

A Model of Mizar Concepts – Unification

Grzegorz Bancerek¹
Białystok Technical University
Poland
The University of Finance and Management
Białystok–Elk, 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: standardized 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 \mathcal{A} and a binary predicate \mathcal{P} , and states that:

¹Partially supported by BTU Grant W/WI/1/06 and UF&M(B) Teaching Support

There exists a set x such that $x \in \mathcal{A}$ and for every set y such that $y \in \mathcal{A}$ holds not $\mathcal{P}[y, x]$

provided the parameters have the following properties:

- For all sets x, y such that $x, y \in \mathcal{A}$ and $\mathcal{P}[x, y]$ holds not $\mathcal{P}[y, x]$, and
- For all sets x, y, z such that $x, y, z \in \mathcal{A}$ and $\mathcal{P}[x, y]$ and $\mathcal{P}[y, z]$ holds $\mathcal{P}[x, z]$.

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

$\mathcal{P}[\mathcal{A}]$

provided the following condition is satisfied:

- For every subset A of \mathcal{A} such that for every set B such that $B \subset A$ holds $\mathcal{P}[B]$ holds $\mathcal{P}[A]$.

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

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

provided the parameters satisfy the following conditions:

- For all sets x, y such that $x, y \in \mathcal{A}$ and $\mathcal{P}[x, y]$ holds not $\mathcal{P}[y, x]$, and
- For all sets x, y, z such that $x, y, z \in \mathcal{A}$ and $\mathcal{P}[x, y]$ and $\mathcal{P}[y, z]$ holds $\mathcal{P}[x, z]$.

One can prove the following two propositions:

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

2. STANDARDIZED CONSTRUCTOR SIGNATURE

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

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

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

The following proposition is true

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

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

Let \mathfrak{C} be a constructor signature. One can verify that there exists a subsignature of \mathfrak{C} which is constructor.

Let \mathfrak{C} be an initialized constructor signature. Note that there exists a constructor subsignature of \mathfrak{C} which is initialized.

Let \mathfrak{C} be a standardized constructor signature. One can verify that every constructor subsignature of \mathfrak{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 \mathfrak{C} holds \mathfrak{C} is standardized iff \mathfrak{C} is a subsignature of \mathfrak{M} .

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

Let us mention that every element of Vars is pair.

The following propositions are true:

- (8) For every element x of Vars such that $\text{vars}(x)$ is natural holds $\text{vars}(x) = 0$.
- (9) Vars misses Constructors.
- (10) For every element x of Vars holds $x \neq *$ and $x \neq \mathbf{non}$.
- (11) For every standardized constructor signature \mathfrak{C} holds Vars misses the operation symbols of \mathfrak{C} .
- (12) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . Then
 - (i) there exists an element x of Vars such that $e = x_{\mathfrak{C}}$ and $e(\emptyset) = \langle x, \mathbf{term} \rangle$, or
 - (ii) there exists an operation symbol o of \mathfrak{C} such that $e(\emptyset) = \langle o, \text{the carrier of } \mathfrak{C} \rangle$ but $o \in \text{Constructors}$ or $o = *$ or $o = \mathbf{non}$.

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

The following propositions are true:

- (13) Let \mathfrak{C} be an initialized constructor signature, e be an expression of \mathfrak{C} , and o be an operation symbol of \mathfrak{C} . Suppose $e(\emptyset) = \langle o, \text{the carrier of } \mathfrak{C} \rangle$. Then e is an expression of \mathfrak{C} from the result sort of o .

