First-order languages, first-order calculus (**QC**) The language $\mathcal{L}^{1*}$

András Máté

28.01.2023

Logical calculi

Logical calculus:

Logical calculus:

- An L family of languages with a distinguished category $Form_L$;

Logical calculi

Logical calculus:

- An L family of languages with a distinguished category $Form_L$;
- Inductive definition of the *syntactic* consequence (deducibility) relation $\Gamma \vdash_L A$, where $\Gamma \subseteq Form_L$ (premises) and $A \in Form_L$ (conclusion).

Logical calculus:

- An L family of languages with a distinguished category $Form_L$;
- Inductive definition of the *syntactic* consequence (deducibility) relation $\Gamma \vdash_L A$, where $\Gamma \subseteq Form_L$ (premises) and $A \in Form_L$ (conclusion).

Base of the inductive definition: a class of formulas deducible from the empty class of premises (*basic formulas* or *logical axioms*).

Logical calculus:

- An L family of languages with a distinguished category $Form_L$;
- Inductive definition of the *syntactic* consequence (deducibility) relation $\Gamma \vdash_L A$, where $\Gamma \subseteq Form_L$ (premises) and $A \in Form_L$ (conclusion).

Base of the inductive definition: a class of formulas deducible from the empty class of premises (*basic formulas* or *logical axioms*).

Inductive rules (rules of deduction, proof rules) prescribe how you can arrive from some given relations $\Gamma \vdash A_1, \Gamma \vdash A_2, \ldots$ to some new relation $\Gamma \vdash A$.

Logical calculi (continuation)

Logical calculi (continuation)

Different ways to define the deducibility relation: many axioms and only one or two rules of deduction (Frege-Hilbert style of calculus) versus no axioms at all, only rules (Gentzen-style or natural deduction systems).

Logical calculi (continuation)

Different ways to define the deducibility relation: many axioms and only one or two rules of deduction (Frege-Hilbert style of calculus) versus no axioms at all, only rules (Gentzen-style or natural deduction systems).

Equivalence of different calculi (for the same family of languages): on the natural way (the extension of the relation $\vdash$ is the same).

Logical calculi (continuation)

Different ways to define the deducibility relation: many axioms and only one or two rules of deduction (Frege-Hilbert style of calculus) versus no axioms at all, only rules (Gentzen-style or natural deduction systems).

Equivalence of different calculi (for the same family of languages): on the natural way (the extension of the relation $\vdash$ is the same).

A natural demand for the class of logical axioms and the rules of deduction: they should be decidable.

First-order languages

First-order languages

All the symbols are strings of some given alphabet $\mathcal{A}$.

First-order languages

All the symbols are strings of some given alphabet $\mathcal{A}$.

The class of arities $A = \{\varnothing,\ o,\ oo, \ldots\}$ was defined inductively earlier.

First-order languages

All the symbols are strings of some given alphabet $\mathcal{A}$.

The class of arities $A = \{\varnothing, \ o, \ oo, \ldots\}$ was defined inductively earlier.

A first-order language $\mathcal{L}^1$ is a quintuple

$$< Log, Var, Con, Term, Form >$$

where $Log = \{(, \ ), \ \neg, \ \supset, \ \forall, \ =\}$ is the class of logical constants, Var is the infinite class of variables defined inductively, and $Con = N \cup P = \bigcup_{a \in A} P_a \cup \bigcup_{a \in A} N_a$ is the class of non-logical constants containing all the classes P_a of a-ary predicates and N_a of a-ary name functors.

All the symbols are strings of some given alphabet $\mathcal{A}$.

The class of arities $A = \{\varnothing,\ o,\ oo, \ldots\}$ was defined inductively earlier.

A first-order language $\mathcal{L}^1$ is a quintuple

$$< Log, Var, Con, Term, Form >$$

where $Log = \{(,\ ),\ \neg,\ \supset,\ \forall,\ =\}$ is the class of logical constants, Var is the infinite class of variables defined inductively, and $Con = N \cup P = \bigcup_{a \in A} P_a \cup \bigcup_{a \in A} N_a$ is the class of non-logical constants containing all the classes P_a of a-ary predicates and N_a of a-ary name functors.

It is assumed that for $a_i \neq a_j \in A$, $N_{a_i} \cap N_{a_j} = P_{a_i} \cap P_{a_j} = \emptyset$ and $N \cap P = \emptyset$.

Terms and a-tuples of terms

The class of a-tuples of terms $a \in A$ is $T(a)$.

