A Model of Mizar Concepts – Unification

Grzegorz Bancerek
Białystok Technical University
Poland
The University of Finance and Management
Białystok–Ełk, Poland

Summary. The aim of this paper is to develop a formal theory of Mizar linguistic concepts following the ideas from [6] and [7]. The theory presented is an abstraction from the existing implementation of the Mizar system and is devoted to the formalization of Mizar expressions. The concepts formalized here are: standarized constructor signature, arity-rich signatures, and the unification of Mizar expressions.

MML identifier: ABCMIZ.A, version: 7.11.04 4.130.1076

The notation and terminology used in this paper are introduced in the following articles: [20], [21], [12], [22], [10], [14], [13], [17], [18], [15], [1], [8], [11], [2], [3], [4], [19], [16], [5], [9], and [7]. For simplicity the abbreviation \( \mathfrak{M} = \text{MaxConstrSign} \) is introduced.

1. Preliminary

In this paper \( i, j \) denote natural numbers.
Next we state two propositions:
(1) For every pair set \( x \) holds \( x = \langle x_1, x_2 \rangle \).
(2) For every infinite set \( X \) there exist sets \( x_1, x_2 \) such that \( x_1, x_2 \in X \) and \( x_1 \neq x_2 \).

In this article we present several logical schemes. The scheme MinimalElement deals with a finite non empty set \( A \) and a binary predicate \( \mathcal{P} \), and states that:

\( ^{1}\text{Partially supported by BTU Grant W/WI/1/06 and UF&M(B) Teaching Support} \)
There exists a set \( x \) such that \( x \in A \) and for every set \( y \) such that \( y \in A \) holds not \( P[y, x] \) provided the parameters have the following properties:
- For all sets \( x, y \) such that \( x, y \in A \) and \( P[x, y] \) holds not \( P[y, x] \),
- For all sets \( x, y, z \) such that \( x, y, z \in A \) and \( P[x, y] \) and \( P[y, z] \) holds \( P[x, z] \).

The scheme \( \text{FiniteC} \) deals with a finite set \( A \) and a unary predicate \( P \), and states that:

\[ P[A] \]

provided the following condition is satisfied:
- For every subset \( A \) of \( A \) such that for every set \( B \) such that \( B \subset A \) holds \( P[B] \) holds \( P[A] \).

The scheme \( \text{Numeration} \) deals with a finite set \( A \) and a binary predicate \( P \), and states that:

There exists an one-to-one finite sequence \( s \) such that \( \text{rng} s = A \)
and for all \( i, j \) such that \( i, j \in \text{dom} s \) and \( P[s(i), s(j)] \) holds \( i < j \)

provided the parameters satisfy the following conditions:
- For all sets \( x, y \) such that \( x, y \in A \) and \( P[x, y] \) holds not \( P[y, x] \),
- For all sets \( x, y, z \) such that \( x, y, z \in A \) and \( P[x, y] \) and \( P[y, z] \) holds \( P[x, z] \).

One can prove the following two propositions:

1. For every variable \( x \) holds \( \text{varcl} \ vars(x) = vars(x) \).
2. Let \( C \) be an initialized constructor signature and \( e \) be an expression of \( C \). Then \( e \) is compound if and only if it is not true that there exists an element \( x \) of Vars such that \( e = x_C \).

2. Standardized Constructor Signature

Let us note that there exists a quasi-locus sequence which is empty.

Let \( C \) be a constructor signature. We say that \( C \) is standardized if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let \( o \) be an operation symbol of \( C \). Suppose \( o \) is constructor. Then \( o \in \text{Constructors} \) and \( o_1 = \text{the result sort of } o \) and \( \text{Card}((o_2)_1) = \text{len} \text{Arity}(o) \).

The following proposition is true

(5) Let \( C \) be a constructor signature. Suppose \( C \) is standardized. Let \( o \) be an operation symbol of \( C \). Then \( o \) is constructor if and only if \( o \in \text{Constructors} \).

Let us note that \( \mathfrak{M} \) is standardized.
Let us observe that there exists a constructor signature which is initialized, standardized, and strict.

Let \( C \) be an initialized standardized constructor signature and let \( c \) be a constructor operation symbol of \( C \). The loci of \( c \) yielding a quasi-locus sequence is defined by:

(Def. 2) The loci of \( c = (c_2)_1 \).

Let \( C \) be a constructor signature. One can verify that there exists a subsignature of \( C \) which is constructor.

