Completeness Proof of Functional Logic, 
A Formalism with Variable-Binding Nonlogical Symbols

Schönbrunner Josef 
Institut für Logistik der Universität Wien 
Universitätsstraße 10/11, A-1090 Wien (Austria) 
e-mail a8121dab@helios.edvz.univie.ac.at

Abstract

We know extensions of first order logic by quantifiers of the kind “there are uncountable many ...”, “most ...” with new axioms and appropriate semantics. Related are operations such as “set of x, such that ...”, Hilbert’s ε-operator, Church’s λ-notation, minimization and similar ones, which also bind a variable within some expression, the meaning of which is however partly defined by a translation into the language of first order logic. In this paper a generalization is presented that comprises arbitrary variable-binding symbols as non-logical operations. The axiomatic extension is determined by new equality-axioms; models allocate functionals to variable-binding symbols. The completeness of this system of the so called functional logic of 1st order will be proved.

Mathematics Subject Classification: 03C80, 03B99.

1 Introduction

Functional logic is a generalization of first order predicate logic with different kinds of objects by adding the following new features:

1. The division of expressions into the categories of sentences and individuals (i.e. formulas and terms) is weakened as with a differentiation of sorts of terms formulas shall also be treated as a sort. Thus the classification of the symbolic entities into logical connectives, predicate symbols, function symbols loses its significance, as the membership to one of it depends only on its signature (i.e. number and sorts of the argument-places and sort of the resulting expression). The sentential sort (formulas) retains its special role and will be referred to as \( \pi \). Thus the signature of a binary connective is \( \pi(\pi, \pi) \), that of a \( n \)-ary predicate symbol \( \pi(\alpha_1, ..., \alpha_n) \), that of a \( n \)-ary function symbol \( \gamma(\alpha_1, ..., \alpha_n) \) and that of a constant symbol \( \gamma \), if each \( \alpha_i \) and \( \gamma \) are sorts. Not to be found in predicate logic are symbolic entities whose argument-places are mixed, partly of sort \( \pi \) and partly of another object-sort. These do not fit
into any of the categories of logical connectives, predicate symbols or function symbols
mentioned above. An example is the expression ‘$E, a, b$’ denoting an object “$a$ if $E,$
b otherwise”, which is built up by a symbolic entity ‘?’ of the signature ‘$\alpha(\pi, \alpha, \alpha)$’.

2. In a formalized theory of predicate logic expressions such as \{x \mid E\}, \{x \in M \mid E\}, \{x\in (E), ax(E), \mu x (E), \mu x \in (E), \int_a^b e \cdot dx \} are characterized only by an external rule
of translation into the language of the theory. In functional logic, however, such
expressions can be generated internally by symbolic entities that bind variables. This
is the essential extension of this formalism.

In standardized symbolisation a symbolic entity ‘op’ of the resulting sort $\gamma$ with
k argument places of signature$(\alpha_1, \beta_1), \beta_1 = (\beta_{1, j})_{1 \leq i \leq k}$ is linked with the
generation rule by which

\[
\begin{align*}
\text{‘op’} & \quad \text{if } k = 0 \quad \text{(constant or variable) – or –} \\
\text{‘op(.., [\bar{q}_i] : a_i, ..)’} & \quad \text{if } k > 0
\end{align*}
\]
is an expression of sort $\gamma$, if each $a_i$ is an expression of sort $\alpha_i$ and each $\bar{q}_i = (q_{i, j})_{1 \leq i \leq r_i}$
is a sequence of variables of sorts-sequence $\beta_1 = (\beta_{i, j})_{1 \leq i \leq r_i}$. The case $r_i = 0$ means
that the optional part, which is written as [...] is to be dropped. This case applies
to logical connectives, predicate symbols and function symbols in all argument-places
$i$, only quantifiers have a signature ‘$\pi_\alpha(\alpha : \pi)$’ with $r_i = 1$. Putting the template
[...] around something is used to consider both cases $r_i = 0$ as well as $r_i > 0$. If
$r_i > 0$ the brackets can be erased and ‘$[(\bar{q}_i) : a_i]$’ stands for ‘$(\bar{q}_i) : a_i$’, which is an
abbreviation of ‘$\langle q_{i,1}, \ldots, q_{i,r_i} \rangle : a_i$’. ‘$\bar{q}_i$’ are the binding variables to ‘$a_i$’.

**Examples:**
1. The extension of a formal Peano-system by axioms like

\[
\text{Apl}_k((x_1, \ldots, x_k) : c, a_1, \ldots, a_k) = e^{x_1 \ldots x_k}_{a_1 \ldots a_k},
\]

allows the representation of each primitive recursive function by a single term. (In
the above schemes of axioms $c, a, b, a_1$ range over arbitrary terms and $n$ is a number
variable.) If all free variables of $a$ and $b$ which are not members of \{‘$y$, ‘$z$’\} are
in \{‘$u_1$, ‘$u_2$, ‘$u_m$’\}, then the term ‘$\text{Apl}_k((y, z) : b, n - 1, \text{PR}(a, (y, z) : b, n - 1))$’
can be associated with a $m + 1$–ary function of arguments $u_1, \ldots, u_m, n$, defined by primitive recursion from
base-function \((u_1, \ldots, u_m) \mapsto a\) and iteration-function \((u_1, \ldots, u_m, y, z) \mapsto b\).

2. Quantifiers to variables of different sorts must be distinguished, the signature
of ‘$\forall^{\alpha}$‘ is ‘$\pi((\alpha) : \pi)$’. In standardized manner, a formula ‘$\forall^{\alpha}(x^\alpha : E)$’
would be

\[
\forall^{\alpha}(x^\alpha : E).
\]

3. A standardized version of expressing “the least $x$ less than $b$ such that $E$ if one
exists, or $b$ if none exists” (usually symbolized by ‘$\mu x (E)$’ is ‘$\mu_x(b,(x) : E)$’, the
signature of ‘$\mu_x$’ being ‘$\forall^{\nu}((\nu) : \pi)$’ if $\nu$ is the sort of natural Numbers.

A standardized symbolic language and an ideal language for application with the
same expression ability are different. The first should be simple in order to avoid
unnecessary expense in metatheoretic treatment. With regard to application this
simplicity can be disadvantageous. For instance in predicate calculus one symbol cannot be used with different signatures depending on the sorts of arguments it appears with. Such multiple use of a symbol became popular in programming languages, when looking at overloaded versions of procedure-names. Application of formal logics could profit from such a technique, too. For instance, consider sorts $\alpha, \beta$ and a class of models such that the range of $\beta$ is a substructure of the range of $\alpha$, if the signature of a symbol w.r.t. a certain argument-place is of sort $\alpha$, then any term of sort $\beta$ also fits into that place. “overloading of symbols” may yield simpler axiom-schemes. Yet it requires change from the notion of symbol to that of symbolic entity (= symbol + signature). As a basis of meta-linguistic reference we shall take the standardized form. Results derived on this basis can easily be transferred into more flexible symbolism.

As to the logical axioms, the usual schemes of predicate calculus may be adapted, but binding of variables (significant to the axioms) is performed by symbols other than quantifiers, too. One part of the equality axioms become

$$(\forall z_1) \ldots (\forall z_{r_i}) \left( a_{i_1} \frac{z_{i_1}}{z_1} = b_{i_1} \frac{y_{i_1}}{z_1} \right) \rightarrow \text{op}(\ldots, ([x_1]_i) : a_{i_1}, \ldots) \overset{\gamma}{=} \text{op}(\ldots, ([y_1]_i) : b_{i_1}, \ldots)$$

where $x_{i_1} \equiv x_1, \ldots, x_{i_t},$ similarly $y_{i_1}, z_{i_1},$ and where $a_{i_1} \frac{z_{i_1}}{z_1}$ designates the expression obtained from $a_{i_1}$ by replacing each free occurrence of $z_{1 j}$ by $y_{1 j}$ (for $1 \leq j \leq r_i$) and $\text{op}(\ldots, ([x_1]_i) : b_{i_1}, \ldots)$ differs from $\text{op}(\ldots, ([x_1]_i) : a_{i_1}, \ldots)$ only by the $i$-th argument. Note:

1. if $r_i = 0$, then the above sequence of universal quantifiers becomes empty; 2. If $\gamma = \pi$ is the sentential sort, then $\pi$ is to be identified with $\leftrightarrow (=\text{logical equivalence})$.

The main problem is introducing appropriate semantics to which the calculus is complete. Let “$\text{op}$” be a symbol with $k > 0$ argument places, at least one of them provides binding variables i.e. $r_i > 0$ for some $i : 1 \ldots k$. At first consideration we suppose an interpretation-structure to assign to ‘$\text{op}$’ the functional

$$\mathfrak{M}(\text{‘op’}): \prod_{i:1..k} \mathcal{V}_i \rightarrow \mathcal{M}_y, \quad \mathcal{V}_i = \begin{cases} \mathfrak{M}_\alpha_i & \text{if } r(i) = 0 \\ \mathfrak{M}(\prod_{j:1..r_i} \mathfrak{M}_{\beta_{i,j}}, \mathfrak{M}_{\alpha_i}) & \text{if } r(i) > 0 \end{cases}$$

($\mathfrak{M}_y$ is the range of $\gamma$ and $\text{Map}(X, Y) = \{f | f: X \rightarrow Y \} = \gamma^X$). But this turns out to fix too much, as assignment only to a part of the functions of $\text{Map}(\prod_{j:1..r_i} \mathfrak{M}_{\beta_{i,j}}, \mathfrak{M}_{\alpha_i})$ will be relevant for evaluation of expressions. Nothing beyond that partial assignment you may expect to come out from the syntactic information of a consistent theory. To overcome this problem a certain restriction of the argument ranges $\mathcal{V}_i$ will help. The notion of a structure $\mathfrak{M}$ must therefore be extended by a new component which assigns a selected set $\mathfrak{M}_{\gamma}^\alpha \subseteq \text{Map}(\prod_{i:1..m} \mathfrak{M}_{\sigma_i}, \mathcal{M}_y)$ to each sequence of sorts $\gamma, \bar{\sigma}$. The selected sets are characterized by some closure qualities similar to those that apply to the set of (primitive-) recursive functions, for instance constant functions and projections are to be included. In a trivial way, however, we find an extension $\overline{\mathfrak{M}}$ of $\mathfrak{M}$ so that $\overline{\mathfrak{M}}_y = \text{Map}(\prod_{i:1..m} \mathfrak{M}_{\sigma_i}, \mathcal{M}_y)$ and the interpretations of expressions by $\mathfrak{M}$ and by $\overline{\mathfrak{M}}$
coincide as well as the semantic consequences $M \models$ and $\mathfrak{M} \models$. To construct a model of a consistent formal theory the method of extension to a complete Henkin Theory as in Henkin’s Proof of the Completeness Theorem [HEN49, SHO67] is still applicable.

2 Survey

As basic structure of a 1st order functional logic language we define the Fnl$_1$ signature. Then a standardized language is specified that determines the notion of an expression ‘e’ of sort γ. This is defined inductively by a characteristic syntactic relation of ‘e’ to a symbol ‘op’ (the root of ‘e’), argument expressions ‘$a_i$’ and possibly variables ‘$v_{ij}$’ binding ‘$a_i$’. As this relation shall frequently appear as a background premise within definitions and proofs constantly using the same arguments ‘e’, ‘op’, ‘$a_i$’ and ‘$v_{ij}$’, we introduce the abbreviation Generation-Premise. The definition of a Fnl$_1$-structure is based on the notion of a Fnl$_1$ signature according to features discussed in the introduction. We shall only consider logic with fixed equality base on normal structure semantics. To derive semantics for the language from the notion of structure based on a signature, that is to establish an interpretation of the expressions (of various sorts), the usual definition as a map from variables-assignments to the domain of the sort the expression belongs to is not suitable. Instead of it now an expression ‘e’ will be evaluated according to a Fnl$_1$ structure $\mathfrak{M}$ by assigning a mapping on the set of the so called perspectives of ‘e’ consisting of all finite sequences of variables, such that all free variables of ‘e’ appear within that sequence. Let γ be the resulting sort of ‘e’. The evaluation of ‘e’ based on $\mathfrak{M}$ maps the empty sequence ⟨⟩ into a member of the range $\mathfrak{M}_\gamma$ of γ, provided that ⟨⟩ is a perspective of ‘e’ (i.e. if ‘e’ has no free variable) and it maps a non-empty perspective ⟨$u_1$,...,$u_m$⟩ of ‘e’ into a function of $\mathfrak{M}_{\sigma_1} \times \ldots \times \mathfrak{M}_{\sigma_m} \rightarrow \mathfrak{M}_\gamma$, if $\sigma_j$ is the sort of the variable ‘$u_j$’ ($j=1,...,m$). The definition will be inductive based on the background-assumption of Generation-Premise.