Terms and a-tuples of terms

The class of a-tuples of terms $a \in A$ is $T(a)$.

The simultaneous inductive definition of the classes $Term$ and $T(a)$:

Terms and a-tuples of terms

The class of a-tuples of terms $a \in A$ is $T(a)$.

The simultaneous inductive definition of the classes $Term$ and $T(a)$:

1. $Var \subseteq Term$
2. $T(\varnothing) = \{\varnothing\}$
3. $(s \in T(a) \ \&\ t \in Term) \Rightarrow \ulcorner s(t) \urcorner \in T(ao)$
4. $(\varphi \in N_a \ \&\ s \in T(a)) \Rightarrow \ulcorner \varphi s \urcorner \in Term$

Formulas

Formulas

1. $\pi \in P_a \;\&\; s \in T(a) \Rightarrow \ulcorner \pi s \urcorner \in Form$
2. $s, t \in Term \Rightarrow \ulcorner s = t \urcorner \in Form$
3. $A \in Form \Rightarrow \ulcorner \neg A \urcorner \in Form$
4. $A, B \in Form \Rightarrow \ulcorner A \supset B \urcorner \in Form$
5. $A \in Form \;\&\; x \in Var \Rightarrow \ulcorner \forall x A \urcorner \in Form$

Formulas

1. $\pi \in P_a \ \& \ s \in T(a) \Rightarrow \ulcorner \pi s \urcorner \in Form$
2. $s, t \in Term \Rightarrow \ulcorner s = t \urcorner \in Form$
3. $A \in Form \Rightarrow \ulcorner \neg A \urcorner \in Form$
4. $A, B \in Form \Rightarrow \ulcorner A \supset B \urcorner \in Form$
5. $A \in Form \ \& \ x \in Var \Rightarrow \ulcorner \forall x A \urcorner \in Form$

<u>Atomic formulas</u> are the formulas generated by the rules 1. and 2.

Formulas

1. $\pi \in P_a \ \& \ s \in T(a) \Rightarrow \ulcorner \pi s \urcorner \in Form$
2. $s, t \in Term \Rightarrow \ulcorner s = t \urcorner \in Form$
3. $A \in Form \Rightarrow \ulcorner \neg A \urcorner \in Form$
4. $A, B \in Form \Rightarrow \ulcorner A \supset B \urcorner \in Form$
5. $A \in Form \ \& \ x \in Var \Rightarrow \ulcorner \forall x A \urcorner \in Form$

Atomic formulas are the formulas generated by the rules 1. and 2.

Other logical constants ($\vee$, $\wedge$, $\equiv$, $\exists$) are introduced by abbreviation conventions.

Formulas

1. $\pi \in P_a \ \& \ s \in T(a) \Rightarrow \ulcorner \pi s \urcorner \in Form$
2. $s, t \in Term \Rightarrow \ulcorner s = t \urcorner \in Form$
3. $A \in Form \Rightarrow \ulcorner \neg A \urcorner \in Form$
4. $A, B \in Form \Rightarrow \ulcorner A \supset B \urcorner \in Form$
5. $A \in Form \ \& \ x \in Var \Rightarrow \ulcorner \forall x A \urcorner \in Form$

<u>Atomic formulas</u> are the formulas generated by the rules 1. and 2.

Other logical constants ($\vee$, $\wedge$, $\equiv$, $\exists$) are introduced by abbreviation conventions.

Be $A, B \in Form$. B is a <u>subformula</u> of A iff A is of the form uBv $(u, v \in \mathcal{A}^\circ)$.

Formulas

1. $\pi \in P_a$ & $s \in T(a) \Rightarrow \ulcorner \pi s \urcorner \in Form$
2. $s, t \in Term \Rightarrow \ulcorner s = t \urcorner \in Form$
3. $A \in Form \Rightarrow \ulcorner \neg A \urcorner \in Form$
4. $A, B \in Form \Rightarrow \ulcorner A \supset B \urcorner \in Form$
5. $A \in Form$ & $x \in Var \Rightarrow \ulcorner \forall x A \urcorner \in Form$

<u>Atomic formulas</u> are the formulas generated by the rules 1. and 2.

Other logical constants ($\vee$, $\wedge$, $\equiv$, $\exists$) are introduced by abbreviation conventions.

Be $A, B \in Form$. B is a <u>subformula</u> of A iff A is of the form uBv $(u, v \in \mathcal{A}^\circ)$.