Let \( C \) be an initialized constructor signature. Note that there exists a constructor subsignature of \( C \) which is initialized.

Let \( C \) be a standardized constructor signature. One can verify that every constructor subsignature of \( C \) is standardized.

One can prove the following two propositions:

(6) Let \( S_1, S_2 \) be standardized constructor signatures. Suppose the operation symbols of \( S_1 = \) the operation symbols of \( S_2 \). Then the many sorted signature of \( S_1 = \) the many sorted signature of \( S_2 \).

(7) For every constructor signature \( C \) holds \( C \) is standardized iff \( C \) is a subsignature of \( \mathcal{M} \).

Let \( C \) be an initialized constructor signature. Observe that there exists a quasi-term of \( C \) which is non compound.

Let us mention that every element of \( \text{Vars} \) is pair.

The following propositions are true:

(8) For every element \( x \) of \( \text{Vars} \) such that \( \vars(x) \) is natural holds \( \vars(x) = 0 \).

(9) \( \text{Vars} \) misses Constructors.

(10) For every element \( x \) of \( \text{Vars} \) holds \( x \neq * \) and \( x \neq \text{non} \).

(11) For every standardized constructor signature \( C \) holds \( \text{Vars} \) misses the operation symbols of \( C \).

(12) Let \( C \) be an initialized standardized constructor signature and \( e \) be an expression of \( C \). Then

(i) there exists an element \( x \) of \( \text{Vars} \) such that \( e = x_C \) and \( e(\emptyset) = \langle x, \text{term} \rangle \), or

(ii) there exists an operation symbol \( o \) of \( C \) such that \( e(\emptyset) = \langle o, \text{the carrier of } C \rangle \) but \( o \in \text{Constructors} \) or \( o = * \) or \( o = \text{non} \).

Let \( C \) be an initialized standardized constructor signature and let \( e \) be an expression of \( C \). Note that \( e(\emptyset) \) is pair.

The following propositions are true:

(13) Let \( C \) be an initialized constructor signature, \( e \) be an expression of \( C \), and \( o \) be an operation symbol of \( C \). Suppose \( e(\emptyset) = \langle o, \text{the carrier of } C \rangle \). Then \( e \) is an expression of \( C \) from the result sort of \( o \).
(14) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. Then
(i) if $e(\emptyset)_1 = \ast$, then $e$ is an expression of $\mathcal{C}$ from $\text{type}_{\mathcal{C}}$, and
(ii) if $e(\emptyset)_1 = \text{non}$, then $e$ is an expression of $\mathcal{C}$ from $\text{adj}_{\mathcal{C}}$.

(15) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. Then
(i) $e(\emptyset)_1 \in \text{Vars}$ and $e(\emptyset)_2 = \text{term}$ and $e$ is a quasi-term of $\mathcal{C}$, or
(ii) $e(\emptyset)_2$ is the carrier of $\mathcal{C}$ but $e(\emptyset)_1 \in \text{Constructors}$ and $e(\emptyset)_1 \in$ the operation symbols of $\mathcal{C}$ or $e(\emptyset)_1 = \ast$ or $e(\emptyset)_1 = \text{non}$.

(16) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. If $e(\emptyset)_1 \in \text{Constructors}$, then $e \in$ (the sorts of \text{Free}_{\mathcal{C}}(\text{Vars} \mathcal{C}))((e(\emptyset)_1)_1)$.

(17) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. Then $e(\emptyset)_1 \notin \text{Vars}$ if and only if $e(\emptyset)_1$ is an operation symbol of $\mathcal{C}$.

(18) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. If $e(\emptyset)_1 \in \text{Vars}$, then there exists an element $x$ of Vars such that $x = e(\emptyset)_1$ and $e = x_{\mathcal{C}}$.

(19) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. Suppose $e(\emptyset)_1 = \ast$. Then there exists an expression $\alpha$ of $\mathcal{C}$ from $\text{adj}_{\mathcal{C}}$ and there exists an expression $q$ of $\mathcal{C}$ from $\text{type}_{\mathcal{C}}$ such that $e = \langle \ast, 3 \rangle \text{-tree}(\alpha, q)$.

(20) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. If $e(\emptyset)_1 = \text{non}$, then there exists an expression $\alpha$ of $\mathcal{C}$ from $\text{adj}_{\mathcal{C}}$ such that $e = \langle \text{non}, 3 \rangle \text{-tree}(\alpha)$.