As to the axiomatization, the logical axioms differ in shape from predicate logic only a little with regard to equality logic. But we must also take into account an extension of some notions which are basic to formulate axioms of logic, namely the notions of free and bound variables, substitution and substitutability. The axioms system together with the rules Modus Ponens and Generalization establishes the calculus of Functional Logic. The extension of this calculus by individual nonlogical axioms is called a functional logic theory. A Fnl$_1$ structure-model of a consistent functional logic theory can be constructed as in predicate logic from an extension of that theory which inherits consistency, admits examples and is complete. (admitting examples is related to the existence of terms t for each formula $\varphi$ with at most one free variable x, so that $\exists x \varphi \rightarrow \varphi[x \leftarrow t]$ is a theorem; we associate this theorem to designate t as an example, if $\exists x \varphi$ is true. ) In Henkin’s proof this is achieved in two steps: The 1st extension produces a theory that admits examples by addition of constant symbols and special axioms (s. [HEN49, SHO67, BAR77]). Consistency continues as this extension is conservative (each theorem of the extended theory, if restricted to the original language, is also provable within the original theory). The 2nd extension by Lindenbaum’s theorem enlarges the set of nonlogical axioms without changing the language. Both extensions can easily be adapted to functional logic. The definition of a “term structure”, which shall prove to be a model of the constructed extension to a closed Henkin Theory and hence also a model of the original theory,
also relies on a so-called norm function that assigns a representative to each closed expression within a congruence class. This class will be defined by the congruence relation, that applies to ‘a’ and ‘b’ iff ‘a = b’ is a theorem of the extended theory. As we suppose completeness of the extended theory, there are exactly two congruence classes of expressions of sort π; hence we choose the constants λ and γ (representing true or false respectively) as values of the norm function of formulae. Upon the set of norms (i.e. values of the norm function), which is a subset of closed expressions to each sort as base-range, we then define our so-called term-structure. The model quality of this structure will be obtained as an immediate consequence of a theorem (by specialization). The claim of this theorem is that the evaluation of an expression ‘e’ by the term-structure X is a function which assigns to each perspective a mapping from a cartesian product of certain ranges ...Xα1... to Xγ, which can be described exclusively by application of multiple substitution (variables by terms) from ‘e’ and application of the norm function. The validity of a formula (= expression of sort π) within a model means that its interpretation maps one (and implicitly all) non-empty expressions of sort α, that might be bound by ri variables of sorts βi,j. Significant for the notion to be defined is also a distinguished sort π (the type of formulae) and distinguished elements of SOPS : ‘γ’, ‘λ’, ‘¬’ ’→’, ‘∧’, ‘∨’, ‘τ’, ‘□’ for each α of VSRTS) and ‘□’ (for α ∈ SRTS) with fixed values relative to signS. In formalized manner now we stipulate all characterizations of this definition as follows:

3 Signature and Language

3.1 Definition FnS Signat S : The notion of “S is a signature of 1st-order functional logic” is determined by the following key-components: SRTS: sorts; SOPS: symbolic operations; VSRTS: sorts for which variables and quantification are provided. VARs: variables; signS: signature map, signS ‘op’ = ⟨γ, α, β⟩ for ‘op’ ∈ SOPS ∪ VARs characterizes ‘op’ as a symbolic operation to generate expressions of sort γ from n argument-expressions of sort αi, that might be bound by ri variables of sorts βi,j. Significant for the notion to be defined is also a distinguished sort π (the type of formulae) and distinguished elements of SOPS : ‘γ’, ‘λ’, ‘¬’, ‘→’, ‘∧’, ‘∨’, ‘τ’, ‘□’ (for each α of VSRTS) and ‘□’ (for α ∈ SRTS) with fixed values relative to signS. In formalized manner now we stipulate all characterizations of this definition as follows:

FnS Signat S ↔ Conjunction of the following attributes:

S = (SRTS, SOPS, VSRTS, VARs, signS) ~> SRTS ⊆ SRTS ~> VARs ⊆ SOPS = ∅

signS: (SOPs ∪ VARs) —> SRTS × \bigcup_{m} (SRTS × VSRTS)

(\forall' \in \VARs) signS ' \in VSRTS × \{\}\ 2 SOPS ∪ VARs can be well-ordered

(\forall \alpha \in VSRTS) ( \{ 'v' \in VARs | signS 'v' = (\alpha, (\)) \} is enumerable )

\pi \in SRTS ‘γ’, ‘λ’, ‘¬’, ‘→’, ‘∧’, ‘∨’, ‘τ’, ‘□’ ∈ SOPS

(\forall \alpha \in VSRTS) ‘□’ (for \alpha ∈ SRTS) ‘□’ ∈ SOPS
signs for the distinguished members of SOPS is specified by a circumscription signs, (s. auxiliary notations below):

| op   | γ, λ | ¬ | →, ∧, ∨, ↔ | ∀γ, γ′, γ′′ | |  
|      |      |   |           |          | for α ∈ VSRTS | for α ∈ VSRTS |

Auxiliary Notations (dependent components) to a given Fnl Signat S:

(∀ α ∈ VSRTS) COPα = { ‘c’ ∈ SOPS | signs ‘c’ = (α, ()()) }  = COPα constants
(∀ α ∈ VSRTS) VARα = { ‘v’ ∈ VAR | signs ‘v’ = (α, ()()) }  = VARα variables
(∀ σ = ⟨α⟩i:1..l ∈ VSRT* ) VARσ = ∪ i:1..l VARσi = { u | u = ⟨ui⟩i:1..l ∧ (∀ i:1..l) ‘ui’ ∈ VARσi }

For signs we use a circumscription that is more convenient for application:

signs : SOPS ∪ VARS → (SRTS ∪ { ‘,’ , ‘‘, ‘’ })* (∀ ‘op’ ∈ SOPS ∪ VARS )
signs ‘op’ = ⟨γ, α, β⟩ iff

I. Notational Clauses (1) Subscript S will be omitted (SRT for VSRTS, ..., VARα for VARγ) if only one Fnl Signat is considered. (2) α, β, γ, ..., αi, βij, ... denote members of SRT. (3) u, u1, ..., vij, ... denote members of VAR. (4) Symbols with an arrow-accent refer to a finite sequence and writing such symbols one after another denotes the concatenation of the sequences (if p = ⟨p1, ..., pl⟩ and q = ⟨q1, ..., qk⟩ then p||q = ⟨p1, ..., pl, q1, ..., qk⟩). (5) If such a symbol e.g. u appears inside a quoted string, as for instance ‘op[(u : α)]’, it denotes the string ‘u1, ..., ui’, which is the concatenation of each ‘ui’ with ‘,’ interspersed. (6) We shall always assume αi = ⟨αi⟩i:1..m, β = ⟨β⟩i:1..m, βi = ⟨βi⟩i:1..r.

3.2 Definition (Lγ)γ∈SRTS : Let S be a Fnl Signat. The standardized language of S is introduced as a mapping on SRTS by stipulating for each γ ∈ VSRTS the set Lγ of expressions of sort γ inductively as follows (by above clause (1) Lγ = Lγ)

e ∈ Lγ ↔ (∃ ‘op’, m, γ, α, β, v)(Generation-Premise)
3.3 Definition \( \mathcal{L}_S = \bigcup_{\gamma \in \text{SRT}} \mathcal{L}_{\gamma} \quad (\vec{a} \in \text{SRT}^\ell) \quad \mathcal{L}_\sigma = \prod_{i:1..\ell} \mathcal{L}_{\alpha_i} \)

4 Semantics of Functional Logic

4.1 Definition Fnl\(^1\) Structures \( \mathcal{M} \) is a Fnl\(^1\)-type normal structure of signature \( S \), iff the following conditions apply to it:

i) Fnl\(^1\) Signat \( S = (\text{SRT}, \text{SOP}, \text{VSRT}, \text{VAR}, \text{sign}) \)

ii) \( \mathcal{M} \) is a mapping defined on \( \text{SRT} \cup (\text{SRT} \times \text{VSRT}^* \cup \text{SOP}) \). This mapping assigns elements of \( \text{SRT} \) to corresponding ranges, members of \( \text{SOP} \) to symbol-interpretations (i.e., corresponding elements of or functions on such ranges or functionals in case of symbols that bind variables). To ordered pairs of \( \text{SRT} \times \text{VSRT}^* \) it assigns those components which determine the classes of functions admitted as arguments of the letter functionals.

1) \( (\forall \alpha \in \text{SRT} \quad \mathcal{M}(\alpha) \equiv_{\text{def}} \emptyset \) and we automatically extend \( \mathcal{M} \) to \( \text{SRT}^* \):

\( (\forall \vec{\sigma} = (\sigma_i)_{i:1..\ell} \in \text{SRT}^*) \quad \mathcal{M}(\vec{\sigma}) \equiv_{\text{def}} \prod_{i:1..\ell} \mathcal{M}_{\sigma_i} \)

2) \( (\forall \text{‘op’} \in \text{SOP}, \text{sign} \text{‘op’} = (\gamma, \vec{\alpha}, \vec{\beta})) \) (using \( \mathcal{M}^{\vec{\alpha}}_{\alpha_i} \equiv_{\text{def}} \mathcal{M}(\alpha_i, \vec{\beta}_1) \))

if \( m = 0 \) then \( \mathcal{M} \text{‘op’} \) in \( \mathcal{M}_\gamma \), otherwise \( \mathcal{M} \text{‘op’} : \prod_{i:1..m} \mathcal{M}^{\vec{\beta}}_{\alpha_i} \rightarrow \mathcal{M}_\gamma \)

3) For arbitrary \( \gamma \in \text{SRT} \), \( \vec{a} = (\sigma_i)_{i:1..\ell} \in \text{VSRT}^\ell \)
3.1) \( \ell = 0 \to M(\gamma, \delta) = M(\gamma, \langle \rangle) = M_\gamma \)
\( \ell > 0 \to M(\gamma, \delta) \subseteq \text{Map}(M_\sigma, M_\gamma) \equiv M^{\varphi}_\gamma \)
alias notation: \( M^{\varphi}_\gamma \equiv M(\gamma, \delta) \)

**completion qualities of \( M^{\varphi}_\gamma \):**

3.2) \((\forall w \in M_\gamma) \ cst^\delta_w \equiv [\overrightarrow{\nu} \to w] \in M^{\varphi}_\gamma \) (constant funcs.)
\((\forall j : 1 \ldots \ell) \ p_{ij}^\delta \equiv [\overrightarrow{\nu} \to \nu_{\alpha_i}] \in M^{\varphi}_\gamma \) (projections)

3.3) \((\forall g \in \text{VSrt}^\beta) \ (\forall \vec{x} \in M_\sigma) \ \forall g \)

\[ \text{if } g \in M^{\varphi}_\gamma \text{ then } g_\vec{r} \equiv [\overrightarrow{\nu} \to g(\vec{r})] \in M^{\varphi}_\gamma \] (partial fixing)

3.4) \((\forall \text{op}' \in \text{Sop}, \ \text{sign} \text{‘op}' = (\gamma, \vec{a}, \vec{B}) ) \) \((\forall g_1, ..., g_m ) \)

the premises \( m > 0, \ell > 0 \) and \( (\forall i : 1 \ldots m) \ g_i \in M^{\varphi}_{\alpha_i} \)
and introducing the auxiliary notation:

\[ h_i = \begin{cases} \ g_i & \text{if } r_i = 0 \\ \text{if } r_i > 0 \end{cases} \]

imply \( (\overrightarrow{\nu} \to \nu_\gamma) M \text{‘op}' (h_1(\vec{y}), ..., h_m(\vec{y})) \in M^{\varphi}_\gamma \) (composition)

**What \( M \) assigns to the fixed components of \( \text{S} \):**

4) \( M_\varpi = \{ M \text{‘} \lambda \text{‘}, M \text{‘} \gamma \text{‘} \} = \mathbb{B} = \{ 0_\mathbb{B}, 1_\mathbb{B} \} \)
\( \langle M_\varpi, M \text{‘} \lambda \text{‘}, M \text{‘} \gamma \text{‘}, M_\varpi \text{‘} \lambda \text{‘}, M_\varpi \text{‘} \gamma \text{‘} \rangle = \mathbb{B} = \{ 0_\mathbb{B}, 1_\mathbb{B}, -_\mathbb{B}, \cap_\mathbb{B}, \cup_\mathbb{B} \} \)
forms a Boolean algebra with two elements, \( M \text{‘} \to \text{‘} \rangle \) and \( M \text{‘} \leftrightarrow \text{‘} \rangle \) are represented by the (dependent) truth-operations \( -_\mathbb{B} \) and \( \uparrow_\mathbb{B} \). \( M \text{‘} \lambda \text{‘} \) and \( M \text{‘} \gamma \text{‘} \) are defined for \( \alpha \in \text{VSrt} \) as follows:

\( M \text{‘} \lambda \text{‘} \), \( M \text{‘} \gamma \text{‘} \): \( M_\alpha \frac{\delta}{\delta} \to M_\varpi \) for each \( \theta \in M_\alpha \frac{\delta}{\delta} \) we stipulate
\( \text{if } (\forall \vec{x} \in M_\alpha \frac{\delta}{\delta}) \ \theta (\vec{x}) = 1_\mathbb{B} \text{ then } M \text{‘} \lambda \text{‘} (\theta) = 1_\mathbb{B} \) otherwise \( M \text{‘} \lambda \text{‘} (\theta) = 0_\mathbb{B} \);
\( \text{if } (\exists \vec{x} \in M_\alpha \frac{\delta}{\delta}) \ \theta (\vec{x}) = 1_\mathbb{B} \text{ then } M \text{‘} \gamma \text{‘} (\theta) = 1_\mathbb{B} \) otherwise \( M \text{‘} \gamma \text{‘} (\theta) = 0_\mathbb{B} \).

5) \((\forall \alpha \in \text{SRT} ) \ M \text{‘} \leq \frac{\delta}{\delta} : M_\alpha \frac{\delta}{\delta} \to M_\varpi, \ (x, y) \to 1_\mathbb{B} \text{ if } x = y \text{ or } 0_\mathbb{B} \text{ otherwise} \)

To extend a structure \( M \) into an interpretation of the language, i.e. to find an evaluation of expressions \( L_\gamma \) another approach than that based on variables-assignments as in predicate logic is required. The following definitions are prerequisites for the new approach.

**4.2 Definition** persp: \( L_\mathcal{S} \to \mathcal{P} (\text{VAR}^\ast) \) (Let \( \text{Fn}^\mathcal{L} \text{Signat} \mathcal{S}, \ L_\mathcal{S} = \bigcup_{\gamma \in \text{SRT}} L_\gamma \))

For ‘e’ \( \in L_\mathcal{S} \), persp’e’ denotes the set of all \( \langle u_i \rangle_{1 \leq i \leq \ell} \in \text{VAR}^\ast \) such that all free variables of ‘e’ are in \( \{ u_i \mid 1 \leq i \leq \ell \} \).
We shall need a more technical approach in defining this conception using syntactic induction. If Generation-Premise (Def. 3.2 on page 4) is assumed, then persp ‘e’ depends on persp ‘a_i’ as follows:

| cases | persp ‘e’ |
|-------|-----------|
| m = 0 | ‘op’ ∈ VAR | \{(‘u_i’)_i:1..ℓ ∈ VAR | (∃ j: 1..ℓ) ‘op’ = ‘u_j’\} |
| ‘op’ ∉ VAR | VAR* |
| m > 0 | \{ū ∈ VAR | (∀ i: 1..m) āv_i ∈ persp ‘a_i’\} |

4.3 Definition (Let Fnl Signat S, āv ∈ VAR*)

\((γ ∈ SRT) \; \mathcal{L}_γ[ū] = \{‘e’ ∈ \mathcal{L}_γ | ū ∈ persp ‘e’\} \; (\bar{σ} ∈ SRT^ℓ) \; \mathcal{L}_σ[ū] = \prod_{i:1..ℓ} \mathcal{L}_σ_i[ū] \) 

\(\mathcal{L}_γ[ū]\) is the set of expressions of \(\mathcal{L}_γ\) whose free variables are among \{‘u_i’ | i: 1..ℓ\} if \(ū = (‘u_i’)_i:1..ℓ\). \(\mathcal{L}_γ[ū]\) therefore is the set of closed \(γ\)–expressions.