If $x \in Var$ and $A \in Form$, an occurrence of x in A is a <u>bound occurrence of x in A</u> iff it lies in a subformula of A of the form $\forall x B$. Other occurrences are called <u>free occurrences</u>.

Some further auxiliary notions

Some further auxiliary notions

A term is open iff at least one variable is a substring of it; otherways it is closed.

Some further auxiliary notions

A term is <u>open</u> iff at least one variable is a substring of it; otherways it is <u>closed</u>.

A formula is <u>open</u> if it contains at least one free occurrence of a variable; otherwise it is <u>closed</u>. Closed formulas are called <u>sentences</u>.

Some further auxiliary notions

A term is <u>open</u> iff at least one variable is a substring of it; otherways it is <u>closed</u>.

A formula is <u>open</u> if it contains at least one free occurrence of a variable; otherwise it is <u>closed</u>. Closed formulas are called <u>sentences</u>.

A formula A is <u>free from</u> the variable x iff x has no free occurrences in A. $\Gamma \subseteq Form$ is <u>free from</u> x if each member of it is.

Some further auxiliary notions

A term is open iff at least one variable is a substring of it; otherways it is closed.

A formula is open if it contains at least one free occurrence of a variable; otherwise it is closed. Closed formulas are called sentences.

A formula A is free from the variable x iff x has no free occurrences in A. $\Gamma \subseteq Form$ is free from x if each member of it is.

Be $A \in Form$, $x, y \in Var$. y is substitutable for x in A iff for every subformula of A of the form $\forall yB$, B is free from x.

$t \in Term$ is substitutable for x in A iff every variable occurring in t is substitutable. If t is substitutable for x in A, then $A^{t/x}$ denotes (in the metalanguage) the formula obtained from A substituting t for every free occurrence of x in A.

The quantification calculus (**QC**) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

The quantification calculus (**QC**) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

(B1) $(A \supset (B \supset A))$

(B2) $((A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)))$

(B3) $((\neg B \supset \neg A) \supset (A \supset B))$

The quantification calculus (**QC**) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

(B1) $(A \supset (B \supset A))$

(B2) $((A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C)))$

(B3) $((\neg B \supset \neg A) \supset (A \supset B))$

(B4) $(\forall x A \supset A^{t/x})$

(B5) $(\forall x(A \supset B) \supset (\forall x A \supset \forall x B))$

(B6) $(A \supset \forall x A)$ provided that A is free from x

The quantification calculus (**QC**) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

(B1) $(A \supset (B \supset A))$

(B2) $((A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C)))$

(B3) $((\neg B \supset \neg A) \supset (A \supset B))$

(B4) $(\forall x A \supset A^{t/x})$

(B5) $(\forall x(A \supset B) \supset (\forall x A \supset \forall x B))$

(B6) $(A \supset \forall x A)$ provided that A is free from x

(B7) $(\mathfrak{x} = \mathfrak{x})$

(B8) $((x = y) \supset (A^{x/z} \supset A^{y/z}))$

The quantification calculus (**QC**) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

(B1) $(A \supset (B \supset A))$

(B2) $((A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C)))$

(B3) $((\neg B \supset \neg A) \supset (A \supset B))$

(B4) $(\forall x A \supset A^{t/x})$

(B5) $(\forall x(A \supset B) \supset (\forall x A \supset \forall x B))$

(B6) $(A \supset \forall x A)$ provided that A is free from x

(B7) $(\mathfrak{x} = \mathfrak{x})$

(B8) $((x = y) \supset (A^{x/z} \supset A^{y/z}))$

The class BF of logical axioms is defined inductively:

The quantification calculus (**QC**) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

(B1) $(A \supset (B \supset A))$

(B2) $((A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C)))$

(B3) $((\neg B \supset \neg A) \supset (A \supset B))$

(B4) $(\forall x A \supset A^{t/x})$

(B5) $(\forall x (A \supset B) \supset (\forall x A \supset \forall x B))$

(B6) $(A \supset \forall x A)$ provided that A is free from x

(B7) $(\mathfrak{x} = \mathfrak{x})$

(B8) $((x = y) \supset (A^{x/z} \supset A^{y/z}))$

The class BF of logical axioms is defined inductively:

i If we substitute for A, B, C formulas, for x, y, z variables and for t terms of $\mathcal{L}^1$ in the above schemes, we get members of BF.

