FORMALIZED MATHEMATICS
Vol. **16**, No. **2**, Pages 207–230, **2008**

Towards the Construction of a Model of Mizar Concepts[1]

Grzegorz Bancerek
Białystok Technical University
Poland

Summary. The aim of this paper is to develop a formal theory of Mizar linguistic concepts following the ideas from [14] and [13]. The theory here presented is an abstract of the existing implementation of the Mizar system and is devoted to the formalization of Mizar expressions. The base idea behind the formalization is dependence on variables which is determined by variable-dependence (variables may depend on other variables). The dependence constitutes a Galois connection between opposite poset of dependence-closed set of variables and the sup-semilattice of widening of Mizar types (smooth type widening).

In the paper the concepts strictly connected with Mizar expressions are formalized. Among them are quasi-loci, quasi-terms, quasi-adjectives, and quasi-types. The structural induction and operation of substitution are also introduced. The prefix *quasi* is used to indicate that some rules of construction of Mizar expressions may not be fulfilled. For example, variables, quasi-loci, and quasi-terms have no assigned types and, in result, there is no possibility to conduct type-checking of arguments. The other gaps concern inconsistent and out-of-context clusters of adjectives in types. Those rules are required in the Mizar *identification* process. However, the expression appearing in later processes of Mizar checker may not satisfy the rules. So, introduced apparatus is enough and adequate to describe data structures and algorithms from the Mizar checker (like *equational classes*).

MML identifier: `ABCMIZ_1`, version: `7.9.01 4.101.1015`

The notation and terminology used in this paper are introduced in the following papers: [24], [41], [33], [30], [42], [20], [43], [39], [23], [17], [34], [21], [22], [32], [2],

[1]This work has been partially supported by the Białystok Technical University grant W/WI/1/06.

ISSN 1426–2630(p), 1898-9934(e)

[38], [35], [26], [1], [4], [15], [19], [28], [3], [7], [8], [9], [36], [25], [5], [40], [6], [11], [12], [27], [18], [37], [29], [31], [10], and [16].

1. Variables

We adopt the following convention: i is a natural number, j is an element of $\mathbb{N}$, and X, Y, x, y, z are sets.

One can prove the following propositions:

(1) For every function f holds $f(x) \subseteq \bigcup f$.

(2) For every function f such that $\bigcup f = \emptyset$ holds $f(x) = \emptyset$.

(3) For every function f and for all sets x, y such that $f = \langle x, y \rangle$ holds $x = y$.

(4) $(\mathrm{id}_X)^\circ Y \subseteq Y$.

(5) Let Σ be a non void signature and X be a non-empty many sorted set indexed by the carrier of Σ. Then every term of Σ over X is non pair.

Let Σ be a non void signature and let X be a non empty yielding many sorted set indexed by the carrier of Σ. Observe that every element of $\mathrm{Free}_\Sigma(X)$ is non pair.

We now state the proposition

(6) For all sets x, y, z such that x, $y \in \{z\}^*$ and $\overline{\overline{x}} = \overline{\overline{y}}$ holds $x = y$.

Let us note that $\emptyset$ is decorated tree yielding.

Let Σ be a non void signature and let $\mathfrak{A}$ be an algebra over Σ. A subset of $\mathfrak{A}$ is a subset of $\bigcup$(the sorts of $\mathfrak{A}$). A finite sequence of elements of $\mathfrak{A}$ is a finite sequence of elements of $\bigcup$(the sorts of $\mathfrak{A}$).

Let Σ be a non void signature and let X be a non empty yielding many sorted set indexed by Σ. Note that every finite sequence of elements of $\mathrm{Free}_\Sigma(X)$ is decorated tree yielding.

Next we state the proposition

(7) Let Σ be a non void signature, X be a non empty yielding many sorted set indexed by the carrier of Σ, and τ be an element of $\mathrm{Free}_\Sigma(X)$. Then

(i) there exists a sort symbol s of Σ and there exists a set v such that $\tau =$ the root tree of $\langle v, s \rangle$ and $v \in X(s)$, or

(ii) there exists an operation symbol o of Σ and there exists a finite sequence p of elements of $\mathrm{Free}_\Sigma(X)$ such that $\tau = \langle o,$ the carrier of $\Sigma \rangle$-tree(p) and $\mathrm{len}\, p = \mathrm{len}\, \mathrm{Arity}(o)$ and p is decorated tree yielding and an argument sequence of $\mathrm{Sym}(o, X \cup ((\text{the carrier of } \Sigma) \longmapsto \{0\}))$.

Let A be a set. The functor $\mathrm{varcl}\, A$ is defined by the conditions (Def. 1).