4.4 Observation (for Fnl Signat S): \(\mathcal{L}_γ = \bigcup_{ū ∈ VAR*} \mathcal{L}_γ[ū]\)

4.5 Observation (for Fnl Signat S  \(ū = (‘u_i’)_i:1..ℓ ∈ VAR_γ\)  \(σ ∈ VSR_γ\)):

\(‘e’ ∈ \mathcal{L}_γ[ū] \iff (∃ ‘op’, m, γ, α, β, ā, \bar{v})(persp.GP)\)

where persp.GP (=perspective G.P.) can be obtained from Generation-Premise (p. 4) by modification of two conditions: if we change ‘op’ ∈ SOP ∪ VAR into ‘op’ ∈ SOP ∪ \(∪ \{‘u_i’ | i: 1..ℓ\}\) and \(ā ∈ \mathcal{L}_γ\) into \(ā ∈ \prod_{i:1..m} \mathcal{L}_σ_i[ūv_i]\) (each ‘a_i’ ∈ \(\mathcal{L}_α_i[ūv_i]\)).

4.6 Definition Interpretation of the language into a structure

Let Fnl Structures \(M\), γ ∈ SRT, ‘e’ ∈ \(\mathcal{L}_γ\) and Generation-Premise be assumed. The evaluation \(‘e’_M(ū)\) of ‘e’ is defined to be a function on persp ‘e’. Let \(ū = (‘u_i’)_i:1..ℓ ∈ persp ‘e’, (∀ i: 1..ℓ) sign ‘u_i’ = ‘a_i’, \bar{σ} = (σ_i)_i:1..m\). Then ‘e’_M(ū) is defined inductively:

| cases | ‘e’_M(ū) |
|-------|-----------|
| ℓ = 0 | m = 0 | = \(M^{‘op’}\) |
| (ū = ο) | m > 0 | = \(M^{‘op’}(‘a_i’_M(ūv_i)_i:1..m)\) |
| ℓ > 0 | m = 0 | ‘op’ ∈ VAR | = \(pj_\bar{v}\) \(\mathcal{M}_\bar{v} \rightarrow \mathcal{M}_{σ_k}\) \(\{x_i\}_i \mapsto x_k\) \(\max_i‘u_i’ = ‘op’\) |
| | m > 0 | ‘op’ ∈ SOP | = \(cst_{M^{‘op’}}\mathcal{M}_\bar{v} \rightarrow \mathcal{M}_\bar{x} \rightarrow \mathcal{M}_{‘op’}\) |
| | | | with M_\bar{h}_i defined below by *) |

*) (case ℓ > 0, m > 0) \(∀ i: 1..m\) \(h_i: \mathcal{M}_\bar{v} \rightarrow \mathcal{M}_{a_i}\), if \(r_ℓ = 0\: h_i: \bar{x} \mapsto ‘a_i’_M(ūv_i)_\bar{x}\)

if \(r_ℓ > 0\) then \(h_i: \bar{x} \mapsto (‘a_i’_M(ūv_i))_\bar{x} = \left[\left[\mathcal{M}_\bar{y} \rightarrow \mathcal{M}_{a_i}\right]_\bar{y} \mapsto ‘a_i’_M(ūv_i)\right]_\bar{x}\).
4.7 Proposition \( 'e' \in \mathcal{L}_\gamma[\bar{u}] \land \bar{\sigma} \in \text{VSRT}^* \land \bar{u} \in \text{VAR}_\bar{\sigma} \rightarrow \ 'e'_{\mathcal{M}}(\bar{u}) \in \mathcal{M}_\gamma^\bar{\sigma} \)

**Proof (+ Remark).** This proposition is already required for the argument expressions \( 'a_1' \) of the preceding definition (4.6) to assert that \( (\mathcal{h}_1(\bar{x}))_{1:1..m} \) belongs to the domain of \( \mathcal{M}_{\mathcal{Op}} \) (this assertion also requires (3.3) of 4.1 def.). Conditions (4.1(3)) imply that the above proposition propagates from the \( 'a_1' \) to \( 'e' \); so syntactic induction ensures its validity and any circularity of 4.6 def. that might result from presupposing it (for \( 'a_1' \)) is avoided as well. \( \square \)

4.8 Observation If \( \mathcal{M}, \mathcal{N} \in \text{FNL}^1 \text{ Structure} \) \( (\forall \gamma \in \text{SRT}) \mathcal{M}_\gamma = \mathcal{N}_\gamma \) and
\( (\forall 'op' \in \text{SOP}, \text{sign} 'op' = \langle \gamma, \bar{a}, \bar{\beta} \rangle) \) \( (\forall \bar{h} \in \prod_{i:1..m} \mathcal{M}_{\mathcal{Op}}^\bar{a}_i \cap \prod_{i:1..m} \mathcal{N}_{\mathcal{Op}}^\bar{a}_i) \mathcal{M}_{\mathcal{Op}}(\bar{h}) = \mathcal{N}_{\mathcal{Op}}(\bar{h}) \) (for \( m = 0, \bar{h} = () : \mathcal{M}_{\mathcal{Op}}(\bar{h}) = \mathcal{N}_{\mathcal{Op}}(\bar{h}) \) ) then \( (\forall 'e' \in \bigcup_{\gamma \in \text{SRT}} \mathcal{L}_\gamma) 'e'_{\mathcal{M}} = 'e'_{\mathcal{N}} \)

**Proof.** syntactic induction on \( 'e' \) \( \square \)

4.9 Conclusion If \( \mathcal{M} \) is characterized by \( \forall \gamma \mathcal{M}_\gamma = \mathcal{M}_\gamma \forall \gamma, \bar{\sigma} \mathcal{M}_\gamma^\bar{\sigma} = \mathcal{M}_\gamma^\bar{\sigma} \) and \( 'op' \in \prod_{i:1..m} \mathcal{M}_{\mathcal{Op}}^\bar{a}_i \cap \mathcal{M}_{\mathcal{Op}} = \mathcal{M}_{\mathcal{Op}} \) then \( (\forall 'e' \in \bigcup_{\gamma \in \text{SRT}} \mathcal{L}_\gamma) 'e'_{\mathcal{M}} = 'e'_{\mathcal{N}} \)

5 Syntactic Matters and the Calculus of Functional Logic

The logical axioms depend on the syntactic notions free variables, bound variables of an expression \( 'e' \in \mathcal{L}_\gamma \), substitutability and substitution (of a variable in an expression for some term).

5.1 Notation \( [\bar{a}] \) symbolizes the set of components of an arbitrary finite sequence \( \bar{a} \) (if \( \bar{a} = ('a_i')_{i:1..\ell} \), then \( [\bar{a}] = \{ 'a_i' \mid i:1..\ell \} \))

5.2 Definition \( \text{frV} 'e' \), \( \text{bdV} 'e' \) (free and bound variables in \( 'e' \in \mathcal{L}_\gamma \)). Provided that \( 'e' \in \mathcal{L}_\gamma \) and \( \text{Generation-Premise} \) we define inductively:

| cases  | \( \text{frV} 'e' = \) | \( \text{bdV} 'e' = \)         |
|--------|------------------------|---------------------------------|
| \( m = 0 \) | \{ 'op' \} \cap \text{VAR} | \( \emptyset \)                   |
| \( m > 0 \) | \( \bigcup_{i:1..m} (\text{frV} 'a_i' \setminus [\bar{v}_i]) \) | \( \bigcup_{i:1..m} (\text{bdV} 'a_i' \cup [\bar{v}_i]) \) |

**Remark:** If \( 'op' \in \text{VAR} \) then \( \{ 'op' \} \cap \text{VAR} = \{ 'op' \} \), otherwise \( \{ 'op' \} \cap \text{VAR} = \emptyset \).

5.3 Observation
(1) \( 'e' \in \mathcal{L}_\gamma[\bar{u}] \leftrightarrow 'e' \in \mathcal{L}_\gamma \land \text{frV} 'e' \subseteq [\bar{u}] \)
(2) \( 'e' \in \mathcal{L}_\gamma[\bar{u}] \rightarrow ([\bar{u}] \cap W = \emptyset \rightarrow \text{frV} 'e' \cap W = \emptyset) \)

5.4 Definition Substitutability: \( \text{Subb} \subseteq \mathcal{L}_S \times \text{VAR} \times \mathcal{L}_S \)
Generation-Premise \( \rightarrow \) Subb(‘d’, ‘x’, ‘e’) ⇔
\( \leftrightarrow (\forall i : 1 \ldots m) \left( \langle x \rangle \in [v]_i \lor (\text{Subb}(d', x', a_i) \land \text{frV} \{d' \} \cap [v]_i = \emptyset) \right) \)

5.5 Notation \( \langle e[x \leftarrow d'] \rangle \overset{\text{def}}{=} e^d_\alpha \) denotes the result of replacing each free occurring ‘x’ by ‘d’ applied to ‘e’. This is a special case of next Definition (with \( l = 1 \)).

5.6 Definition \( \langle e[x \leftarrow d] \rangle \overset{\text{def}}{=} e^d_\alpha \) (for \( e \in L_Y \), \( \sigma = (\sigma_i)_{i:1..l} \in \text{VSrt}^\ell \), \( x = \langle 'x_1 \rangle \), \( \ell \in \text{VAR}_\sigma \), \( d = \langle 'd_1 \rangle \) \( \in L_\sigma \)) denotes the result of simultaneously replacing each free occurring ‘x\( _i \)' by ‘d\( _i \)' (i : 1..\( \ell \)) applied to ‘e’. If a variable appears more than once within the sequence \( x \), the rightmost \( d_\ell \) of the corresponding position replaces the variable. This is defined inductively: if Generation-Premise is supposed, then

| cases       | ‘e[x ← d]’ =                                                                 |
|-------------|------------------------------------------------------------------------------|
| m = 0       | ‘\text{op}’ \in [x] = ‘d_k’ with \( k = \max_{i:1..\ell}('\text{op}’ = 'x_i') \) |
| otherwise   | = ‘\text{op}’’ (= ‘\text{e}’).                                               |
| m > 0       | = ‘\text{op}[\ldots, [([v]_i): a_i[\bar{x}_v \leftarrow \bar{d}_v], \ldots]’ |

Substitution \( [\bar{x}_v \leftarrow \bar{d}_v] \) differs from \([x ← d] \) exactly if the sequences \( x \) and \( v \) have common members. If \( 'x_i' = 'v_{ij} \) then the replacement of \( x_i \) by \( d_i \) is prohibited in the substitution \([\bar{x}_v \leftarrow \bar{d}_v] \). This prevents replacing bound variables.

II. Notational Clause Inside of [...] an identifier \( \alpha \) for any expression ‘\( \alpha' \) \( \in L_\alpha \) denotes the sequence (‘\( \alpha' \)) of length 1. E.g. \( e[x\gamma z ← a\beta c] \) is to be read as \( e[x('y')z ← a('b')c] \).

Laws of substitution enumerated within the subsequent five lemmas shall prove to be essential prerequisites for propositions concerning the term structure obtained from a consistent theory in our final section. The proofs of these (intuitively clear) lemmas to 5.11 mainly rely on syntactic induction using Generation-Premise.

5.7 Lemma Let \( e' \in L_Y \), \( \bar{\sigma}, \bar{\bar{\sigma}} \in \text{VSrt}^* \), \( \bar{u} \in \text{VAR}_\bar{\sigma} \), \( \bar{z} \in L_{\bar{\bar{\sigma}}} \).
If \( \text{frV} \{e'\} \cap [\bar{u}] = \emptyset \) then \( 'e[\bar{u} ← \bar{z}]' \) = ‘e’. This is a special case of the next

5.8 Lemma Let \( e' \in L_Y \), \( \bar{y} \in \text{VAR}_\bar{\sigma} \), \( \bar{u} \in \text{VAR}_\bar{\sigma} \), \( \bar{\eta}, \bar{\bar{\sigma}} \in \text{VSrt}^* \), \( \bar{f} \in L_{\bar{\bar{\sigma}}} \), \( \bar{s} \in L_{\bar{\eta}} \).
If \( \text{frV} \{e'\} \cap [\bar{u}] \subseteq [\bar{y}] \) then \( 'e[\bar{u}\bar{y} ← \bar{f}\bar{s}]' \) = ‘e[\bar{u} ← \bar{s}]’.

Proof. From the premises (i) \( \text{frV} \{e'\} \cap [\bar{u}] \subseteq [\bar{y}] \) (ii) Generation-Premise and (iii) induction hypotheses we shall infer the succedent left =right..

Case \( m = 0 \). Then ‘op’ and \( \text{frV} \{e'\} \cap \text{VAR} \). By (i) and the fact, that \([\bar{u}] \subseteq \text{VAR} \) we obtain (iv) \{‘e’\} \cap [\bar{u}] \subseteq [\bar{y}] \).
Case ‘e’ ∈ [u[y]]. Then (iv) implies (v) ‘e’ ∈ [y]. According to 5.6 def., case ‘op’ ∈ [...] on the preceding page we have

(vi) left = ‘e[u[y] ← r[y]]’ = pj_k(r[y]), where k is the maximal within range 1 ... ℓ_m, so that pj_k(u[y]) = ‘e’.

(v) implies that (vii) k = ℓ_θ + j, where j is the maximal within range 1 ... ℓ_θ so that pj_j(y) = ‘e’.

(vi)+(vii) yield left = pj_ℓ_θ+j(r[y]) = ‘s_j’ and right = ‘e[y] ← s’ = ‘s_j’; left = right.

Case ‘e’ /∈ [u[y]]. According to 5.6 def., case ‘op’ /∈ [...], on the page before left = right.

Case m ≠ 0.