The quantification calculus (QC) 1: the axioms

Given a first-order language $\mathcal{L}^1$, the logical axioms (basic formulas) are defined by the help of the following schemes:

(B1) $(A \supset (B \supset A))$

(B2) $((A \supset (B \supset C) \supset ((A \supset B) \supset (A \supset C)))$

(B3) $((\neg B \supset \neg A) \supset (A \supset B))$

(B4) $(\forall x A \supset A^{t/x})$

(B5) $(\forall x(A \supset B) \supset (\forall x A \supset \forall x B))$

(B6) $(A \supset \forall x A)$ provided that A is free from x

(B7) $(\mathfrak{x} = \mathfrak{x})$

(B8) $((x = y) \supset (A^{x/z} \supset A^{y/z}))$

The class BF of logical axioms is defined inductively:

i If we substitute for A, B, C formulas, for x, y, z variables and for t terms of $\mathcal{L}^1$ in the above schemes, we get members of BF.

ii If $A \in BF$ and $x \in Var$, then $\ulcorner \forall x A \urcorner \in BF$.

QC 2: deducibility and some metatheorems

QC 2: deducibility and some metatheorems

Base for the inductive definition of $\Gamma \vdash A$: if $A \in \Gamma \cup BF$, then $\Gamma \vdash A$. Inductive rule is detachment: if $\Gamma \vdash A$ and $\Gamma \vdash A \supset B$, then $\Gamma \vdash B$.

QC 2: deducibility and some metatheorems

Base for the inductive definition of $\Gamma \vdash A$: if $A \in \Gamma \cup BF$, then $\Gamma \vdash A$. Inductive rule is detachment: if $\Gamma \vdash A$ and $\Gamma \vdash A \supset B$, then $\Gamma \vdash B$.

- Deduction Theorem: If $\Gamma \cup \{A\} \vdash C$, then $\Gamma \vdash A \supset C$.

QC 2: deducibility and some metatheorems

Base for the inductive definition of $\Gamma \vdash A$: if $A \in \Gamma \cup BF$, then $\Gamma \vdash A$. Inductive rule is detachment: if $\Gamma \vdash A$ and $\Gamma \vdash A \supset B$, then $\Gamma \vdash B$.

- Deduction Theorem: If $\Gamma \cup \{A\} \vdash C$, then $\Gamma \vdash A \supset C$.
- Cut: If $\Gamma \vdash A$ and $\Gamma' \cup \{A\} \vdash B$ then $\Gamma \cup \Gamma' \vdash B$.

QC 2: deducibility and some metatheorems

Base for the inductive definition of $\Gamma \vdash A$: if $A \in \Gamma \cup BF$, then $\Gamma \vdash A$. Inductive rule is detachment: if $\Gamma \vdash A$ and $\Gamma \vdash A \supset B$, then $\Gamma \vdash B$.

- Deduction Theorem: If $\Gamma \cup \{A\} \vdash C$, then $\Gamma \vdash A \supset C$.
- Cut: If $\Gamma \vdash A$ and $\Gamma' \cup \{A\} \vdash B$ then $\Gamma \cup \Gamma' \vdash B$.
- Universal generalization: If $\Gamma \vdash A$ and Γ is free from x, then $\Gamma \vdash \forall x A$.

QC 2: deducibility and some metatheorems

Base for the inductive definition of $\Gamma \vdash A$: if $A \in \Gamma \cup BF$, then $\Gamma \vdash A$. Inductive rule is detachment: if $\Gamma \vdash A$ and $\Gamma \vdash A \supset B$, then $\Gamma \vdash B$.

- Deduction Theorem: If $\Gamma \cup \{A\} \vdash C$, then $\Gamma \vdash A \supset C$.
- Cut: If $\Gamma \vdash A$ and $\Gamma' \cup \{A\} \vdash B$ then $\Gamma \cup \Gamma' \vdash B$.
- Universal generalization: If $\Gamma \vdash A$ and Γ is free from x, then $\Gamma \vdash \forall x A$.
- Universal generalization 2.: If $t \in T(\varnothing)$ s.t. it occurs neither in A nor in the members of Γ and $\Gamma \vdash A^{t/x}$ then $\Gamma \vdash \forall x A$.

QC 2: deducibility and some metatheorems

Base for the inductive definition of $\Gamma \vdash A$: if $A \in \Gamma \cup BF$, then $\Gamma \vdash A$. Inductive rule is detachment: if $\Gamma \vdash A$ and $\Gamma \vdash A \supset B$, then $\Gamma \vdash B$.