(Def. 1)(i) $A \subseteq \mathrm{varcl}\, A$,

(ii) for all x, y such that $\langle x, y \rangle \in \mathrm{varcl}\, A$ holds $x \subseteq \mathrm{varcl}\, A$, and

(iii) for every set B such that $A \subseteq B$ and for all x, y such that $\langle x, y\rangle \in B$ holds $x \subseteq B$ holds $\operatorname{varcl} A \subseteq B$.

Let us observe that the functor $\operatorname{varcl} A$ is projective.

We now state three propositions:

(8) $\operatorname{varcl} \emptyset = \emptyset$.

(9) For all sets Y, Z such that $Y \subseteq Z$ holds $\operatorname{varcl} Y \subseteq \operatorname{varcl} Z$.

(10) For every set Z holds $\operatorname{varcl} \bigcup Z = \bigcup\{\operatorname{varcl} z : z \text{ ranges over elements of } Z\}$.

The scheme *Sch14* deals with a set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding a set, and a unary predicate $\mathcal{P}$, and states that:

$\operatorname{varcl} \bigcup\{\mathcal{F}(z); z \text{ ranges over elements of } \mathcal{A} : \mathcal{P}[z]\} = \bigcup\{\operatorname{varcl} \mathcal{F}(z); z$ ranges over elements of $\mathcal{A} : \mathcal{P}[z]\}$

for all values of the parameters.

Next we state three propositions:

(11) $\operatorname{varcl}(X \cup Y) = \operatorname{varcl} X \cup \operatorname{varcl} Y$.

(12) For every non empty set Z such that for every element z of Z holds $\operatorname{varcl} z = z$ holds $\operatorname{varcl} \bigcap Z = \bigcap Z$.

(13) $\operatorname{varcl}(\operatorname{varcl} X \cap \operatorname{varcl} Y) = \operatorname{varcl} X \cap \operatorname{varcl} Y$.

Let Z be an empty set. Observe that $\operatorname{varcl} Z$ is empty.

The functor Vars is defined by the condition (Def. 2).

(Def. 2) There exists a many sorted set V indexed by $\mathbb{N}$ such that

(i) $\operatorname{Vars} = \bigcup V$,

(ii) $V(0) = \{\langle \emptyset, i\rangle : i \text{ ranges over elements of } \mathbb{N}\}$, and

(iii) for every natural number n holds $V(n+1) = \{\langle \operatorname{varcl} Z, j\rangle; Z \text{ ranges over subsets of } V(n), j \text{ ranges over elements of } \mathbb{N}: Z \text{ is finite}\}$.

Next we state a number of propositions:

(14) Let V be a many sorted set indexed by $\mathbb{N}$. Suppose that

(i) $V(0) = \{\langle \emptyset, i\rangle : i \text{ ranges over elements of } \mathbb{N}\}$, and

(ii) for every natural number n holds $V(n+1) = \{\langle \operatorname{varcl} Z, j\rangle; Z \text{ ranges over subsets of } V(n), j \text{ ranges over elements of } \mathbb{N}: Z \text{ is finite}\}$.

Let i, j be elements of $\mathbb{N}$. If $i \leq j$, then $V(i) \subseteq V(j)$.

(15) Let V be a many sorted set indexed by $\mathbb{N}$. Suppose that

(i) $V(0) = \{\langle \emptyset, i\rangle : i \text{ ranges over elements of } \mathbb{N}\}$, and

(ii) for every natural number n holds $V(n+1) = \{\langle \operatorname{varcl} Z, j\rangle; Z \text{ ranges over subsets of } V(n), j \text{ ranges over elements of } \mathbb{N}: Z \text{ is finite}\}$.

Let Z be a finite subset of Vars. Then there exists an element i of $\mathbb{N}$ such that $Z \subseteq V(i)$.

(16) $\{\langle \emptyset, i\rangle : i \text{ ranges over elements of } \mathbb{N}\} \subseteq \operatorname{Vars}$.

(17) For every finite subset Z of Vars and for every natural number i holds $\langle \operatorname{varcl} Z, i\rangle \in \operatorname{Vars}$.

(18) Vars = $\{\langle \operatorname{varcl} Z, j\rangle; Z$ ranges over subsets of Vars, j ranges over elements of $\mathbb{N}$: Z is finite$\}$.

(19) $\operatorname{varcl} \mathrm{Vars} = \mathrm{Vars}$.

(20) For every X such that $\mathrm{rk}(X)$ is finite holds X is finite.

(21) $\mathrm{rk}(\operatorname{varcl} X) = \mathrm{rk}(X)$.

(22) For every finite subset X of $\mathbf{R}_\omega$ holds $X \in \mathbf{R}_\omega$.

(23) $\mathrm{Vars} \subseteq \mathbf{R}_\omega$.

(24) For every finite subset Z of Vars holds $\operatorname{varcl} Z$ is a finite subset of Vars.

One can verify that Vars is non empty.

A variable is an element of Vars.

Let x be a variable. Observe that $x_{\mathbf{1}}$ is finite.