(iv) (frV ‘a_1’ \ [v_i]) ∩ u ⊆ y

(v) frV ‘a_1’ ∩ [u] ⊆ [y] ∩ [v_i] = [y[v_i]] as A ⊆ B implies A ∪ [v_i] ⊆ B ∪ [v_i]

(vi) left = ‘e[u[y] ← r[y]]’ = ‘op([v_i]; ] a_1[u[y]v_i ← r[y]]v_i[ ])’

= ‘op([v_i]; ] a_1[yv_i] ← s[v_i][ ])

□

5.9 Lemma Let ‘e’ ∈ L_y, x ∈ VAR_L y ∈ VAR_θ, r_θ, y_θ ∈ VStr, r ∈ L_θ, s ∈ L_θ

If \( \bigcup \) frV ‘r_j’ ∩ ([y] ∪ bdV ‘e’) = ∅ then ‘e[x[y] ← r[y]]y ← s[y]’.

Remark If r ∈ L_θ[ ], then the last condition is true.

Proof. From the premises (ii,iii) we shall infer left = right (succedent of the lemma).

(i) premises of lemma; (ii) Generation-Premise; (iii) induction hypothesis

Case m = 0. Then ‘e’ = ‘op’

Case ‘e’ /∈ [x[y]]. Then left = ‘e’ = right (according to 5.6 def. on the page before)

Case ‘e’ ∈ [y] \ [u[y]]. Then left = ‘r_m’ , where m is the maximal m ≤ ℓ_y so that ‘e’ = ‘x_m’. Hence right = ‘r_m[y ← s]’. (i) and 5.7 yield right = ‘r_m’; left = right

Case ‘e’ ∈ [y]. Then left = ‘s_m’ where m is the maximal so that ‘e’ = ‘y_m’ , hence right = ‘y_m[y ← s]’ = ‘s_m’, left = right.

Case m > 0. Now the premises (ii,iii) become relevant. Again we infer left = right.

(iv) left = ‘op([v_i]; ] a_1[x[y]v_i ← r[y]]v_i[ ])

As [v_i] ⊆ bdV ‘e’ (i) implies \( \bigcup \) frV ‘r_j’ ∩ ([y[v_i]] ∪ bdV e) = ∅, then (iii) yields

(v) ‘a_1[x[y]v_i ← r[y]]v_i[ ]’ = ‘a_1[x[y]v_i ← r[y]]v_i[ ]’

Substituting in (iv) and application of 5.6 yield
(vi) $\text{left} = \text{op}[\ldots[(v'\cdot a_1[\bar{x}\bar{y}\bar{u}] \leftarrow \bar{r}\bar{y}\bar{v}][\bar{y} \leftarrow \bar{s}]]$ if $r_2 > 0$

$$\text{left[\bar{x}\bar{y} \leftarrow \bar{r}\bar{y}][\bar{y} \leftarrow \bar{s}]} = \text{right}$$

\[5.4\]

5.10 Lemma Let $e' \in L_x \quad x \in \text{VAR}_x \quad \bar{y} \in \text{VAR}_y \quad \rho, \bar{r} \in \text{VSRT}^* \quad 'r' \in L_x$.

'\text{left} = \text{left}'

Proof. Again rely on Generation-Premise and use induction, the result then comes immediately from 5.6 def.

For the purpose of proving the next lemma we observe

**Sublemma 1** Let $e' \in L_x \quad \bar{x} \in \text{VAR}_x \quad \bar{y} \in \text{VAR}_y \quad \bar{x}, \bar{r} \in \text{VSRT}^* \quad 'r' \in L_x$.

If $|\bar{x}| \cap |\bar{y}| = \emptyset$ then 'left[\bar{x}\bar{y} \leftarrow \bar{r}\bar{y}] = 'left[\bar{x} \leftarrow \bar{r}]' .

5.11 Lemma Let $e' \in L_x \quad \bar{x} \in \text{VAR}_x \quad \bar{y} \in \text{VAR}_y \quad \bar{x}, \bar{r} \in \text{VSRT}^* \quad \bar{c} \in L_x \quad \bar{d} \in L_y$.

If $|\bar{x}| \cap |\bar{y}| = \emptyset$ then 'left[\bar{x}\bar{y} \leftarrow \bar{c}\bar{d}] = 'left[\bar{x} \leftarrow \bar{c}]|\bar{y} \leftarrow \bar{d}]' = 'left[\bar{y} \leftarrow \bar{d}]|\bar{x} \leftarrow \bar{c}]' .

Proof. Proving the sublemma is as easy as for the previous lemma. For the current lemma two equations are considered. From 5.9+remark and the preceding sublemma we immediately obtain the first ('left[\bar{x}\bar{y} \leftarrow \bar{c}\bar{d}] = 'left[\bar{x} \leftarrow \bar{c}]|\bar{y} \leftarrow \bar{d}]') . In order to enable the induction step for the second equation we prove more generally for arbitrary $\bar{u} \in \text{VAR}^*$

(*)

\[e[\bar{x}\bar{y}\bar{u} \leftarrow \bar{c}\bar{d}\bar{u}] = 'e[\bar{y}\bar{x}\bar{u} \leftarrow \bar{d}\bar{c}\bar{u}]' .

Assume premises of the lemma, Generation-Premise and induction hypothesis.

Case $m = 0$. Then 'left' = 'op'

Case $e' \notin |\bar{x}\bar{y}\bar{u}]$. Then $\text{left} = e' = \text{right}$ (according to 5.6 def. on page 11, 2nd line of the table)

Case $e' \in |\bar{u}]$. Then $e' = 'u_p' . From 5.4 def. (1st line of the table) and the observation, that the rightmost occurrence of 'e' within \(\bar{x}\bar{y}\bar{u}\) belongs to \(\bar{u}\) we infer $\text{left} = p_j t_{x'} t_{y'} p (\bar{x}\bar{y}\bar{u}) = 'u_p' = e' . and by obvious symmetric consideration $\text{right} = e' .

Case $e' \in |\bar{x}] \setminus |\bar{u}]$. Then, as $|\bar{x}] \cap |\bar{y}] = \emptyset$ is assumed, $e' \notin |\bar{y}\bar{u}]$ and 5.4 def., 1st line of table yields $\text{left} = c_m , where m is the maximal $m \leq \ell_z$ so that $e' = 'x_m' . Then $\ell_{\bar{g}} + m is the corresponding position for right, i.e. $\text{right} = p_j t_{x'} m (d\bar{c}\bar{u}) = p_j m (e) = 'c_m' ; left = right .

Case $e' \in |\bar{y}] \setminus |\bar{u}] is obviously similar to the preceding case (exchange x with y as well as left with right)
Case $m > 0$. From Generation-Premise and $\exists \theta$ def., 3rd line of table we obtain

\[\text{left}= \text{op}\left(\exists \theta \left[ (\forall \theta_i) : a_1 [x \bar{\theta}_1^\prime ] \leftarrow \bar{d} \bar{u}_1^\prime \right] \right)\]

\[\text{right}= \text{op}\left(\exists \theta \left[ (\forall \theta_i) : a_1 [y \bar{\theta}_1^\prime ] \leftarrow \bar{d} \bar{u}_1^\prime \right] \right)\]

Replacing the argument-expressions according to the induction hypothesis (‘$a_1$’ in place of ‘$e$’ and $\bar{u}_1^\prime$ in place of $\bar{u}$) yields \(\text{left} = \text{right}\).

III. Notational Clauses

(1) We write ‘\(\forall x \varphi\)’ for ‘\(\forall x \in \text{Var}_l\)’, and (2) ‘\(\forall \bar{z} \varphi\)’ for ‘\(\forall z_1 \ldots z_k \in \text{Var}_l\)’. This includes case \(l = 0\), as for \(\bar{u} = \langle \rangle\) ‘\(\forall \bar{u} \varphi\)’ = ‘\(\varphi\)’ is stipulated. (3) ‘\(a = b\)’ is an alias for ‘\(a \equiv b\)’ if \(a, b \in \mathcal{L}_y\).

5.12 Calculus of Functional Logic

\(\text{Fnl} = \text{Fnl}_{\pi}\) is defined for \(\text{Fnl}^1\) Signa S as a triple of the component sets formulae, axioms and rules specified as follows:

**Formulae** \(\mathcal{L}_\pi\) (s. 3.2 on page 3). **Axioms** Propositional Tautologies.

**Predicate Logic** Axioms. Like in 1st order predicate logic but related to the extended notions of free vars., bound vars., substitutability and substitution. The Axioms are ‘\(\forall x \varphi \rightarrow \varphi^x_a\)’, ‘\(\varphi^x_a \rightarrow \exists x \varphi\)’ (meta condition Subb(‘a’, ‘x’, ‘\varphi’)) provided

\(\forall x (\varphi \rightarrow \psi) \rightarrow (\psi \rightarrow \forall x \varphi)\), ‘\(\forall x (\varphi \rightarrow \psi) \rightarrow (\exists x \varphi \rightarrow \psi)\)’ (if ‘x’ \(\not\in\text{frV}\) ‘\(\psi\)’).

**Equality Axioms**

(11) ‘\(a \equiv a\)’ (for \(a \in \text{SRT}\), ‘\(a \in \mathcal{L}_\alpha\)’)

(12) ‘\(\forall \bar{z} \; b_1[x \bar{z} \leftarrow \bar{z}] \equiv b_2[\bar{y} \bar{z} \leftarrow \bar{z}] \rightarrow \text{op}(\Delta, [\bar{x} : ] b_1, \Gamma) \equiv \text{op}(\Delta, [\bar{y} : ] b_2, \Gamma)\)’

\[\text{if } \tau_i > 0 \quad \quad \quad \quad \text{if } \tau_i > 0\]

where ‘op’ \(\in\) SOP, sign ‘op’ = \(\langle r, \alpha, \beta, \gamma \rangle\), \(x, y, z \in \text{Var}_{\beta}\), \(b_1, b_2 \in \mathcal{L}_{\alpha}\), \([\bar{z}] \cap (\text{frV} \cdot b_1 \cup \text{frV} \cdot b_2) = \emptyset\) and \(\Delta\) and \(\Gamma\) may be further arguments or empty.

**Rules** Detachment ‘\(\varphi\)’, ‘(\(\varphi \rightarrow \psi\))’ \(\rightarrow \psi\)’ and Generalization : ‘\(\varphi\)’ \(\rightarrow \forall x \varphi\)’ (‘\(\varphi\)’, ‘\(\psi\)’ \(\in \mathcal{L}_\pi\), ‘\(x\)’ \(\in\text{Var}\)’).

Remark: For \(\alpha \in \text{VSRT}\) scheme (11) could be replaced by a single axiom ‘\(x \equiv x\)’ (for one fixed ‘x’ \(\in \text{Var}_{\alpha}\)).

5.13 Notation ‘\(\varphi_1\), \ldots, \(\varphi_m\)’ \(\vdash \psi\) (m \(\geq 0\)) is used for ‘\(\psi\) can be inferred or deduced within the calculus \(\text{Fnl}_{\pi}\) augmented by premises or additional axioms \(\varphi_1, \ldots, \varphi_m\) if \(m > 0\)’. We shall write only \(\vdash\) for \(\vdash_{\pi}\) if reference to S is clear.

5.14 Laws of equality (Let \(a, b, c \in \mathcal{L}_y\))

(13) \(\vdash \text{a} = \text{b} \rightarrow \text{b} = \text{a}\) (symmetry of equality)

(14) \(\vdash \text{a} = \text{b} \rightarrow \text{b} = \text{c} \rightarrow \text{a} = \text{c}\) (transitivity of equality)

**Proof.** (13) \(\Leftarrow \quad \text{a} \equiv \text{b} \rightarrow (\text{a} \equiv \text{a}) \equiv (\text{b} \equiv \text{a})\) is an inst. of (12),

\[\rightarrow \text{a} \equiv \text{a} \rightarrow \text{b} \equiv \text{a}\]

as ‘\(\equiv\)’ coincides with ‘\(\equiv\)’ interchanging the premises by virtue of a propositional tautology and detaching \(a = a\) yields (13)

(14) \(\Leftarrow \quad \text{a} \equiv \text{b} \rightarrow (\text{a} \equiv \text{c}) \equiv (\text{b} \equiv \text{c})\) is an instance of (12)

\[\rightarrow \text{b} = \text{c} \rightarrow \text{a} = \text{c}\]

as ‘\(\equiv\)’ coincides with ‘\(\equiv\)’ \(\Box\)
5.15 Equality Theorem Let \( 'e' \in \mathcal{L} \) \( 'r' \), \( 's' \in \mathcal{L}_q \) \( \vec{y} \in \text{VAR}^* \) and \( 'z' \in \text{VAR}_r \).

(i5) If \( \text{bdV} 'e' \cap \text{frV} 'r' = 's' \subseteq [\vec{y}] \) then \( \vdash '\forall \vec{y} (r = s) \rightarrow e[z \leftarrow r] = e[z \leftarrow s] ' \)

**Proof.** Assume (1) the premises of the theorem, (2) Generation-Premise, (3) induction premise, then we observe \( 'e' = \text{op}(\ldots, [\vec{v}_i]; \alpha_1, \ldots) ' \) and infer the succeeding

(4) \( \text{bdV} 'e' = \bigcup_{i=1}^{m} (\text{bdV} 'a_i' \cup [\vec{v}_i]) \)

therefore \( \text{bdv} 'a_i' \subseteq \text{bdv} 'e' \),

(5) \( \text{bdv} 'a_i' \cap \text{frv} 'r' = 's' \subseteq [\vec{y}] \);

(6) \( 'e[z \leftarrow t]' = \text{op}(\ldots, [\vec{v}_i]; \alpha_i[\vec{z}_i \leftarrow \vec{v}_i], \ldots) ' \) by 5.6 def. on page 11

(with \( t ' \) intended to be replaced both by \( 'r ' \) and \( 's ' \))

**Case** '\( z \in [\vec{v}_i] ' \):
- \( 'a_i[z, \vec{v}_i \leftarrow t, \vec{v}_i]' = 'a_i ' \) by 5.10 lemma on page 13
- \( 'a_i[z, \vec{v}_i \leftarrow r, \vec{v}_i]' = 'a_i[z, \vec{v}_i \leftarrow s, \vec{v}_i]' ' \) hence

\( \vdash 'a_i[z, \vec{v}_i \leftarrow r, \vec{v}_i]' = 'a_i[z, \vec{v}_i \leftarrow s, \vec{v}_i]' ' \) (instance of (I1))

**Case** '\( z \notin [\vec{v}_i] ' \):
- \( 'a_i[z, \vec{v}_i \leftarrow t, \vec{v}_i]' = 'a_i[z \leftarrow t]' ' \) from 5.11 lemma on page 13
- \( \vdash 'a_i[z \leftarrow r] = a_i[z \leftarrow s]' ' \) by (3)
- \( \vdash 'a_i[z, \vec{v}_i \leftarrow r, \vec{v}_i]' = 'a_i[z, \vec{v}_i \leftarrow s, \vec{v}_i]' ' \) from the preceding 2 lines

Both cases yield \( \vdash '\forall \vec{y} (r = s) \rightarrow \forall \vec{v}_i (a_i[z, \vec{v}_i \leftarrow r, \vec{v}_i]' = 'a_i[z, \vec{v}_i \leftarrow s, \vec{v}_i]' ' \).