- (14) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . Then
- (i) if $e(\emptyset)_1 = *$, then e is an expression of \mathfrak{C} from $\mathbf{type}_{\mathfrak{C}}$, and
 - (ii) if $e(\emptyset)_1 = \mathbf{non}$, then e is an expression of \mathfrak{C} from $\mathbf{adj}_{\mathfrak{C}}$.
- (15) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . Then
- (i) $e(\emptyset)_1 \in \mathbf{Vars}$ and $e(\emptyset)_2 = \mathbf{term}$ and e is a quasi-term of \mathfrak{C} , or
 - (ii) $e(\emptyset)_2 =$ the carrier of \mathfrak{C} but $e(\emptyset)_1 \in \mathbf{Constructors}$ and $e(\emptyset)_1 \in$ the operation symbols of \mathfrak{C} or $e(\emptyset)_1 = *$ or $e(\emptyset)_1 = \mathbf{non}$.
- (16) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . If $e(\emptyset)_1 \in \mathbf{Constructors}$, then $e \in$ (the sorts of $\mathbf{Free}_{\mathfrak{C}}(\mathbf{Vars} \mathfrak{C})((e(\emptyset)_1)_1)$).
- (17) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . Then $e(\emptyset)_1 \notin \mathbf{Vars}$ if and only if $e(\emptyset)_1$ is an operation symbol of \mathfrak{C} .
- (18) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . If $e(\emptyset)_1 \in \mathbf{Vars}$, then there exists an element x of \mathbf{Vars} such that $x = e(\emptyset)_1$ and $e = x_{\mathfrak{C}}$.
- (19) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . Suppose $e(\emptyset)_1 = *$. Then there exists an expression α of \mathfrak{C} from $\mathbf{adj}_{\mathfrak{C}}$ and there exists an expression q of \mathfrak{C} from $\mathbf{type}_{\mathfrak{C}}$ such that $e = \langle *, 3 \rangle\text{-tree}(\alpha, q)$.
- (20) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . If $e(\emptyset)_1 = \mathbf{non}$, then there exists an expression α of \mathfrak{C} from $\mathbf{adj}_{\mathfrak{C}}$ such that $e = \langle \mathbf{non}, 3 \rangle\text{-tree}(\alpha)$.
- (21) Let \mathfrak{C} be an initialized standardized constructor signature and e be an expression of \mathfrak{C} . Suppose $e(\emptyset)_1 \in \mathbf{Constructors}$. Then there exists an operation symbol o of \mathfrak{C} such that $o = e(\emptyset)_1$ and the result sort of $o = o_1$ and e is an expression of \mathfrak{C} from the result sort of o .
- (22) Let \mathfrak{C} be an initialized standardized constructor signature and τ be a quasi-term of \mathfrak{C} . Then τ is compound if and only if $\tau(\emptyset)_1 \in \mathbf{Constructors}$ and $(\tau(\emptyset)_1)_1 = \mathbf{term}$.
- (23) Let \mathfrak{C} be an initialized standardized constructor signature and τ be an expression of \mathfrak{C} . Then τ is a non compound quasi-term of \mathfrak{C} if and only if $\tau(\emptyset)_1 \in \mathbf{Vars}$.
- (24) Let \mathfrak{C} be an initialized standardized constructor signature and τ be an expression of \mathfrak{C} . Then τ is a quasi-term of \mathfrak{C} if and only if $\tau(\emptyset)_1 \in \mathbf{Constructors}$ and $(\tau(\emptyset)_1)_1 = \mathbf{term}$ or $\tau(\emptyset)_1 \in \mathbf{Vars}$.
- (25) Let \mathfrak{C} be an initialized standardized constructor signature and α be an expression of \mathfrak{C} . Then α is a positive quasi-adjective of \mathfrak{C} if and only if

- $\alpha(\emptyset)_1 \in \text{Constructors}$ and $(\alpha(\emptyset)_1)_1 = \mathbf{adj}$.
- (26) Let \mathfrak{C} be an initialized standardized constructor signature and α be a quasi-adjective of \mathfrak{C} . Then α is negative if and only if $\alpha(\emptyset)_1 = \mathbf{non}$.
- (27) Let \mathfrak{C} be an initialized standardized constructor signature and τ be an expression of \mathfrak{C} . Then τ is a pure expression of \mathfrak{C} from $\mathbf{type}_{\mathfrak{C}}$ if and only if $\tau(\emptyset)_1 \in \text{Constructors}$ and $(\tau(\emptyset)_1)_1 = \mathbf{type}$.