(21) Let $\mathcal{C}$ be an initialized standardized constructor signature and $e$ be an expression of $\mathcal{C}$. Suppose $e(\emptyset)_1 \in \text{Constructors}$. Then there exists an operation symbol $o$ of $\mathcal{C}$ such that $o = e(\emptyset)_1$ and the result sort of $o = o_1$ and $e$ is an expression of $\mathcal{C}$ from the result sort of $o$.

(22) Let $\mathcal{C}$ be an initialized standardized constructor signature and $\tau$ be a quasi-term of $\mathcal{C}$. Then $\tau$ is compound if and only if $\tau(\emptyset)_1 \in \text{Constructors}$ and $(\tau(\emptyset)_1)_1 = \text{term}$.

(23) Let $\mathcal{C}$ be an initialized standardized constructor signature and $\tau$ be an expression of $\mathcal{C}$. Then $\tau$ is a non compound quasi-term of $\mathcal{C}$ if and only if $\tau(\emptyset)_1 \in \text{Vars}$.

(24) Let $\mathcal{C}$ be an initialized standardized constructor signature and $\tau$ be an expression of $\mathcal{C}$. Then $\tau$ is a quasi-term of $\mathcal{C}$ if and only if $\tau(\emptyset)_1 \in \text{Constructors}$ and $(\tau(\emptyset)_1)_1 = \text{term}$ or $\tau(\emptyset)_1 \in \text{Vars}$.

(25) Let $\mathcal{C}$ be an initialized standardized constructor signature and $\alpha$ be an expression of $\mathcal{C}$. Then $\alpha$ is a positive quasi-adjective of $\mathcal{C}$ if and only if
$\alpha(\emptyset)_{1} \in \text{Constructors}$ and $(\alpha(\emptyset)_{1})_{1} = \text{adj}$.

(26) Let $C$ be an initialized standardized constructor signature and $\alpha$ be a quasi-adjective of $C$. Then $\alpha$ is negative if and only if $\alpha(\emptyset)_{1} = \text{non}$.

(27) Let $C$ be an initialized standardized constructor signature and $\tau$ be an expression of $C$. Then $\tau$ is a pure expression of $C$ from $\text{type}_{C}$ if and only if $\tau(\emptyset)_{1} \in \text{Constructors}$ and $(\tau(\emptyset)_{1})_{1} = \text{type}$.

3. Expressions

In the sequel $i$ is a natural number, $x$ is a variable, and $\ell$ is a quasi-locus sequence.

An expression is an expression of $M$. A valuation is a valuation of $M$. A quasi-adjective is a quasi-adjective of $M$. The subset QuasiAdjs of $\text{Free}_{M}(\text{Vars} M)$ is defined as follows:

(Def. 3) QuasiAdjs = QuasiAdjs $M$.

A quasi-term is a quasi-term of $M$. The subset QuasiTerms of $\text{Free}_{M}(\text{Vars} M)$ is defined as follows:

(Def. 4) QuasiTerms = QuasiTerms $M$.

A quasi-type is a quasi-type of $M$. The functor QuasiTypes is defined as follows:

(Def. 5) QuasiTypes = QuasiTypes $M$.

One can verify the following observations:

* QuasiAdjs is non empty,
* QuasiTerms is non empty, and
* QuasiTypes is non empty.

Modes is a non empty subset of Constructors. Then Attrs is a non empty subset of Constructors. Then Funcs is a non empty subset of Constructors.

In the sequel $C$ denotes an initialized constructor signature.

The element set-constr of Modes is defined by:

(Def. 6) set-constr = \{type, (\emptyset, 0)\}.

One can prove the following propositions:

(28) The kind of set-constr = type and the loci of set-constr = $\emptyset$ and the index of set-constr = 0.

(29) Constructors = \{type, adj, term\} $\times$ (QuasiLoci $\times$ N).

(30) $(\text{rng } \ell, i) \in \text{Vars}$ and $\ell \subset \langle(\text{rng } \ell, i)\rangle$ is a quasi-locus sequence.

(31) There exists $\ell$ such that len $\ell = i$.

(32) For every finite subset $X$ of Vars there exists $\ell$ such that $\text{rng } \ell = \text{varcl } X$.