Let \( 'p_i' = '\{[\vec{v}_i]; \alpha_i ' \) \( 'p_i' = 'a_i[z, \vec{v}_i \leftarrow r, \vec{v}_i]' ' \) \( 'Q_i ' = '\{[\vec{v}_i]; \alpha_i ' \) \( 'q_i ' = 'a_i[z, \vec{v}_i \leftarrow s, \vec{v}_i]' ' \)

Then \( \vdash '\forall \vec{y} (r = s) \rightarrow \forall \vec{v}_i (p_i = q_i)' ' \)

\( \vdash '\forall \vec{y} (p_i = q_i) \rightarrow \text{op}(P_1, \ldots, P_{l-1}, P_l, Q_{l+1}, \ldots) = \text{op}(P_1, \ldots, P_{l-1}, Q_l, Q_{l+1}, \ldots)' ' \)

By chaining these implications and using (I4) on p. on the preceding page we obtain

\( \vdash '\forall \vec{y} (r = s) \rightarrow \text{op}(P_1, \ldots, P_m) = \text{op}(Q_1, \ldots, Q_m)' ' \)

note that \( '\text{op}(\ldots, P_{l}, \ldots)' = 'e[z \leftarrow r]' ' \) and \( '\text{op}(\ldots, Q_{l}, \ldots)' = 'e[z \leftarrow s]' ' \), hence

\( \vdash '\forall \vec{y} (r = s) \rightarrow e[z \leftarrow r] = e[z \leftarrow s]' ' \). This concludes the induction step. \( \square \)

5.16 Corollary (Equality Rule) \( 'r = s ' \) \( \vdash 'e[z \leftarrow r] = e[z \leftarrow s]' ' \)

6 Formalized Theories

Throughout the paragraph we assume \( S \in \text{Fn}_1 \text{ Signat} ' \).

6.1 Definition A functional logic theory is an extension of the calculus \( \text{Fn}_1 \text{ Signat} ' \) by adding a set \( \mathcal{T} \subseteq \mathcal{L}_q^* ' \) to the axioms. The resulting system is symbolized by \( \text{Fn}_3(\mathcal{T}) ' \). Members of \( \mathcal{T} ' \) are called nonlogical axioms of the system. (We shall often refer to \( \text{Fn}_3(\mathcal{T}) ' \) when using \( \mathcal{T} ' \)
6.2 Notation \( T \models \varphi \) iff there is a proof of \( \varphi \) within the formal system \( \text{Fn}_S[T] \). (alias \( \varphi \) is a theorem of \( T \), or can be inferred from \( T \)) We shall omit \( S \) and write \( T \models \varphi \) instead if only one \( S \) is considered. In the sequel we shall always assume \( \text{Fn}^1 \text{Signat} S \).

6.3 Definition Consistent\( (T) \leftrightarrow T \subseteq L_\pi \wedge T \not\models \neg \varphi \)

6.4 Compactness Theorem Each theorem of \( T \subseteq L_\pi \) is inferable from a finite subset of \( T \), that is \( T \subseteq L_\pi \rightarrow T \models \varphi \leftrightarrow \exists \Delta (\Delta \subseteq T \land \text{finite}(\Delta) \land \Delta \models \varphi) \)

Proof. A (formal) proof only comprises a finite number of axioms, hence only a finite subset of \( T \).

6.5 Definition \( T \) is called complete relative to \( S \), if \( T \subseteq L_\pi^S \) and each closed formula of \( S \) is decidable in \( T \), that is \( \text{Complete}_S(T) \leftrightarrow T \subseteq L_\pi^S \land (\forall \varphi \in L_\pi^S) (T \models \varphi \lor T \not\models \neg \varphi) \)

(Dependency from \( S \) is significant, but we shall omit \( S \) if confusion is impossible)

6.6 Definition \( S_1 \subseteq S_2 \) symbolizes (for \( \text{Fn}^1 \text{Signat} S_1, S_2 \): “\( S_2 \) is an extension of \( S_1 \)”, i.e. all components of \( S_1 \) and \( S_2 \) but \( \text{Sop} \) and sign agree, \( \text{Sop}_{S_1} \subseteq \text{Sop}_{S_2} \) and \( \text{sign}_{S_1} = \text{sign}_{S_2} \).

6.7 Deduction Theorem (a special form) \( \Gamma \subseteq L_\pi \land ' \varphi ' \in L_\pi[] \land ' \psi ' \in L_\pi \rightarrow (\Gamma \cup \{ ' \varphi ' \}) \models ' \psi ' \rightarrow \Gamma \models ' \varphi \rightarrow ' \psi ' \)

Proof. see [SHO67], p.33

6.8 Theorem of Lindenbaum For a consistent theory a consistent complete simple extension exists.

\( T \subseteq L_\pi^S \land \text{Consistent}(T) \rightarrow \exists T_2 (T \subseteq T_2 \land \text{Consistent}(T_2) \land \text{Completes}(T_2)) \)

Proof. By Def. of \( \text{Fn}^1 \text{Signat} \) \( \text{Sop}_S \cup \text{VAR}_S \) can be well-ordered. This implicitly applies to the sets of expressions \( L_\gamma \) and \( L_\pi[] \) (= set of closed formulae), too. Hence we may suppose an ordinal enumeration \( \langle ' \varphi_\alpha ' \rangle_{\alpha \in } \) of \( L_\pi[] \), that is

\[ L_\pi[] = \{ ' \varphi_\alpha ' \mid \alpha < \kappa \} \]

We claim the following Lemma:

\( \forall T \forall ' \psi ' (T \subseteq L_\pi \land ' \psi ' \in L_\pi[] \land \text{Consistent}(T) \rightarrow \text{Consistent}(T \cup \{ ' \psi ' \}) \lor \text{Consistent}(T \cup \{ ' \neg \psi ' \})) \)

For if the opposite is assumed, then \( \neg \text{Consistent}(T \cup \{ ' \psi ' \}) \) and \( \neg \text{Consistent}(T \cup \{ ' \neg \psi ' \}) \). By 6.3 then \( T \cup \{ ' \psi ' \} \models ' \varphi ' \) and \( T \cup \{ ' \neg \psi ' \} \models ' \neg \varphi ' \). By
(4.3) limit ordinal $\alpha \in (\kappa + 1)$

Subproof. If $\neg \text{Consistent}(\bigcup_{\alpha \in \lambda} \mathcal{T}_\alpha)$, then by (4.2) (compactness theorem) there are $\theta_1, \ldots, \theta_k$ so that $(\ast)$ $\theta_1, \ldots, \theta_k \in \bigcup_{\alpha \in \lambda} \mathcal{T}_\alpha \land \{\theta_1, \ldots, \theta_k\} \not\vdash \lambda$. Thus, there are $\alpha_1, \ldots, \alpha_k < \lambda$, so that $(\forall i:1..k) \theta_i \in \mathcal{T}_{\alpha_i}$. Let $\beta = \bigcup_{i:1..k} \alpha_i$ then also $\beta < \lambda$ and $(\theta_1, \ldots, \theta_k) \subseteq \mathcal{T}_\beta$, due to $(\ast)$ then $\neg \text{Consistent}(\mathcal{T}_\beta)$ contradicting the antecedent of implication (4.3). This confirms (4.3). By \textit{ordinal induction} (limited to range $\kappa + 1$) we obtain (5); By (1) using \textbf{Lemma} we infer (6) (Subproof below).

(5) $\text{Consistent}(\mathcal{T}_\kappa)$

(6) Complete($\mathcal{T}_\kappa$)

\textit{Subproof.} According to (1) for an arbitrary formula $\psi \in \mathcal{L}_\kappa[] \alpha < \kappa$ exists so that $\psi = \varphi_\alpha$. (3.2) yields $\psi \in \mathcal{T}_{\alpha + 1} \lor \neg \psi \in \mathcal{T}_{\alpha + 1}$. From $\alpha + 1 \leq \kappa$ by (3.1-3.3) obviously $\mathcal{T}_{\alpha + 1} \subseteq \mathcal{T}_\kappa$, hence $\psi \in \mathcal{T}_\kappa \lor \neg \psi \in \mathcal{T}_\kappa$. $\psi \in \mathcal{T}_\kappa$ implies $\mathcal{T}_\kappa \not\vdash \neg \psi$, hence $\mathcal{T}_\kappa \vdash \psi \lor \mathcal{T}_\kappa \vdash \neg \psi$. This confirms $\square$. 

\section*{6.9 Theorem (Henkin)}

For a theory $\mathcal{T}$ a conservative extension $\mathcal{T}'$ and a mapping

$$
\varphi \mapsto c_\varphi : \bigcup_{\mathcal{S} \in \text{VAR}} \mathcal{L}^{\mathcal{S}}_\kappa[\cdot] \rightarrow \text{COP}_\mathcal{S},
$$

so that for each $\cdot \in \text{VAR}$ and $\varphi \in \mathcal{L}^{\mathcal{S}}_\kappa[\cdot]$: $$(\exists x \varphi \rightarrow \varphi[x \leftarrow c_\varphi]) \in \mathcal{T}'$$ and sign $'c_\varphi' =$ sign $'\mathcal{S}'$. \textit{Conservative extension} means $S \subseteq S' \subseteq \mathcal{L}^{\mathcal{S}}_\kappa \subseteq \mathcal{L}^{\mathcal{T}}_\kappa, \mathcal{T} \subseteq \mathcal{T}'$ and $(\forall \psi \in \mathcal{L}^{\mathcal{S}}_\kappa) (\mathcal{T}' \vdash \psi \rightarrow \mathcal{T} \vdash \psi)$

\textbf{Proof.} (as in [SHO67], p.46) Starting with $S_0 = S$ we inductively define a sequence of extensions: from $S_k$ we obtain $S_{k+1}$ by adding new constant symbols $'c_\varphi'$, each for one $\varphi \in \mathcal{L}^{S_k}_\kappa[\cdot] \land \mathcal{L}^{S_{k-1}}_\kappa[\cdot]$, $\cdot \in$ VAR (if $k = 0$ suppress $'\land \mathcal{L}^{S_{-1}}_\kappa[\cdot]$) and signs $S_{k+1}$ $'c_\varphi' =$ sign $S_k$ $'\varphi$' if $'\varphi$' $\in \mathcal{S}_k$ $[\cdot]$. Let $S'$ be the extension of $S$ by adding \{ $'c_\varphi'$ | (\exists k) $'\varphi$' $\in \mathcal{L}^{S_k}_\kappa[\cdot] \land \cdot \in$ VAR \} to component SOP and stipulating signs $'c_\varphi' =$ sign $S_k$ $'\varphi$' for $'\varphi$' $\in \mathcal{L}^{S_k}_\kappa[\cdot]$ and $\cdot \in$ VAR;

let $\mathcal{T}' = \mathcal{T} \cup \{ (\exists x \varphi \rightarrow \varphi[x \leftarrow c_\varphi]) | \varphi \in \mathcal{L}^{S_k}_\kappa[\cdot] \land \cdot \in$ VAR \}.
Claim. \( \mathcal{T}' \) is a conservative extension of \( \mathcal{T} \): Assume (1) \( \mathcal{T}' \models \varphi \) and (2) \( \varphi \in \mathcal{L}_S^\mathcal{T} \).

Using \( \text{6.4 compactness thm. on page 10} \) and \( \text{6.7 deduction thm. on page 16} \), as \( \mathcal{T}' \prec \mathcal{T} \subseteq \mathcal{L}_\mathcal{T}^\mathcal{T} \), we conclude that there are \( \psi_1', \ldots, \psi_m' \in \mathcal{T}' \prec \mathcal{T} \) so that

\[
(3) \quad \mathcal{T} \vdash \psi_1 \rightarrow \ldots \rightarrow \psi_m \rightarrow \varphi' \quad \text{and} \quad \psi_1' = \exists x \theta \rightarrow \theta[x \leftarrow c_\theta]'
\]

By appropriate arrangement we may assume that \( c_\theta \) does not appear in any other \( \psi_j' \) (\( j = 2, \ldots, m \)), hence we may replace \( c_\theta \) by a new variable \( y \) so that

\[
(4) \quad \mathcal{T} \vdash (\exists x \theta \rightarrow \theta_y) \rightarrow \psi_2 \rightarrow \ldots \rightarrow \psi_m \rightarrow \varphi'
\]

applying well known logical rules and theorems we infer

\[
\mathcal{T} \vdash (\exists x \theta \rightarrow \theta_y) \rightarrow \psi_2 \rightarrow \ldots \rightarrow \psi_m \rightarrow \varphi'
\]

The antecedent \( (\exists x \theta \rightarrow \theta_y) \) is a logical theorem (deducible from the variant theorem \( (\exists x \theta \rightarrow \exists y \theta_y) \)), hence we can detach it and obtain

\[
\mathcal{T} \vdash \psi_2 \rightarrow \ldots \rightarrow \psi_m \rightarrow \varphi'.
\]

Iterated application finally yields \( \mathcal{T} \vdash \varphi' \)

6.10 Definition For \( \text{Fnl}_1 \) Structures \( \mathcal{M} \), \( \varphi \in \mathcal{L}_\mathcal{T} \) and \( \mathcal{S} \subseteq \mathcal{L}_\mathcal{T} \) we define:

\[
(1) \quad \mathcal{M} \models \varphi \quad \leftrightarrow \quad (\forall \bar{\sigma} \in \text{VStr}^\mathcal{T}) \quad (\forall \bar{u} \in \text{VAR}_{\bar{\sigma}}) \quad \bar{u} \in \text{persp } \varphi \quad \rightarrow \quad (\forall \bar{x} \in \mathcal{M}_{\bar{\sigma}}) \quad \mathcal{M} \models \varphi(\bar{u})(\bar{x}) = 1_{\mathcal{M}}
\]

\[
(2) \quad \mathcal{M} \models \mathcal{T} \quad \leftrightarrow \quad (\forall \varphi \in \mathcal{T}) \quad \mathcal{M} \models \varphi
\]

\[
(3) \quad \mathcal{T} \models \varphi \quad \leftrightarrow \quad \forall \mathcal{M} (\text{Fnl}_1 \text{ Structures } \mathcal{M} \land \mathcal{M} \models \mathcal{T} \rightarrow \mathcal{M} \models \varphi)
\]

For (1) we say \( \mathcal{M} \) is a model of (or \( \mathcal{M} \) satisfies) \( \varphi \), this also applies to (2) w.r.t. \( \mathcal{T} \). We write \( \models \) for \( \models_{\mathcal{T}} \) as long as only one \( \mathcal{S} \) is considered.