3. EXPRESSIONS

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

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

(Def. 3) $\text{QuasiAdjs} = \text{QuasiAdjs } \mathfrak{M}$.

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

(Def. 4) $\text{QuasiTerms} = \text{QuasiTerms } \mathfrak{M}$.

A quasi-type is a quasi-type of \mathfrak{M} . The functor QuasiTypes is defined as follows:

(Def. 5) $\text{QuasiTypes} = \text{QuasiTypes } \mathfrak{M}$.

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 \mathfrak{C} denotes an initialized constructor signature.

The element set-constr of Modes is defined by:

(Def. 6) $\text{set-constr} = \langle \mathbf{type}, \langle \emptyset, 0 \rangle \rangle$.

One can prove the following propositions:

- (28) The kind of $\text{set-constr} = \mathbf{type}$ and the loci of $\text{set-constr} = \emptyset$ and the index of $\text{set-constr} = 0$.
- (29) $\text{Constructors} = \{\mathbf{type}, \mathbf{adj}, \mathbf{term}\} \times (\text{QuasiLoci} \times \mathbb{N})$.
- (30) $\langle \text{rng } \ell, i \rangle \in \text{Vars}$ and $\ell \hat{\ } \langle \langle \text{rng } \ell, i \rangle \rangle$ is a quasi-locus sequence.
- (31) There exists ℓ such that $\text{len } \ell = i$.
- (32) For every finite subset X of Vars there exists ℓ such that $\text{rng } \ell = \text{varcl } X$.
- (33) Let X, o be sets and p be a decorated tree yielding finite sequence. Given \mathfrak{C} such that $X = \bigcup (\text{the sorts of } \text{Free}_{\mathfrak{C}}(\text{Vars } \mathfrak{C}))$. If $o\text{-tree}(p) \in X$, then p is a finite sequence of elements of X .

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

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

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

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

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

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

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

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

Next we state three propositions:

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

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

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

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

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

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

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

Let us consider \mathfrak{C} and let e be a constructor expression of \mathfrak{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 τ 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 \mathfrak{C} holds $\text{main-constr } e \in \text{constrs } e$.

4. ARITY

For simplicity, we follow the rules: α is a quasi-adjective, τ, τ_1, τ_2 are quasi-terms, ϑ is a quasi-type, and c is an element of Constructors.

Let \mathfrak{C} be a non void signature. We say that \mathfrak{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 \mathfrak{C} . Then $\{o; o \text{ ranges over operation symbols of } \mathfrak{C}: \text{the result sort of } o = s \wedge \text{len Arity}(o) = n\}$ is infinite.

Let o be an operation symbol of \mathfrak{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 \mathfrak{C} be a non void signature and o be an operation symbol of \mathfrak{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 \mathfrak{C} be a constructor signature. Observe that $\mathbf{non}_{\mathfrak{C}}$ is unary and $*_{\mathfrak{C}}$ is binary.

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

The following proposition is true

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

Let \mathfrak{C} be an initialized constructor signature. One can verify that there exists an operation symbol of $\mathbf{type}_{\mathfrak{C}}$ which is nullary and constructor and there exists an operation symbol of $\mathbf{adj}_{\mathfrak{C}}$ which is nullary and constructor.

Let \mathfrak{C} be an initialized constructor signature. Observe that there exists an operation symbol of \mathfrak{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 \mathfrak{M} is arity-rich.

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

Let \mathfrak{C} be an arity-rich constructor signature and let s be a sort symbol of \mathfrak{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 \mathfrak{C} be an arity-rich constructor signature. One can check that there exists an operation symbol of \mathfrak{C} which is unary and constructor and there exists an operation symbol of \mathfrak{C} which is binary and constructor.