(33) Let $X$, $o$ be sets and $p$ be a decorated tree yielding finite sequence. Given $C$ such that $X = \bigcup (\text{the sorts of } \text{Free}_{C}(\text{Vars } C))$. If $o\text{-tree}(p) \in X$, then $p$ is a finite sequence of elements of $X$. 


Let us consider $\mathcal{C}$ and let $e$ be an expression of $\mathcal{C}$. An expression of $\mathcal{C}$ is called a subexpression of $e$ if:

(Def. 7) \quad \text{It } \in \text{Subtrees}(e).

The functor $\text{constrs} e$ is defined by:

(Def. 8) \quad \text{constrs} e = \pi_1(\text{rng} e) \cap \{ o : o \text{ ranges over constructor operation symbols of } \mathcal{C} \}.

The functor $\text{main-constr} e$ is defined by:

(Def. 9) \quad \text{main-constr} e = \begin{cases} e(\emptyset)_1, & \text{if } e \text{ is compound}, \\ \emptyset, & \text{otherwise}. \end{cases}

The functor $\text{args} e$ yields a finite sequence of elements of $\text{Free}_e(\text{Vars} \mathcal{C})$ and is defined by:

(Def. 10) \quad e = e(\emptyset)-\text{tree}(\text{args} e).

Next we state three propositions:

(34) For every $\mathcal{C}$ holds every expression $e$ of $\mathcal{C}$ is a subexpression of $e$.

(35) $\text{main-constr}(x_\mathcal{C}) = \emptyset$.

(36) Let $c$ be a constructor operation symbol of $\mathcal{C}$ and $p$ be a finite sequence of elements of $\text{QuasiTerms} \mathcal{C}$. If $\text{len} p = \text{len} \text{Arity}(c)$, then $\text{main-constr}(c^\uparrow(p)) = c$.

Let us consider $\mathcal{C}$ and let $e$ be an expression of $\mathcal{C}$. We say that $e$ is constructor if and only if:

(Def. 11) \quad e \text{ is compound and } \text{main-constr} e \text{ is a constructor operation symbol of } \mathcal{C}.

Let us consider $\mathcal{C}$. Observe that every expression of $\mathcal{C}$ which is constructor is also compound.

Let us consider $\mathcal{C}$. Observe that there exists an expression of $\mathcal{C}$ which is constructor.

Let us consider $\mathcal{C}$ and let $e$ be a constructor expression of $\mathcal{C}$. One can verify that there exists a subexpression of $e$ which is constructor.

Let $S$ be a non void signature, let $X$ be a non empty yielding many sorted set indexed by $S$, and let $\tau$ be an element of $\text{Free}_S(X)$. Observe that $\text{rng} \tau$ is relation-like.

One can prove the following proposition

(37) For every constructor expression $e$ of $\mathcal{C}$ holds $\text{main-constr} e \in \text{constrs} e$.

4. Arity

For simplicity, we follow the rules: $\alpha$ is a quasi-adjective, $\tau$, $\tau_1$, $\tau_2$ are quasi-terms, $\vartheta$ is a quasi-type, and $c$ is an element of Constructors.

Let $\mathcal{C}$ be a non void signature. We say that $\mathcal{C}$ is arity-rich if and only if the condition (Def. 12) is satisfied.
(Def. 12) Let \( n \) be a natural number and \( s \) be a sort symbol of \( \mathcal{C} \). Then \( \{ o; o \text{ ranges over operation symbols of } \mathcal{C}; \text{ the result sort of } o = s \land \text{len Arity}(o) = n \} \) is infinite.

Let \( o \) be an operation symbol of \( \mathcal{C} \). We say that \( o \) is nullary if and only if:

(Def. 13) \( \text{Arity}(o) = \emptyset \).

We say that \( o \) is unary if and only if:

(Def. 14) \( \text{len Arity}(o) = 1 \).

We say that \( o \) is binary if and only if:

(Def. 15) \( \text{len Arity}(o) = 2 \).

The following proposition is true

(38) Let \( \mathcal{C} \) be a non void signature and \( o \) be an operation symbol of \( \mathcal{C} \). Then

(i) if \( o \) is nullary, then \( o \) is not unary,
(ii) if \( o \) is nullary, then \( o \) is not binary, and
(iii) if \( o \) is unary, then \( o \) is not binary.

Let \( \mathcal{C} \) be a constructor signature. Observe that \( \text{non}_{\mathcal{C}} \) is unary and \( \ast_{\mathcal{C}} \) is binary.