- Deduction Theorem: If $\Gamma \cup \{A\} \vdash C$, then $\Gamma \vdash A \supset C$.
- Cut: If $\Gamma \vdash A$ and $\Gamma' \cup \{A\} \vdash B$ then $\Gamma \cup \Gamma' \vdash B$.
- Universal generalization: If $\Gamma \vdash A$ and Γ is free from x, then $\Gamma \vdash \forall x A$.
- Universal generalization 2.: If $t \in T(\varnothing)$ s.t. it occurs neither in A nor in the members of Γ and $\Gamma \vdash A^{t/x}$ then $\Gamma \vdash \forall x A$.

A definition: If $A \in Form$ and the variables having free occurrences in A are $x_1, x_2, \ldots x_n$, then the universal closure of A is the formula $\forall x_1 \forall x_2 \ldots \forall x_n A$.

Given any logical calculus Σ in a language $\mathcal{L}$ and a class Γ of formulas of $\mathcal{L}$, the class of the consequences of Γ is the class

$$Cns(\Gamma) = \{A \in Form : \Gamma \vdash_{\Sigma} A\}$$

Consequences, consistency, first-order theories

Given any logical calculus Σ in a language $\mathcal{L}$ and a class Γ of formulas of $\mathcal{L}$, the class of the consequences of Γ is the class

$$Cns(\Gamma) = \{A \in Form : \Gamma \vdash_{\Sigma} A\}$$

Γ is <u>inconsistent</u> if $Cns(\Gamma) = Form$, <u>consistent</u> in the other case.

Consequences, consistency, first-order theories

Given any logical calculus Σ in a language $\mathcal{L}$ and a class Γ of formulas of $\mathcal{L}$, the class of the consequences of Γ is the class

$$Cns(\Gamma) = \{A \in Form : \Gamma \vdash_\Sigma A\}$$

Γ is <u>inconsistent</u> if $Cns(\Gamma) = Form$, <u>consistent</u> in the other case.

In first-order logic, Γ is consistent iff there is no $A \in Form$ s. t. both $\Gamma \vdash A$ and $\Gamma \vdash \neg A$.

Given any logical calculus Σ in a language $\mathcal{L}$ and a class Γ of formulas of $\mathcal{L}$, the class of the consequences of Γ is the class

$$Cns(\Gamma) = \{A \in Form : \Gamma \vdash_{\Sigma} A\}$$

Γ is inconsistent if $Cns(\Gamma) = Form$, consistent in the other case.

In first-order logic, Γ is consistent iff there is no $A \in Form$ s. t. both $\Gamma \vdash A$ and $\Gamma \vdash \neg A$.

The pair $T =< \mathcal{L}^1, \Gamma >$ is a first-order theory if $\mathcal{L}^1$ is a first-order language and Γ is a class of its *closed* formulas (called *axioms* of T).

Given any logical calculus Σ in a language $\mathcal{L}$ and a class Γ of formulas of $\mathcal{L}$, the class of the consequences of Γ is the class

$$Cns(\Gamma) = \{A \in Form : \Gamma \vdash_{\Sigma} A\}$$

Γ is inconsistent if $Cns(\Gamma) = Form$, consistent in the other case.

In first-order logic, Γ is consistent iff there is no $A \in Form$ s. t. both $\Gamma \vdash A$ and $\Gamma \vdash \neg A$.

The pair $T =< \mathcal{L}^1, \Gamma >$ is a first-order theory if $\mathcal{L}^1$ is a first-order language and Γ is a class of its *closed* formulas (called *axioms* of T).

The theorems of T are the members of $Cns(\Gamma)$. T is said consistent resp. inconsistent if Γ is consistent resp. inconsistent.

The theory $\mathbf{CC}^*$ and its language $\mathcal{L}^{1*}$

The theory $\mathbf{CC}^*$ and its language $\mathcal{L}^{1*}$

$\mathbf{CC}^*$ is the rewriting of the (hyper)calculus $\mathbf{H}_3$ in the form of a first-order theory.
$\mathbf{H}_3$ derives strings like Ka, Wb, aDb, aGb, Aa with the intended meanings 'a is a calculus', ... 'a is an autonomous number'. We want $\mathbf{CC}^*$ to prove formulas like $K(a), \ldots A(a)$ just in the same case.