6.11 Observation If the premises of \( \text{4.8 on page 10} \) apply to \( \mathcal{M}, \mathcal{N} \) then \( \mathcal{M} \models \mathcal{T} \) and \( \mathcal{N} \models \mathcal{T} \) are interchangeable. This applies to \( \mathcal{N} = \mathcal{M} \) of \( \text{4.8} \) as well.

Semantics can be reduced to a notion of full structure as \( \overline{\mathcal{M}} \), characterized by \( \overline{\mathcal{M}}_{\gamma} = \overline{\mathcal{M}}_{\gamma}^{\overline{\mathcal{M}}_{\varphi}} \) in place of conditions \( \text{4.1(3)} \) on p. 10.

7 Obtaining a Model of a Consistent Theory

7.1 Extension Theorem Assume \( \mathcal{T} \) is a consistent theory of signature \( \mathcal{S} \) (technically \( \text{Fnl}_1 \text{ Signat } \mathcal{S} \land \mathcal{T} \subseteq \mathcal{L}_\mathcal{T}^\mathcal{T} \land \text{Consistent}(\mathcal{T}) \)), then a so called complete and consistent Henkin theory \( \mathcal{T}' \) exists, which is an extension of \( \mathcal{T} \). That is, in the sequel we shall rely on the following conditions:
7.2 Lemma: If $\mathcal{T}'$ is a Henkin theory due to \ref{lem:persp.T'}, $\varphi \in \mathcal{L}_{\pi}[\vec{z}]$, $\vec{z} \in \caps_r$, $\vec{a} \in \caps_r$, and all components of $\vec{z}$ are different then there are special constants ‘$c_i' \in \caps_{\vec{a}}$, $(i : 1 \ldots r)$ so that for $\vec{c} = \langle c_i \rangle_{1 \ldots r}$: $\mathcal{T}' \vdash \varphi[\vec{z} \leftarrow \vec{c}] \rightarrow \forall \vec{z} \varphi'$

Proof: For ‘$\varphi' \in \mathcal{L}_{\pi}[\vec{x}]$, ‘$d' = \epsilon_{\vec{x} = \vec{a}}'$, $\varphi' \rightarrow \forall \vec{x} \varphi'$, instances of such theorems are ‘$\psi_0 \rightarrow \psi_1' \ldots , \psi_r', \psi_0 \rightarrow \psi_r'$, where each ‘$\psi_i' = \forall \vec{z}_i \ldots \forall \vec{z}_r (\varphi[\vec{z}_{i+1} \leftarrow c_{i+1} \ldots \vec{z}_r \leftarrow c_r])'$. Transitivity of implication yields $\mathcal{T}' \vdash \psi_0 \rightarrow \psi_r'$, according to \ref{lem:7.3} on page 17. ‘$\varphi_0 = \forall \vec{z}_1 \leftarrow c_1 \ldots \vec{z}_r \leftarrow c_r' = \forall \vec{z}_1 \leftarrow c_1, \ldots \vec{z}_r \leftarrow c_r' \varphi[\vec{z} \leftarrow \vec{c}]$; hence $\mathcal{T}' \vdash \forall \vec{z} \varphi'_0$

7.3 Lemma: Assume $\mathcal{T}'$ due to \ref{lem:persp.T'}, $\vec{b} \in \caps_{\vec{a}}$, $\vec{u}, \vec{v} \in \caps_{\vec{a}}$, ‘$\vec{a} = \mathcal{L}_{\gamma}[\vec{a}]$’, ‘$\vec{b} = \mathcal{L}_{\gamma}[\vec{b}]$’ and no component of $\vec{u}$ or $\vec{v}$ occurs more than once. If $(\forall \vec{d} \in \mathcal{L}_{\gamma}[\vec{x}]) \mathcal{T}' \vdash \forall \vec{a} \vec{[u \leftarrow \vec{d}]} = \vec{b}[\vec{v} \leftarrow \vec{d}]$ and if $\vec{z} = \langle \vec{z}_1 \rangle_{1 \ldots t} \in \caps_{\vec{a}}$ is such that each ‘$\vec{z}_i$’ is different from the members of $\vec{u}$ and $\vec{v}$ and neither occurs in ‘$\vec{a}$’ nor in ‘$\vec{b}$’, then $\mathcal{T}' \vdash \forall \vec{z} \vec{a} [\vec{u} \leftarrow \vec{z}] = \vec{b}[\vec{v} \leftarrow \vec{z}]$

Proof. Let ‘$s' = \langle \vec{a} \vec{u} \leftarrow \vec{z} \rangle$’ and ‘$t' = \langle \vec{b} \vec{v} \leftarrow \vec{z} \rangle$’. Our supposition for $\vec{z}$ implies that ‘$[\vec{u} \leftarrow \vec{z}] [\vec{z} \leftarrow \vec{d}]$’ operates exactly like ‘$[\vec{u} \leftarrow \vec{d}]$’, hence ‘$s[\vec{z} \leftarrow \vec{d}] = \vec{a}[\vec{u} \leftarrow \vec{d}]$’ and ‘$t[\vec{z} \leftarrow \vec{d}] = \vec{b}[\vec{v} \leftarrow \vec{d}]$’. Premise $\mathcal{T}' \vdash \forall \vec{z} [\vec{u} \leftarrow \vec{d}] = \forall \vec{z} [\vec{v} \leftarrow \vec{d}]$ then yields $\mathcal{T}' \vdash \forall \vec{z} [\vec{u} \leftarrow \vec{d}] = \forall \vec{z} [\vec{v} \leftarrow \vec{d}]$, by \ref{lem:7.3} def. This is (*) $\mathcal{T}' \vdash \forall \vec{z} (t = s)[\vec{z} \leftarrow \vec{d}]$ for arbitrary $\vec{d} \in \mathcal{L}_{\gamma}$. Now we are ready to use \ref{lem:7.2} the lemma that provides special constants $\vec{c} = \langle \vec{c}_1 \rangle_{1 \ldots t}$ (each ‘$\vec{c}_i' \in \caps_{\vec{a}}$) so that $\mathcal{T}' \vdash \forall \vec{z} (t = s)[\vec{z} \leftarrow \vec{c}] \rightarrow \forall \vec{z} t = s$, detachment with (*) yields $\mathcal{T}' \vdash \forall \vec{z} t = s$, recalling the definition of ‘$s'$ and ‘$t'$ this is $\mathcal{T}' \vdash \forall \vec{z} \forall \vec{z} \forall \vec{z} \langle \vec{b} \vec{v} \leftarrow \vec{z} \rangle = \vec{a}[\vec{u} \leftarrow \vec{z}]$.

7.4 Lemma: Assume $\mathcal{T}'$ due to \ref{lem:persp.T'}, persp.$\mathcal{GP}$ and persp.$\mathcal{GP}^*$ with $\vec{u} = \vec{u}^* = \langle \vec{r} \rangle$. persp.$\mathcal{GP}^*$ denotes the change of all symbolic parameters except ‘op, m and r, by appending a star-superscript performed on persp.$\mathcal{GP}$ (as new names for corresponding
but possibly different things are required).

If \((\forall i:1..m) \ (\forall s \in \mathcal{L}_{\alpha_i}^\parallel) \ \mathcal{T} \vdash \text{‘}a_i^\parallel [v_i^\parallel \leftarrow s] = a_i [v_i \leftarrow s]\) then \(\mathcal{T} \vdash \text{‘}e^* = e\)

**Proof.** Applying the preceding *lemma* to the last premise yields \((\forall i:1..m) \ \mathcal{T} \vdash \text{‘}\forall \exists a_i^\parallel [v_i^\parallel \leftarrow \bar{s}] = a_i [v_i \leftarrow \bar{s}]\) (by appropriate choice of \(\bar{s} \in \text{VAR}_{\alpha_i}\)). Now we are ready to use equality *axioms* (page 4). Iteration application of (12) yields the chain of equations \(\text{op}(P_1,P_2,\ldots,P_m) = \text{op}(Q_1,Q_2,\ldots,Q_m) = \cdots = \text{op}(Q_1,\ldots,Q_m)\) where \(\text{‘}P_i = \text{‘}[(v_i^\parallel): a_i^\parallel]\) and \(\text{‘}Q_i = \text{‘}[(v_i): a_i]\). From (14) *transitivity* we obtain \(\mathcal{T} \vdash \text{‘}L = R\) with \(\text{‘}L = \text{‘}\text{op}(P_1,\ldots,P_m)\) = \(\text{‘}\text{op}(\ldots,[(v_i^\parallel): a_i^\parallel])\) = \(\text{‘}e^*\) and \(\text{‘}R = \text{‘}\text{op}(Q_1,\ldots,Q_m)\) = \(\text{‘}\text{op}(\ldots,[(v_i): a_i])\) = \(\text{‘}e\), hence \(\mathcal{T} \vdash \text{‘}e^* = e\). \(\Box\)

7.5 Definition (norm of closed expressions) Assume \(\text{Fnl}^1\text{Signat} \ S', \ \mathcal{T} \subseteq \mathcal{L}_S^\parallel\).

For each \(\gamma \in \text{SRT}\) and ‘\(e\)’ \(\in \mathcal{L}_S^\parallel\) the *norm* ‘\(e^\parallel\)’ is defined as follows:

If \(\gamma = \pi\) then if \(\mathcal{T} \vdash \text{‘}e\) then ‘\(e^\parallel\)’ = ‘\(\gamma\)’ otherwise ‘\(e^\parallel\)’ = ‘\(\lambda\)’.

If \(\gamma \neq \pi\) then we rely on an ordinal enumeration of \(\mathcal{L}_S\) and define ‘\(e\)’ to be the first within the subset \{‘\(a\)’ \(\in \mathcal{L}_S\) \mid \(\mathcal{T} \vdash \text{‘}a = e\)\}. w.r.t. this enumeration.

7.6 Proposition Let \(\text{Fnl}^1\text{Signat} \ S', \ \mathcal{T} \subseteq \mathcal{L}_S^\parallel\), \(\gamma \in \text{SRT} \setminus \{\pi\}\) and ‘\(e\)’, ‘\(e_1\)’, ‘\(e_2\)’ \(\in \mathcal{L}_S\) then (1) \(\mathcal{T} \vdash \text{‘}e = e_1^\parallel \) \(\supseteq \) (2) \(\mathcal{T} \vdash \text{‘}e_1^\parallel = e_2^\parallel \) \(\iff \) (‘\(e_1\)’ = ‘\(e_2\)’)

(3) ‘\(\phi\)’ \(\in \mathcal{L}_S\) \(\land \) \(\text{Complete}_S(\mathcal{T}) \rightarrow \mathcal{T} \vdash \text{‘}\phi\)’ \(\iff \) ‘\(\bar{\phi}\)’ = ‘\(\gamma\)’

**Proof.** immediately from the preceding definition. \(\Box\)

7.7 Definition (term structure) Let \(\mathcal{T}\) be a *complete and consistent Henkin theory*, i.e. we presuppose the conditions of \(\Box\) except (1) for \(\mathcal{T}\). We define the *term structure* \(\mathcal{X}\) as a function on \(\text{SRT} \cup (\text{SRT} \times \text{VSRT}^*) \cup \text{SOP}\). As we now consider only one *signature* \((S',\) of that \(\mathcal{T}\) and do not refer to \(\mathcal{T}\) and \(S\) we shall omit subscript \(S'\).