The following proposition is true

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

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

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

- (Def. 16) ${}^{\textcircled{a}}c = c$.

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

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

We now state the proposition

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

The quasi-type set-type is defined by:

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

The following proposition is true

- (42) $\text{adjs set-type} = \emptyset$ and the base of $\text{set-type} = ({}^{\textcircled{a}}\text{set-constr})_t$.

Let ℓ 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 $c = ({}^{\textcircled{a}}c)^{\neg}(\text{args}(\text{the loci of } c))$.

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 $\text{main-constr}(m_t) = m$.
- (45) For every unary constructor operation symbol m of \mathfrak{M} and for every τ holds $\text{main-constr}(m(\tau)) = m$.
- (46) For every α holds $\text{main-constr}(\mathbf{non}_{\mathfrak{M}}(\alpha)) = \mathbf{non}$.
- (47) For every binary constructor operation symbol m of \mathfrak{M} and for all τ_1, τ_2 holds $\text{main-constr}(m(\tau_1, \tau_2)) = m$.
- (48) For every expression q of \mathfrak{M} from $\mathbf{type}_{\mathfrak{M}}$ and for every α holds $\text{main-constr}(*_{\mathfrak{M}}(\alpha, q)) = *$.

Let ϑ be a quasi-type. The functor $\text{constrs } \vartheta$ is defined by:

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

The following two propositions are true:

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

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

5. UNIFICATION

Let \mathfrak{C} be an initialized constructor signature and let τ, p be expressions of \mathfrak{C} . We say that τ matches p if and only if:

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

Let us note that the predicate τ matches p is reflexive.

The following proposition is true

(51) For all expressions τ_1, τ_2, τ_3 of \mathfrak{C} such that τ_1 matches τ_2 and τ_2 matches τ_3 holds τ_1 matches τ_3 .

Let \mathfrak{C} be an initialized constructor signature and let A, B be subsets of $\text{QuasiAdjs } \mathfrak{C}$. We say that A matches B if and only if:

(Def. 22) There exists a valuation f of \mathfrak{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 $\text{QuasiAdjs } \mathfrak{C}$ such that A_1 matches A_2 and A_2 matches A_3 holds A_1 matches A_3 .

Let \mathfrak{C} be an initialized constructor signature and let ϑ, P be quasi-types of \mathfrak{C} . We say that ϑ matches P if and only if:

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

Let us note that the predicate ϑ matches P is reflexive.

One can prove the following proposition

(53) For all quasi-types $\vartheta_1, \vartheta_2, \vartheta_3$ of \mathfrak{C} such that ϑ_1 matches ϑ_2 and ϑ_2 matches ϑ_3 holds ϑ_1 matches ϑ_3 .

Let \mathfrak{C} be an initialized constructor signature, let τ_1, τ_2 be expressions of \mathfrak{C} , and let f be a valuation of \mathfrak{C} . We say that f unifies τ_1 with τ_2 if and only if:

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

The following proposition is true

(54) Let τ_1, τ_2 be expressions of \mathfrak{C} and f be a valuation of \mathfrak{C} . If f unifies τ_1 with τ_2 , then f unifies τ_2 with τ_1 .

Let \mathfrak{C} be an initialized constructor signature and let τ_1, τ_2 be expressions of \mathfrak{C} . We say that τ_1 and τ_2 are unifiable if and only if:

(Def. 25) There exists a valuation f of \mathfrak{C} such that f unifies τ_1 with τ_2 .

Let us notice that the predicate τ_1 and τ_2 are unifiable is reflexive and symmetric.

Let \mathfrak{C} be an initialized constructor signature and let τ_1, τ_2 be expressions of \mathfrak{C} . We say that τ_1 and τ_2 are weakly-unifiable if and only if:

(Def. 26) There exists an irrelevant one-to-one valuation g of \mathfrak{C} such that $\text{Var } \tau_2 \subseteq \text{dom } g$ and τ_1 and $\tau_2[g]$ are unifiable.