The theory $\mathbf{CC}^*$ and its language $\mathcal{L}^{1*}$

$\mathbf{CC}^*$ is the rewriting of the (hyper)calculus $\mathbf{H}_3$ in the form of a first-order theory.
$\mathbf{H}_3$ derives strings like Ka, Wb, aDb, aGb, Aa with the intended meanings 'a is a calculus', …'a is an autonomous number'. We want $\mathbf{CC}^*$ to prove formulas like $K(a), \ldots A(a)$ just in the same case.

The language of $\mathbf{CC}^*$ is the first-order language $\mathcal{L}^{1*}$.
Non-logical components :

The theory $\mathbf{CC}^*$ and its language $\mathcal{L}^{1*}$

$\mathbf{CC}^*$ is the rewriting of the (hyper)calculus $\mathbf{H}_3$ in the form of a first-order theory.
$\mathbf{H}_3$ derives strings like Ka, Wb, aDb, aGb, Aa with the intended meanings 'a is a calculus', … 'a is an autonomous number'. We want $\mathbf{CC}^*$ to prove formulas like $K(a), \ldots A(a)$ just in the same case.

The language of $\mathbf{CC}^*$ is the first-order language $\mathcal{L}^{1*}$.
Non-logical components :

- $N_\varnothing = \{\vartheta,\ \alpha,\ \beta,\ \xi,\ \gg,\ *\}$
  ϑ denotes the empty string, the other name constants denote (autonymously) the letters of $\mathcal{A}_{cc}$.

The theory $\mathbf{CC}^*$ and its language $\mathcal{L}^{1*}$

$\mathbf{CC}^*$ is the rewriting of the (hyper)calculus $\mathbf{H}_3$ in the form of a first-order theory.
$\mathbf{H}_3$ derives strings like Ka, Wb, aDb, aGb, Aa with the intended meanings 'a is a calculus', … 'a is an autonomous number'. We want $\mathbf{CC}^*$ to prove formulas like $K(a), \ldots A(a)$ just in the same case.

The language of $\mathbf{CC}^*$ is the first-order language $\mathcal{L}^{1*}$.
Non-logical components :

- $N_\varnothing = \{\vartheta,\ \alpha,\ \beta,\ \xi,\ \gg,\ *\}$
  ϑ denotes the empty string, the other name constants denote (autonymously) the letters of $\mathcal{A}_{cc}$.
- $N_{oo} = \{\varnothing\}$
  The empty string denotes concatenation (and we omit the parentheses around its arguments), i.e., we write the concatenation of the strings x and y as xy.

The language $\mathcal{L}^{1*}$ (continuation)

The auxiliary letters of the hypercalculi $\mathbf{H}_1$ – $\mathbf{H}_3$ become predicates and we keepthe intended meanings:

The auxiliary letters of the hypercalculi $\mathbf{H}_1$ – $\mathbf{H}_3$ become predicates and we keepthe intended meanings:

- $P_o = \{I,\ L,\ V,\ W,\ T,\ R,\ K,\ A\}$

The language $\mathcal{L}^{1*}$ (continuation)

The auxiliary letters of the hypercalculi $\mathbf{H}_1$ – $\mathbf{H}_3$ become predicates and we keepthe intended meanings:

- $P_o = \{I,\ L,\ V,\ W,\ T,\ R,\ K,\ A\}$
- $P_{oo} = \{D,\ F,\ G\}$

The language $\mathcal{L}^{1*}$ (continuation)

The auxiliary letters of the hypercalculi $\mathbf{H}_1$ – $\mathbf{H}_3$ become predicates and we keepthe intended meanings:

- $P_o = \{I,\ L,\ V,\ W,\ T,\ R,\ K,\ A\}$
- $P_{oo} = \{D,\ F,\ G\}$
- $P_{oooo} = \{S\}$
  $S(v)(u)(y)(x)$: if we substitute the word y for the variable x, we get the string v from the string u.)

The language $\mathcal{L}^{1*}$ (continuation)

The auxiliary letters of the hypercalculi $\mathbf{H}_1$ – $\mathbf{H}_3$ become predicates and we keepthe intended meanings:

- $P_o = \{I,\ L,\ V,\ W,\ T,\ R,\ K,\ A\}$
- $P_{oo} = \{D,\ F,\ G\}$
- $P_{oooo} = \{S\}$
  $S(v)(u)(y)(x)$: if we substitute the word y for the variable x, we get the string v from the string u.)

Logical constants, variables (let us write them as $\mathfrak{x}, \mathfrak{x}_1, \ldots$), the syntax of terms and formulas are like in any other first-order language. The intended universe (the domain of the variables) is the class of $\mathcal{A}_{cc}$-strings.