(1) \( ( \gamma \in \text{SRT}) \ (\mathcal{X}(\gamma)^{\mathcal{X}} \overset{\text{def}}{=} \mathcal{X}_S(\gamma) = \{‘\gamma’ \mid ‘e \in \mathcal{L}_S\})\)

(2) \( ( \gamma \in \text{SRT}, \bar{\sigma} \in \text{VSRT}^*) \ (\mathcal{X}(\gamma,\bar{\sigma})^{\mathcal{X}} \overset{\text{def}}{=} \mathcal{X}_S^\parallel(\gamma,\bar{\sigma}) = \{‘\gamma(\bar{\sigma})’ \rightarrow ‘\gamma(\bar{\sigma})’; \ ‘\gamma(\bar{\sigma})’ \rightarrow ‘\gamma(\bar{\sigma})’ \} \) \(\text{if } l = 0\)

(3) \( (‘\text{op}\’ \in \text{SOP}, \text{sign} ‘\text{op}\’ = \langle \gamma, \bar{\sigma}, \bar{\beta} \rangle ) \) \(\text{If } m = 0 \ (\mathcal{X} ‘\text{op}\’ = ‘\text{op}\’ ‘\text{op}\’ \)

If \(m \neq 0 \ (\mathcal{X} ‘\text{op}\’ = \{\langle \bar{\alpha}, ‘\gamma’ \rangle \mid \bar{\alpha} = (a_i)_{i:1..m} \in \prod \mathcal{X}_S^\parallel_i \land \exists \bar{\alpha} \exists \bar{\nu} (i)-(iv)\})\)

(i) \(\bar{a} = \langle ‘a_1’ \rangle_i \in \prod_{i:1..m} \mathcal{L}_S^\parallel[v_i] \) (ii) \(\bar{\nu} = \langle v_i \rangle_i \in \prod_{i:1..m} \text{VAR}_{\alpha_i}^\parallel\)
(iii) \( (\forall i : 1..m) \ a_i = \left\{ \begin{array}{ll}
\frac{a_i'}{x(\vec{b}_i)} \rightarrow x(a_i) & \text{if } r_i = 0 \\
\frac{\vec{a}_i'}{b} \rightarrow \vec{a}_i[v_i \leftarrow \vec{b}] & \text{if } r_i \neq 0
\end{array} \right. \)

(iv) \( \vec{e}' = 'op(\ldots,[v_i:] \ a_{i1},\ldots)' \)

7.8 Remark (i)-(iv) in \( \ref{7.5}(3) \) imply \text{persp.GP} with \( \vec{u} = \emptyset \)

7.9 Proposition \( \text{Fn}^1 \ \text{Structure}_S \ X \) \hspace{1cm} (Def. s. \ref{4.1} on page \ref{7})

\textbf{Proof.} We show that the laws of Def. \ref{4.1} on page \ref{7} apply to \( X \) (in place of \( \mathfrak{M} \)) referring to them with the numerator used in that definition. We rely on \textit{lemmas} \ref{6.7} to \ref{6.11} and Def. \ref{7} on page \ref{8} of \( X \). For (1) and (3.1) there is nothing to prove. Our first goal is to prove validity of (2), this only requires verification that \( X \ \text{‘}op\text{’} \) is a function if \( m \neq 0 \). We refer to \( \ref{7.5}(3) \), case \( m \neq 0 \) (def. of \( X \ \text{‘}op\text{’} \)). The \textit{goal} is then reduced to the task of deduction from the \textit{premises} \( \langle \vec{a}, \vec{e}' \rangle \in X \ \text{‘}op\text{’} \), \( \langle \vec{a}^*, \vec{e}'^* \rangle \in X \ \text{‘}op\text{’} \) and \( \vec{a}^* = \vec{d} \) to the \textit{conclusion} \( \vec{e}' = \vec{e}'^* \). \textbf{Subproof (Sketch).} By expansion of 1st and 2nd premise according to \( \ref{7.5}(3) \) (case \( m \neq 0 \)) we obtain (i)-(iv) of \( \ref{7.5}(3) \) and a starred isomorphic variant (i)* - (iv)*. By \( \ref{7.8} \) \textit{remark} we succeed to \text{persp.GP} and \text{persp.GP}* with \( \vec{u} = \vec{u}^* = \emptyset \). Still one condition lacks to use \( \ref{7.4} \) \textit{lemma}. From \textit{premise} \( \vec{a}^* = \vec{d} \) and expanding (iii) of \( \ref{7.5}(3) \) we obtain \textit{equations} \( \vec{a}_i^*[v_i \leftarrow \vec{b}] = \vec{a}_i[v_i \leftarrow \vec{b}] \) for each \( \vec{b} \in X_{\vec{b}_i} \). These \textit{equations} can be transformed into theorems of \( \mathcal{T}' \) according to \( \ref{7.6}(2) \) and then generalized by \( \ref{7.6}(1) \) to apply to each \( \vec{b} \in L_{\vec{b}_i} \), so that finally \( \ref{7.6} \) can be used to deduce \( \mathcal{T}' \vdash \vec{e}' = \vec{e}' \). To present this in detail: \( \ref{7.6}(2) \) yields \( \mathcal{T}' \vdash \vec{a}_i^*[v_i \leftarrow \vec{b}] = \vec{a}_i[v_i \leftarrow \vec{b}] \) for \( \vec{b} \in X_{\vec{b}_i} \). Taking into account, that \( \vec{d}_i \in L_{\vec{d}_i} \) implies \( \vec{d}_i \in X_{\vec{d}_i} \), we obtain \( \forall \vec{d} \in L_{\vec{b}_i} \ \mathcal{T}' \vdash \vec{a}_i^*[v_i \leftarrow \vec{d}] = \vec{a}_i[v_i \leftarrow \vec{d}] \). Referring to \( \ref{7.6}(1) \): \( \mathcal{T}' \vdash \vec{d}_i = \vec{d}_i \) by virtue of \( \ref{5.16} \) \textit{corollary (equality rule)} on page \ref{15} we may replace each \( \vec{d}_i \) by \( \vec{d}_i \), hence \( \vec{d} \) by \( \vec{d} \) and are ready to apply \( \ref{7.4} \), that yields \( \mathcal{T}' \vdash \vec{e}' = \vec{e}' \) and by \( \ref{7.6}(2) \) \( \vec{e}' \vdash \vec{e}' \boxdot (2) \). We turn to the \textit{completion qualities}.

Next goal is (3.2 a) \( \text{cat}^w_{\vec{c}} \in X^w_{\vec{c}} \) for \( w \in X_{\vec{c}} \). \textbf{Subproof.} Suppose \( w \in X_{\vec{c}} \), then (applying (1) of \( \ref{7.5} \)) \( w = \vec{c} ' \) for some \( 'c' \in L_{\vec{c}} \), by \( \ref{5.3} \) on page \( \ref{10} \) \( \forall \vec{v} \ 'c' \vdash \vec{c}' \), hence by \( \ref{5.7} \) on page \( \ref{11} \) \( \vec{c}' = 'c'[\vec{u} \leftarrow \vec{c}] \), then \( \text{cat}^w_{\vec{c}} = \text{cat}^w_{\vec{c}} = \left\lfloor \begin{array}{l}
x_{\vec{v}} \rightarrow \vec{x}_{\vec{v}} \\
x_{\vec{v}} \rightarrow 'c'[\vec{u} \leftarrow \vec{c}]
\end{array} \right. \), that is \( \text{cat}^w_{\vec{c}} \in X^w_{\vec{c}} \)

due to (2) of \( \ref{7.5} \) Def. (3.2 a). Next goal is (3.2 b) \( p_{j}^w \beta = \left\lfloor \begin{array}{l}
x_{\vec{v}} \rightarrow \vec{x}_{\vec{v}} \\
x_{\vec{v}} \rightarrow 'c'[\vec{u} \leftarrow \vec{c}]
\end{array} \right. \in X^w_{\vec{c}} \).

\textbf{Subproof.} Let \( \vec{c} = \langle \vec{c} ' \rangle_{i \in 1..1} \). If we apply \( \ref{7.6} \) Def., case \( m = 0 \), \( 'op' \in [\vec{x}] \), perform a substitution such that the second \textit{condition of case} changes into \( 'u_i' \in [\vec{u}] \) and use \( 'c_j' = 'c_j' \) (as \( 'c_j' \in X_{\vec{c}_j} \subseteq \{ 'c' : 'c' \in L_{\vec{c}_j} \} \)), then we obtain \( p_{j}^w \beta = \left\lfloor \begin{array}{l}
x_{\vec{v}} \rightarrow \vec{x}_{\vec{v}} \\
x_{\vec{v}} \rightarrow 'c'[\vec{u} \leftarrow \vec{c}]
\end{array} \right. \). This matches the pattern of \( \ref{7.5} \) (2) for \( p_{j}^w \beta \in X^w_{\vec{c}_j} \) (3.2 b).

Next goal is (3.3) \( \vec{c} = \langle \vec{c} ' \rangle \in X_{\vec{c}} \wedge \vec{g} \in X_{\vec{g}} \rightarrow \vec{g}_{\vec{c}} \in X_{\vec{g}} \). \textbf{Subproof.} Applying \( \ref{7.5} \) we can...
replace the 2nd premise by \( \mathbf{g} = \left[ x_{\tilde{\beta}} \rightarrow x_{\gamma} \right] \). If we replace \( \mathbf{g} \) in (3.3) by the right hand term, it only remains to show: if \( \tilde{x} \in \text{VAR}_\tilde{\sigma}, \tilde{y} \in \text{VAR}_{\tilde{\rho}}, \) \( \tilde{c} \in \{ \langle c_i \rangle \}_{\tilde{\sigma}} \) \( \in \mathcal{X}_\tilde{\sigma} \), then \( \mathbf{g}_c = \left[ x_{\tilde{\beta}} \rightarrow x_{\gamma} \right] \). The expression for \( \mathcal{Y}_\tilde{\sigma} \) must be transformed in such a manner that it matches the pattern of \( \mathcal{Y}_\gamma \). This is achieved by aid of lemma 5.9: \( \langle c_i \tilde{X} \tilde{Y} \rangle = \langle c_i \rangle \). Substitution yields \( \mathbf{g}_c = \left[ x_{\tilde{\beta}} \rightarrow x_{\gamma} \right] \). According to 7.7 (2) then \( \mathbf{g}_c \in \mathcal{Y}_\gamma \).

Next we show (3.4). Outline: It requires inheritance of the quality \( \mathcal{X}_\beta \) of a map \( \mathcal{X}_\beta \rightarrow \mathcal{X}_\gamma \). When composition of such maps with \( \mathcal{X}_\alpha \) is performed. Quality \( \mathcal{X}_\beta \) applies to \( \mathcal{X}_\beta \rightarrow \mathcal{X}_\gamma \) iff \( \mathcal{X}_\gamma \) can be defined by an expression \( \langle c_i \rangle \) as a substitution mapping \( \gamma \rightarrow \langle c_i \rangle \). To show this inheritance we start from (7.7 (3)), compare the terms that arise with the obtained from quality \( \mathcal{X}_\alpha \). The goal that the composed map \( \mathcal{X}_\beta \rightarrow \mathcal{X}_\alpha \) belongs to \( \mathcal{X}_\beta \) is achieved if we find an \( \langle c_i \rangle \in \mathcal{X}_\gamma \) so that \( \mathcal{X}_\alpha \) is defined by \( \langle c_i \rangle \). The solution will be quite natural \( \langle c_i \rangle \). If \( r_i > 0 \)

Subproof of (3.4). From the premises (p1)–(p4) we infer \( \langle c_i \rangle \). Let \( \tilde{i} \) is understood to be bound and \( \langle \tilde{i} \rangle \) in subsequent context.

From (p4) we infer
\[
\mathbf{g}_i = \begin{cases} \begin{array}{l}
x_{\tilde{\beta}} \rightarrow x_{\alpha}, \quad \mathbf{g}_i = \left[ x_{\tilde{\beta}} \rightarrow x_{\alpha} \right] \end{array} \end{cases}
\]

Our goal is to show \( \langle c_i \rangle \) \( \mathcal{X}_\beta \rightarrow \mathcal{X}_\alpha \) \( \mathcal{X}_\beta \rightarrow \mathcal{X}_\alpha \).

From (p4) we infer
\[
\mathbf{h}_i = \begin{cases} \begin{array}{l}
\mathbf{g}_i \in \mathcal{X}_\alpha, \quad \mathbf{h}_i = \left[ x_{\tilde{\beta}} \rightarrow x_{\alpha} \right] \end{array} \end{cases}
\]

In order to verify \( \langle c_i \rangle \) we look at (7.7 (2) Def. \( \mathcal{X}_\gamma \). It requires transforming of \( \mathcal{X}_\alpha \) into an expression of shape \( \langle c_i \rangle \). This can be achieved by application of (7.7 (3)) if we find a substitute of \( \langle c_i \rangle \) so that \( \mathcal{X}_\alpha \) is \( \langle c_i \rangle \). We have to show \( \langle c_i \rangle \). (7.7 (3)) \( \mathcal{X}_\beta \) \( \mathcal{X}_\alpha \).

This makes evident what to be substituted for \( \langle c_i \rangle \) and the place to be occupied by the \( \langle [\tilde{V}_i] \rangle \) in (p1)–(p4).

22
may be considered. The problem is now reduced to showing

\[ (---) \quad \text{for } r_1 > 0 \quad \overline{c} \in \mathcal{X}_\sigma \quad b_1 \in \mathcal{X}_{\sigma'} \quad \left( a_i | \overline{x} \overline{v}_i \leftarrow \overline{c} b_1 \right) = \left( a_i | \overline{x} \leftarrow \overline{c} | \overline{v}_i \leftarrow b_1 \right) \]

As \( \mathcal{X}_\sigma \subseteq \mathcal{L}_\sigma \), this equation is supplied by \( 5.11 \) lemma, but only if \( |\overline{x}| \cap |\overline{v}_i| = \emptyset \). We can achieve this requirement by renaming the variables \( \overline{v}_i \) using equality axioms (12) of \( 5.12 \) on page 14, namely the following instance:

\[
(p7) \quad \forall \overline{w}_i \quad a_i[\overline{v}_i \leftarrow \overline{w}_i] = a_i[\overline{v}_i \leftarrow \overline{q}_i][\overline{q}_i \leftarrow \overline{w}_i] \to \quad \text{op}(\Delta, ([\overline{v}_i]: \ a_i), \Gamma) = \text{op}(\Delta, ([\overline{q}_i]: \ a_i), \Gamma)
\]

(in place of \( 'b_1' \) at \( 5.12 \) now \( 'a_i[\overline{v}_i \leftarrow \overline{q}_i]' \)). If we choose \( \overline{w}_i \) and \( \overline{q}_i \) so that the variables of the sequences \( \overline{v}_i, \overline{w}_i \) and \( \overline{q}_i \) are pairwise disjoint, then also

\[
'a_i[\overline{v}_i \leftarrow \overline{w}_i] = 'a_i[\overline{v}_i \leftarrow \overline{q}_i][\overline{q}_i \leftarrow \overline{w}_i]' \quad \text{and the antecedent of } p7 \text{ can be detached on account of axiom (11). We are now with with}
\]

\[ \text{if } r_1 > 0 \]

and due to \( 7.4(2) \):

\[ \text{if } r_1 > 0 \]

we can interchange the two terms (left and right side of equation) within the considered context and the requirement for applicability of \( 5.11 \) will be provided. Hence without loss of generality we can assume the required condition for \( \overline{x} \) and \( \overline{v}_i \), too. This confirms \((---)\) and through the chain \((!!) \to (!!!), (!!!) \to (!) \) we succeed to \((---)\). \( 3.4 \) To check (4) of \( 4.1 \) Def. that \( \mathcal{X}_\pi \) is a Boolean algebra with two elements, whose operations are the \( \mathcal{X} \)-values of certain logical connectives: this can be seen from \( 7.7 \) Def. of term structure together with \( 7.7 \) Def. of norm and simple instances of propositional tautologies as e.g. \( 'x \lor y \equiv y' \) (note \( \equiv \) is \( '\leftrightarrow' \)). Finally check (5): From \( 7.7 \) Def. \( \mathcal{X} \), special case \( '\text{op}' = '0' \), only to consider case \( r_1 = 0 \), we obtain \( \mathcal{X} = 0 \) \( ('a_1', 'a_2') = 'a_1 = a_2' \) and have to show \( 'a_1 = a_2' \) iff \( 'a_1 = a_2' \) otherwise \( '\lambda' \). This follows from \( 7.4(2) \) \( 'a_1 = a_2' \) iff \( 'a_1 = a_2' \) and \( 7.4(1) \) if \( 'a_1 = a_2' \) \( 'a_1 = a_2' \).