Let \( \mathcal{C} \) be a constructor signature. Note that every operation symbol of \( \mathcal{C} \) which is nullary is also constructor.

The following proposition is true

(39) Let \( \mathcal{C} \) be a constructor signature. Then \( \mathcal{C} \) is initialized if and only if there exists an operation symbol \( m \) of \( \text{type}_{\mathcal{C}} \) and there exists an operation symbol \( \alpha \) of \( \text{adj}_{\mathcal{C}} \) such that \( m \) is nullary and \( \alpha \) is nullary.

Let \( \mathcal{C} \) be an initialized constructor signature. One can verify that there exists an operation symbol of \( \text{type}_{\mathcal{C}} \) which is nullary and constructor and there exists an operation symbol of \( \text{adj}_{\mathcal{C}} \) which is nullary and constructor.

Let \( \mathcal{C} \) be an initialized constructor signature. Observe that there exists an operation symbol of \( \mathcal{C} \) which is nullary and constructor.

One can check that every non void signature which is arity-rich has also an operation for each sort and every constructor signature which is arity-rich is also initialized.

One can check that \( \mathcal{M} \) is arity-rich.

Let us mention that there exists a constructor signature which is arity-rich and initialized.

Let \( \mathcal{C} \) be an arity-rich constructor signature and let \( s \) be a sort symbol of \( \mathcal{C} \).

One can verify the following observations:

- there exists an operation symbol of \( s \) which is nullary and constructor,
- there exists an operation symbol of \( s \) which is unary and constructor, and
- there exists an operation symbol of \( s \) which is binary and constructor.
Let $\mathcal{C}$ be an arity-rich constructor signature. One can check that there exists an operation symbol of $\mathcal{C}$ which is unary and constructor and there exists an operation symbol of $\mathcal{C}$ which is binary and constructor.

The following proposition is true

(40) Let $o$ be a nullary operation symbol of $\mathcal{C}$. Then $(o,\text{the carrier of }\mathcal{C})$-tree($\emptyset$) is an expression of $\mathcal{C}$ from the result sort of $o$.

Let $\mathcal{C}$ be an initialized constructor signature and let $m$ be a nullary constructor operation symbol of $\text{type}_{\mathcal{C}}$. Then $m_\mathcal{C}$ is a pure expression of $\mathcal{C}$ from $\text{type}_{\mathcal{C}}$.

Let $c$ be an element of Constructors. The functor $^c\text{c}$ yielding a constructor operation symbol of $\mathfrak{M}$ is defined by:

(Def. 16) $^c\text{c} = c$.

Let $m$ be an element of Modes. Then $^m\text{c}$ is a constructor operation symbol of $\text{type}_{\mathfrak{M}}$.

Let us note that $^\text{set-constr}\text{c}$ is nullary.

We now state the proposition

(41) $\text{Arity}(^\text{set-constr}\text{c}) = \emptyset$.

The quasi-type set-type is defined by:

(Def. 17) set-type = $\emptyset_{\text{QuasiAdjs}\mathfrak{M}} * (^\text{set-constr}\text{c})_\mathfrak{M}$.

The following proposition is true

(42) $\text{adjs set-type} = \emptyset$ and the base of set-type = $(^\text{set-constr}\text{c})_\mathfrak{M}$.

Let $\ell$ be a finite sequence of elements of Vars. The functor $\text{args }\ell$ yields a finite sequence of elements of quasiTerms $\mathfrak{M}$ and is defined as follows:

(Def. 18) $\text{len args }\ell = \text{len }\ell$ and for every $i$ such that $i \in \text{dom }\ell$ holds $(\text{args }\ell)(i) = (\ell_i)_\mathfrak{M}$.

Let us consider $c$. The base expression of $c$ yields an expression and is defined as follows:

(Def. 19) The base expression of $c = (^c\text{c})^{\text{args (the loci of }c)}$.

Next we state several propositions:

(43) For every operation symbol $o$ of $\mathfrak{M}$ holds $o$ is constructor iff $o \in \text{Constructors}$.

(44) For every nullary operation symbol $m$ of $\mathfrak{M}$ holds $\text{main-constr}(m_\mathfrak{M}) = m$.

(45) For every unary constructor operation symbol $m$ of $\mathfrak{M}$ and for every $\tau$ holds $\text{main-constr}(m(\tau)) = m$.

(46) For every $\alpha$ holds $\text{main-constr}(\text{non}_{\mathfrak{M}}(\alpha)) = \text{non}$.