Let us note that the predicate τ_1 and τ_2 are weakly-unifiable is reflexive.

We now state the proposition

(55) For all expressions τ_1, τ_2 of \mathfrak{C} such that τ_1 and τ_2 are unifiable holds τ_1 and τ_2 are weakly-unifiable.

Let \mathfrak{C} be an initialized constructor signature and let τ, τ_1, τ_2 be expressions of \mathfrak{C} . We say that τ is a unification of τ_1 and τ_2 if and only if:

(Def. 27) There exists a valuation f of \mathfrak{C} such that f unifies τ_1 with τ_2 and $\tau = \tau_1[f]$.

We now state two propositions:

(56) For all expressions τ_1, τ_2, τ of \mathfrak{C} such that τ is a unification of τ_1 and τ_2 holds τ is a unification of τ_2 and τ_1 .

(57) For all expressions τ_1, τ_2, τ of \mathfrak{C} such that τ is a unification of τ_1 and τ_2 holds τ matches τ_1 and τ matches τ_2 .

Let \mathfrak{C} be an initialized constructor signature and let τ, τ_1, τ_2 be expressions of \mathfrak{C} . We say that τ is a general-unification of τ_1 and τ_2 if and only if the conditions (Def. 28) are satisfied.

(Def. 28)(i) τ is a unification of τ_1 and τ_2 , and
(ii) for every expression u of \mathfrak{C} such that u is a unification of τ_1 and τ_2 holds u matches τ .

6. TYPE DISTRIBUTION

The following three propositions are true:

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

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

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

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, ϑ such that $x \in \text{dom } d$ and $\vartheta = d(x)$ holds $\text{vars}(\vartheta) = \text{vars}(x)$.

Let ℓ be a quasi-locus sequence. A partial function from Vars to QuasiTypes is said to be a type-distribution for ℓ 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 ℓ holds \emptyset is a type-distribution for ℓ .

REFERENCES

- [1] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(3):589–593, 1990.
- [2] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(4):547–552, 1991.
- [3] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(1):77–82, 1993.
- [4] Grzegorz Bancerek. Subtrees. *Formalized Mathematics*, 5(2):185–190, 1996.
- [5] Grzegorz Bancerek. Institution of many sorted algebras. Part I: Signature reduct of an algebra. *Formalized Mathematics*, 6(2):279–287, 1997.
- [6] Grzegorz Bancerek. On the structure of Mizar types. In Herman Geuvers and Fairouz Kamareddine, editors, *Electronic Notes in Theoretical Computer Science*, volume 85. Elsevier, 2003.
- [7] Grzegorz Bancerek. Towards the construction of a model of Mizar concepts. *Formalized Mathematics*, 16(2):207–230, 2008, doi:10.2478/v10037-008-0027-x.
- [8] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(1):107–114, 1990.
- [9] Grzegorz Bancerek and Artur Korniłowicz. Yet another construction of free algebra. *Formalized Mathematics*, 9(4):779–785, 2001.
- [10] Grzegorz Bancerek and Yatsuka Nakamura. Full adder circuit. Part I. *Formalized Mathematics*, 5(3):367–380, 1996.
- [11] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(3):529–536, 1990.
- [12] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(1):55–65, 1990.
- [13] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(1):153–164, 1990.
- [14] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(2):357–367, 1990.
- [15] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(1):165–167, 1990.
- [16] Beata Perkowska. Free many sorted universal algebra. *Formalized Mathematics*, 5(1):67–74, 1996.
- [17] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(2):329–334, 1990.
- [18] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(1):97–105, 1990.
- [19] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(1):37–42, 1996.
- [20] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(1):67–71, 1990.
- [21] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(1):73–83, 1990.
- [22] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(1):181–186, 1990.

Received November 20, 2009