7.10 \textbf{Theorem} \ Assume \( \mathcal{X} \) is the term structure of \( \mathcal{T} \) due to Def. \( 7.7 \) \( \overline{\sigma} = (\sigma_i)_{i=1..t} \in \text{VSrt}^t, \ \mathcal{X} = \langle \overline{x}_i \rangle_{i=1..t} \in \text{VAR}_{\sigma} \), then

\[
( 'e' \in \mathcal{L}_\mathcal{V}[\mathcal{X}] ) \quad 'e'_{\mathcal{X}}(\mathcal{X}) = \begin{cases} 'e'_{\mathcal{X}}(\mathcal{X}) = 'c' & \text{if } \ell = 0 \\ ['x_{\sigma} \rightarrow 'x_{\overline{s}}' \leftarrow 'e[\overline{x} \leftarrow \overline{s}]' & \text{if } \ell \neq 0 \end{cases}
\]

\textbf{Proof.} We assume \( 'e' \in \mathcal{L}_\mathcal{V}[\mathcal{X}] \) and \( \text{persp.GP} \) (ref. \( 4.7 \) on page 13, \( 7.7 \) and \( 4.6 \) on page 14), w.r.t. \( 4.3 \) substitute \( 'e'_{\mathcal{X}}(\mathcal{X}) \) for \( 'e'_{\mathcal{M}}(\overline{u}) \). (---) \quad 'e'_{\mathcal{X}}(\mathcal{X})[\overline{s}] = 'e[\overline{x} \leftarrow \overline{s}]' \quad \text{for } \overline{s} \in \mathcal{X}_\sigma \quad \text{(note that } \mathcal{X}_\sigma \subseteq \mathcal{L}_\sigma[[]])

We shall evaluate the left side of the equation according to \( 4.6 \) Def. of interpretation and the right side due to \( 5.7 \) and utilize from the induction premise that the law to be shown already applies to the argument terms \( 'a_i' \) until \( \ell = \text{right} \) becomes evident.
Case \( m = 0 \): From \( e' \in \mathcal{L}_\gamma[X] \), by 4.3 on page \( \text{[11]} \) and the supposed \( \text{persp.GP} \) we obtain \( e' = \text{op} \in \text{SOP} \cup \{\hat{x}\} \). First consider case \( e' = \text{op} \in \{\hat{x}\} \). \( \text{left:} \) According to 5.6 (Def. \( \text{interpretation} \)) \( e' \hat{x}(\hat{s}) = p_1^\theta \hat{s} = \hat{s}_k \). \( \text{right:} \) 5.6 yields \( e[\hat{x} \leftarrow \hat{s}] = \hat{s}_k \). For both \( \text{left} \) and \( \text{right} \) \( k \) is defined to be the biggest so that \( x_k = \hat{x}' \). \( s_k \in X_{\alpha_k} \) implies \( s_k = \hat{s}_k \), hence \( e[\hat{x} \leftarrow \hat{s}] = \hat{s}_k \). \( (--) \) is evident.

Now consider the other case \( e' = \text{op} \in \text{SOP} \). \( \text{left:} \) by 5.6 \( e' \hat{x}(\hat{s}) = \text{csf}_{X_{\text{op}}}^\theta \), then \( e' \hat{x}(\hat{s}) = X_{\text{op}}^\theta \) \( \text{right:} \) by 5.6 \( e[\hat{x} \leftarrow \hat{s}] = \text{op} = \hat{e} \). By Def. \( \text{interpretation} \) \( e' \hat{x}(\hat{s}) = X_{\text{op}}^\theta \) for \( \{h, \hat{s}\} \) with \( h_1 \colon \hat{x}_\delta \rightarrow X_{\alpha_1} \), so that for arbitrary \( \bar{s} \in X_\delta \)

\[
\begin{align*}
&\text{if } r_1 = 0 \text{ then } h_1 = \left[ X_{\alpha_1}^\to \right]_{b \mapsto \hat{a}_1} \quad \text{else } h_1 = \left[ X_{\alpha_1}^\to \right]_{b \mapsto \hat{a}_1} \quad
\end{align*}
\]

\( \text{induction hypothesis yields} 
\]

\( (i) \) \( \text{if } r_1 = 0 \text{ then } h_1 = \left[ X_{\alpha_1}^\to \right]_{b \mapsto \hat{a}_1} \quad \text{else } h_1 = \left[ X_{\alpha_1}^\to \right]_{b \mapsto \hat{a}_1} \quad
\]

In order to apply 7.3 (Def. \( \hat{x} \)) we transform the equation next to label “\( \text{left:} \)” into \( \langle h_1 \hat{s} \rangle \{a^* \}, v^* \rangle \) \( e' \hat{x}(\hat{s}) = X_{\text{op}}^\theta \) and in order to distinguish already used symbols (meta variables) from corresponding ones that may denote different objects we supply the one with star as superscript. Applying 7.3 (3) in this way we obtain \( X_{\text{op}}^\theta \) \( \prod_{i=1}^m X_{\alpha_i}^\delta \rightarrow X_Y \) and there are \( a^*, v^* \) \( e'^* \) so that \( v^* = \{v_i^*\}_{i=1}^m \) \( a^* \) is \( \langle \{a_i^*\}_{i=1}^m \rangle \) \( \left[ a_i^* \right] \in \mathcal{L}_{\alpha_i} \) \( v_i^* \) \( \text{VAR}_{\beta_i} \), each \( \left[ a_i^* \right] \in \mathcal{L}_{\alpha_i} \) \( v_i^* \) \( \text{VAR}_{\beta_i} \). \( \text{combining (i)+(ii)} \)

\( \text{metavariable depends on parameter } \hat{s} \). Our \( \text{goal:} \) to deduce \( e'^* = e[\hat{x} \leftarrow \hat{s}] \). To achieve it we try to obtain the premises suited for an application of 7.4 \( \text{lemma.} \)

Combining (i)+(ii) yields

\( \text{if } r_1 = 0 \text{ then } \left[ a_i^* \left[ \hat{x} \leftarrow \hat{s} \right] \right] = \left[ a_i^* \right] \) \( \text{else } \left( \forall b \in X_{\beta_i} \right) \left[ a_i^* \left[ \hat{x} \leftarrow \hat{s} \right] \right] = \left[ a_i^* \right][v_i^* \leftarrow b] \)

Note that case \( r_1 = 0 \) can be treated like the other case, as \( \left[ a_i^* \right][v_i^* \leftarrow b] \) is stipulated by 5.6 \( \text{def.} \) 7.6 yields \( \left[ a_i [\hat{x} \leftarrow \hat{s} b] = a_i^* [v_i^* \leftarrow b] \right] \)

\( \text{by 5.6} \) \( \left[ a_i [\hat{x} \leftarrow \hat{s} b] = a_i [\hat{x} \leftarrow \hat{s} b] \right] \rightarrow \left[ a_i^* [v_i^* \leftarrow b] \right] \) hence

\( \text{(iii)} \)

\( \forall \hat{b} \in X_{\beta_i} \rightarrow \left[ a_i [\hat{x} \leftarrow \hat{s} b] = a_i^* [v_i^* \leftarrow b] \right] \)

This is the prerequisite for applying 7.4 which yields the required \( e'^* = e[\hat{x} \leftarrow \hat{s}] \).

\( \square \)

7.11 \textit{Observation} \( X^\gamma = \langle \gamma \rangle = 1_{\text{m}_3} \) \( X^\lambda = \langle \lambda \rangle = 0_{\text{m}_3} \)
7.12 Observation If $T \subseteq L_\pi$, `$\varphi' \in L_\pi$ then $T \vdash \forall \vec{x} \varphi' \iff T \vdash \varphi'$

7.13 Theorem Assume the premises as in 7.10, let $\gamma = \pi$ and `$\varphi' \in L_\pi[\vec{x}]$, then

1. $T' \vdash \varphi' \iff (\forall \vec{a} \in \mathcal{X}_\gamma) \ '\varphi'_{\vec{X}}(\vec{a}) = \forall \gamma' = 1_\mathfrak{B}$ 
   if $n = 0$

2. $T' \vdash \varphi' \iff \mathcal{X} \downarrow S' \varphi'$ 
   if $n \neq 0$

3. $T' \vdash \varphi' \iff \mathcal{X} \downarrow S' \varphi'$ 
   if $n \geq 0$

Proof. The two preceding observations and application of 7.10 and 7.6 yield this result. \[\square\]

7.14 Observation Let $T$ be a consistent theory and $T'$ its extension due to 7.1 on page 18 (extension theorem). If $\mathcal{X}$ is a model of $T'$ then the restriction $S \upharpoonright \mathcal{X}$ of $\mathcal{X}$ to the signature $S \subseteq S'$ is a model of $T$, that is: $\mathcal{X} \downarrow S' \mathcal{T} \rightarrow S \upharpoonright \mathcal{X}$

7.15 Completeness Theorem (2nd Version) A consistent $\text{Fnl}_1$ theory has a model.

$\text{Fnl}_1 \ \text{Signat} \ S \land T \subseteq L_\pi \land \text{Consistent}(T) \rightarrow (\exists \mathfrak{M} \ \text{Fnl}_1 \ Structure) \ \mathfrak{M} \downarrow S$ $\text{T}$

Proof. This is a consequence of 7.1, 7.3, 7.14 according to 6.10. \[\square\]

7.16 Completeness Theorem (Gödel) Any formula that is valid in a $\text{Fnl}_1$ theory is provable in it, that is $\text{Fnl}_1 \ \text{Signat} \ S \land T \cup \{\varphi'\} \subseteq L_\pi \land T \downarrow S \varphi' \rightarrow T \downarrow S \varphi'$

Proof. $T \downarrow S \varphi'$ iff $T \cup \{\neg \varphi'\}$ has no model

$T \downarrow S \varphi'$ iff $\neg \text{Consistent}(T \cup \{\neg \varphi'\})$

Use 7.13 with $T \cup \{\neg \varphi'\}$ in place of $T$, then the first 2 premises of the theorem imply

$\neg (\exists \mathfrak{M}) \mathfrak{M} \downarrow S \varphi' \rightarrow \neg \text{Consistent}(T \cup \{\neg \varphi'\})$ by propositional tautology

$T \downarrow S \varphi' \rightarrow T \downarrow S \varphi'$ according to the first two lines of the proof \[\square\]

Remarks. The construction of a term model could be carried out also with a theory $T'$, if the requirement of 6.9 on page 17 (or 7.11 on page 18 respectively) is weakened by rewriting $\text{Cop}_S$ into $L_\pi[\vec{x}]$ (note that $\text{Cop}_S \subseteq L_\pi[\vec{x}]$). We then say $T'$ admits examples. If we prove the validity of Hilbert’s 2nd $\varepsilon$-Theorem for Functional Logic, which claims $T'$ to be a conservative extension of $T$, if the new symbols are $\varepsilon_\gamma$ with signature `$\gamma(\gamma : \pi)$ for each $\gamma \in \text{VSRT}$ and the additional axioms are $(\varepsilon_0)$ and $(\varepsilon_2)$ as defined in [LEI69] (but actually multiplied by indexing range VSRT), then $T'$ admits examples and inherits consistency from $T$. We observe that
\((\varepsilon_2) \ \forall \ z (\varphi^x_z \leftrightarrow \psi^y_z) \rightarrow \varepsilon x.\varphi = \varepsilon y.\psi\) is accurately the equality axiom. (I2) for the symbol ‘\(\varepsilon\)’ = ‘\(\varepsilon_\xi\)’ (see p. 14 and remember that ‘\(\leftrightarrow\)’ is the same as ‘\(\pi\)’).

\((\varepsilon_0) \ \exists x \ \varphi \rightarrow \varphi[x \leftarrow \varepsilon x.\varphi]\) is a non-logical axiom, which makes \(\mathcal{T}'\) admit examples. Thus \(\mathcal{T}'\) with only one new symbol for each \(\gamma \in \text{VSrt}\) would be equally suitable for constructing a term model. Yet proving Hilbert’s 2nd \(\varepsilon\)-Theorem (ref. to [LEI69]) for this purpose is considerably more difficult and largescale than Henkin’s approach.

References

[ASS57] G. Asser. Theorie der logischen Auswahlfunktionen. Zeitschrift für mathematische Logik und Grundlagen der Mathematik, 3:30–68, 1957.
[BAR77] Jon Barwise. An Introduction to First-Order Logic. In Handbook of Mathematical Logic, volume 90 of Studies in Logic and the Foundations of Mathematics, chapter 4. The Completeness Theorem, pages 22–40. North-Holland, 1977.
[CHR93] C. Christian. Allgemeiner Korrektheits und Vollständigkeitssatz für maximal konsistente Kalküle. Sitzungsberichte der Österr. Akademie d. Wissenschaften, Mathematisch-naturwissensch. Klasse, II, Bd. 202, Heft 1-10:23–45, 1993.
[GÖD30] Kurt Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. Monatshefte für Mathematik u. Physik, 37:349–360, 1930.
[HIL23] D. Hilbert. Die logischen Grundlagen der Mathematik. Mathematische Annalen, 88:151–165, 1923.
[HIL28] D. Hilbert. Die Grundlagen der Mathematik. Abh. Math. Seminar Hamburg, 6:65–85, 1928.
[KEI70] H.J. Keisler. Logic with the quantifier “there exist uncountably many”. Ann. Math. Logic, 1:1–93, 1970.
[LEI69] A.C. Leisenring. Mathem. Logic and Hilbert’s Epsilon-Symbol. Macdonald & Co, London W.1, 1969.
[RAS56] H. Rasiowa. On the \(\varepsilon\)-theorems. Fundamenta Mathematicae, 43, 1956.
[SHO67] J.R. Shoenfield. Mathematical Logic. Addison-Wesley, Reading, Mass., 1967.