(47) For every binary constructor operation symbol $m$ of $\mathfrak{M}$ and for all $\tau_1, \tau_2$ holds $\text{main-constr}(m(\tau_1, \tau_2)) = m$.

(48) For every expression $q$ of $\mathfrak{M}$ from $\text{type}_{\mathfrak{M}}$ and for every $\alpha$ holds $\text{main-constr}(*_{\mathfrak{M}}(\alpha, q)) = *$. 
Let \( \vartheta \) be a quasi-type. The functor \( \text{constrs} \vartheta \) is defined by:

(Def. 20) \( \text{constrs} \vartheta = \text{constrs} (\text{the base of } \vartheta) \cup \bigcup \{ \text{constrs} \alpha : \alpha \in \text{adjs} \vartheta \} \).

The following two propositions are true:

(49) For every pure expression \( q \) of \( \mathfrak{M} \) from type \( \text{type}_M \) and for every finite subset \( A \) of QuasiAdjs \( \mathfrak{M} \) holds \( \text{constrs}(A \ast q) = \text{constrs} q \cup \bigcup \{ \text{constrs} \alpha : \alpha \in A \} \).

(50) \( \text{constrs}(\alpha \ast \vartheta) = \text{constrs} \alpha \cup \text{constrs} \vartheta \).

5. Unification

Let \( C \) be an initialized constructor signature and let \( \tau, p \) be expressions of \( C \). We say that \( \tau \) matches \( p \) if and only if:

(Def. 21) There exists a valuation \( f \) of \( C \) such that \( \tau = p[f] \).

Let us note that the predicate \( \tau \) matches \( p \) is reflexive.

The following proposition is true

(51) For all expressions \( \tau_1, \tau_2, \tau_3 \) of \( C \) such that \( \tau_1 \) matches \( \tau_2 \) and \( \tau_2 \) matches \( \tau_3 \) holds \( \tau_1 \) matches \( \tau_3 \).

Let \( C \) be an initialized constructor signature and let \( A, B \) be subsets of QuasiAdjs \( C \). We say that \( A \) matches \( B \) if and only if:

(Def. 22) There exists a valuation \( f \) of \( C \) such that \( B[f] \subseteq A \).

Let us note that the predicate \( A \) matches \( B \) is reflexive.

The following proposition is true

(52) For all subsets \( A_1, A_2, A_3 \) of QuasiAdjs \( C \) such that \( A_1 \) matches \( A_2 \) and \( A_2 \) matches \( A_3 \) holds \( A_1 \) matches \( A_3 \).

Let \( C \) be an initialized constructor signature and let \( \vartheta, P \) be quasi-types of \( C \). We say that \( \vartheta \) matches \( P \) if and only if:

(Def. 23) There exists a valuation \( f \) of \( C \) such that \( (\text{adjs} P)[f] \subseteq \text{adjs} \vartheta \) and (the base of \( P \))[\( f \) = the base of \( \vartheta \).

Let us note that the predicate \( \vartheta \) matches \( P \) is reflexive.

One can prove the following proposition

(53) For all quasi-types \( \vartheta_1, \vartheta_2, \vartheta_3 \) of \( C \) such that \( \vartheta_1 \) matches \( \vartheta_2 \) and \( \vartheta_2 \) matches \( \vartheta_3 \) holds \( \vartheta_1 \) matches \( \vartheta_3 \).

Let \( C \) be an initialized constructor signature, let \( \tau_1, \tau_2 \) be expressions of \( C \), and let \( f \) be a valuation of \( C \). We say that \( f \) unifies \( \tau_1 \) with \( \tau_2 \) if and only if:

(Def. 24) \( \tau_1[f] = \tau_2[f] \).

The following proposition is true

(54) Let \( \tau_1, \tau_2 \) be expressions of \( C \) and \( f \) be a valuation of \( C \). If \( f \) unifies \( \tau_1 \) with \( \tau_2 \), then \( f \) unifies \( \tau_2 \) with \( \tau_1 \).

Let \( C \) be an initialized constructor signature and let \( \tau_1, \tau_2 \) be expressions of \( C \). We say that \( \tau_1 \) and \( \tau_2 \) are unifiable if and only if:
(Def. 25) There exists a valuation \( f \) of \( \mathcal{C} \) such that \( f \) unifies \( \tau_1 \) with \( \tau_2 \).

