathbf{C} \mapsto c$, and there is no autonomous numeral d for that $\mathbf{C} \mapsto d$.

Suppose that $\mathbf{C} \mapsto g$. In this case, **C** is an autonomous calculus, g is an autonomous number, therefore **C** does not derive g. Contradiction.

Proof of the theorem

Let us assume toward contradiction that we have a calculus $\mathbf{C}$ with the Gödel number g s.t for every non-autonomous numeral c, $\mathbf{C} \mapsto c$, and there is no autonomous numeral d for that $\mathbf{C} \mapsto d$.

Suppose that $\mathbf{C} \mapsto g$. In this case, $\mathbf{C}$ is an autonomous calculus, g is an autonomous number, therefore $\mathbf{C}$ does not derive g. Contradiction.

Suppose that $\mathbf{C}$ does not derive g. In this case, $\mathbf{C}$ is not an autonomous calculus, g is a non-autonomous number, therefore $\mathbf{C} \mapsto g$. Contradiction again, q.e.d.

Proof of the theorem

Let us assume toward contradiction that we have a calculus $\mathbf{C}$ with the Gödel number g s.t for every non-autonomous numeral c, $\mathbf{C} \mapsto c$, and there is no autonomous numeral d for that $\mathbf{C} \mapsto d$.

Suppose that $\mathbf{C} \mapsto g$. In this case, $\mathbf{C}$ is an autonomous calculus, g is an autonomous number, therefore $\mathbf{C}$ does not derive g. Contradiction.

Suppose that $\mathbf{C}$ does not derive g. In this case, $\mathbf{C}$ is not an autonomous calculus, g is a non-autonomous number, therefore $\mathbf{C} \mapsto g$. Contradiction again, q.e.d.

This theorem is Gödel-like because it shows that no inductive definition can be given for the notion "non-autonomous calculus" just like Gödel's first incompleteness theorem shows that no inductive definition can be given for the notion "arithmetical truth". And this proof uses an analogue of the Liar Paradox, too.