Let us notice that the predicate \( \tau_1 \) and \( \tau_2 \) are unifiable is reflexive and symmetric.

Let \( \mathcal{C} \) be an initialized constructor signature and let \( \tau_1, \tau_2 \) be expressions of \( \mathcal{C} \). We say that \( \tau_1 \) and \( \tau_2 \) are weakly-unifiable if and only if:

(Def. 26) There exists an irrelevant one-to-one valuation \( g \) of \( \mathcal{C} \) such that \( \text{Var} \tau_2 \subseteq \text{dom} g \) and \( \tau_1 \) and \( \tau_2[g] \) are unifiable.

Let us note that the predicate \( \tau_1 \) and \( \tau_2 \) are weakly-unifiable is reflexive.

We now state the proposition

(55) For all expressions \( \tau_1, \tau_2 \) of \( \mathcal{C} \) such that \( \tau_1 \) and \( \tau_2 \) are unifiable holds \( \tau_1 \) and \( \tau_2 \) are weakly-unifiable.

Let \( \mathcal{C} \) be an initialized constructor signature and let \( \tau, \tau_1, \tau_2 \) be expressions of \( \mathcal{C} \). We say that \( \tau \) is a unification of \( \tau_1 \) and \( \tau_2 \) if and only if:

(Def. 27) There exists a valuation \( f \) of \( \mathcal{C} \) such that \( f \) unifies \( \tau_1 \) with \( \tau_2 \) and \( \tau = \tau_1[f] \).

We now state two propositions:

(56) For all expressions \( \tau_1, \tau_2, \tau \) of \( \mathcal{C} \) such that \( \tau \) is a unification of \( \tau_1 \) and \( \tau_2 \) holds \( \tau \) is a unification of \( \tau_2 \) and \( \tau_1 \).

(57) For all expressions \( \tau_1, \tau_2, \tau \) of \( \mathcal{C} \) such that \( \tau \) is a unification of \( \tau_1 \) and \( \tau_2 \) holds \( \tau \) matches \( \tau_1 \) and \( \tau \) matches \( \tau_2 \).

Let \( \mathcal{C} \) be an initialized constructor signature and let \( \tau, \tau_1, \tau_2 \) be expressions of \( \mathcal{C} \). We say that \( \tau \) is a general-unification of \( \tau_1 \) and \( \tau_2 \) if and only if the conditions (Def. 28) are satisfied.

(Def. 28)(i) \( \tau \) is a unification of \( \tau_1 \) and \( \tau_2 \), and

(ii) for every expression \( u \) of \( \mathcal{C} \) such that \( u \) is a unification of \( \tau_1 \) and \( \tau_2 \) holds \( u \) matches \( \tau \).

6. Type Distribution

The following three propositions are true:

(58) Let \( n \) be a natural number and \( s \) be a sort symbol of \( \mathcal{M} \). Then there exists a constructor operation symbol \( m \) of \( s \) such that \( \text{len Arity}(m) = n \).

(59) Let given \( \ell, s \) be a sort symbol of \( \mathcal{M} \), and \( m \) be a constructor operation symbol of \( s \). If \( \text{len Arity}(m) = \text{len} \ell \) then \( \text{Var}(m'(\text{args} \ell)) = \text{rng} \ell \).

(60) Let \( X \) be a finite subset of Vars. Suppose \( \text{varcl} X = X \). Let \( s \) be a sort symbol of \( \mathcal{M} \). Then there exists a constructor operation symbol \( m \) of \( s \) and there exists a finite sequence \( p \) of elements of QuasiTerms \( \mathcal{M} \) such that \( \text{len} p = \text{len Arity}(m) \) and \( \text{vars}(m'(p)) = X \).

Let \( d \) be a partial function from Vars to QuasiTypes. We say that \( d \) is even if and only if:
(Def. 29) For all $x$, $\vartheta$ such that $x \in \text{dom } d$ and $\vartheta = d(x)$ holds $\text{vars}(\vartheta) = \text{vars}(x)$.

Let $\ell$ be a quasi-locus sequence. A partial function from Vars to QuasiTypes is said to be a type-distribution for $\ell$ if:

(Def. 30) $\text{dom it} = \text{rng } \ell$ and it is even.

We now state the proposition

(61) For every empty quasi-locus sequence $\ell$ holds $\emptyset$ is a type-distribution for $\ell$.

References


Received November 20, 2009