Let x be a variable. We introduce $\mathrm{vars}(x)$ as a synonym of $x_{\mathbf{1}}$.

Let x be a variable. Then $\mathrm{vars}(x)$ is a subset of Vars.

The following propositions are true:

(25) $\langle \emptyset, i\rangle \in \mathrm{Vars}$.

(26) For every subset Z of Vars holds $\operatorname{varcl}\{\langle \operatorname{varcl} Z, j\rangle\} = \operatorname{varcl} Z \cup \{\langle \operatorname{varcl} Z, j\rangle\}$.

(27) For every variable x holds $\operatorname{varcl}\{x\} = \mathrm{vars}(x) \cup \{x\}$.

(28) For every variable x holds $\langle \mathrm{vars}(x) \cup \{x\}, i\rangle \in \mathrm{Vars}$.

2. Quasi-loci

Let R be a binary relation and let X be a set. We introduce $R \operatorname{dom} X$ as a synonym of $R{\restriction}X$.

The set QuasiLoci of finite sequences of Vars is defined by the condition (Def. 3).

(Def. 3) Let p be a finite sequence of elements of Vars. Then $p \in \mathrm{QuasiLoci}$ if and only if the following conditions are satisfied:

(i) p is one-to-one, and

(ii) for every i such that $i \in \operatorname{dom} p$ holds $p(i)_{\mathbf{1}} \subseteq \mathrm{rng}(p \operatorname{dom} i)$.

One can prove the following proposition

(29) $\varepsilon_{\mathrm{Vars}} \in \mathrm{QuasiLoci}$.

Let us observe that QuasiLoci is non empty.

A quasi-locus sequence is an element of QuasiLoci.

One can verify that every quasi-locus sequence is one-to-one.

Next we state several propositions:

(30) Let l be an one-to-one finite sequence of elements of Vars. Then l is a quasi-locus sequence if and only if for every natural number i and for every variable x such that $i \in \operatorname{dom} l$ and $x = l(i)$ and for every variable y such

that $y \in \mathrm{vars}(x)$ there exists a natural number j such that $j \in \mathrm{dom}\, l$ and $j < i$ and $y = l(j)$.

(31) Let l be a quasi-locus sequence and x be a variable. Then $l \frown \langle x \rangle$ is a quasi-locus sequence if and only if $x \notin \mathrm{rng}\, l$ and $\mathrm{vars}(x) \subseteq \mathrm{rng}\, l$.

(32) Let p_1, p_2 be finite sequences. Suppose $p_1 \frown p_2$ is a quasi-locus sequence. Then p_1 is a quasi-locus sequence and p_2 is a finite sequence of elements of Vars.

(33) For every quasi-locus sequence l holds $\mathrm{varcl}\, \mathrm{rng}\, l = \mathrm{rng}\, l$.

(34) For every variable x holds $\langle x \rangle$ is a quasi-locus sequence iff $\mathrm{vars}(x) = \emptyset$.

(35) For all variables x, y holds $\langle x, y \rangle$ is a quasi-locus sequence iff $\mathrm{vars}(x) = \emptyset$ and $x \neq y$ and $\mathrm{vars}(y) \subseteq \{x\}$.

(36) Let x, y, z be variables. Then $\langle x, y, z \rangle$ is a quasi-locus sequence if and only if $\mathrm{vars}(x) = \emptyset$ and $x \neq y$ and $\mathrm{vars}(y) \subseteq \{x\}$ and $x \neq z$ and $y \neq z$ and $\mathrm{vars}(z) \subseteq \{x, y\}$.

Let l be a quasi-locus sequence. Then l^{-1} is a partial function from Vars to $\mathbb{N}$.

3. Mizar Constructor Signature

The functor **type** is defined by:

(Def. 4) $\mathbf{type} = 0$.

The functor **adj** is defined by:

(Def. 5) $\mathbf{adj} = 1$.

The functor **term** is defined as follows:

(Def. 6) $\mathbf{term} = 2$.

The functor $*$ is defined by:

(Def. 7) $* = 0$.

The functor **non** is defined as follows:

(Def. 8) $\mathbf{non} = 1$.

Let $\mathfrak{C}$ be a signature. We say that $\mathfrak{C}$ is constructor if and only if the conditions (Def. 9) are satisfied.

(Def. 9) The carrier of $\mathfrak{C} = \{\mathbf{type}, \mathbf{adj}, \mathbf{term}\}$ and $\{*, \mathbf{non}\} \subseteq$ the operation symbols of $\mathfrak{C}$ and (the arity of $\mathfrak{C}$)$(*) = \langle \mathbf{adj}, \mathbf{type} \rangle$ and (the arity of $\mathfrak{C}$)$(\mathbf{non}) = \langle \mathbf{adj} \rangle$ and (the result sort of $\mathfrak{C}$)$(*) = \mathbf{type}$ and (the result sort of $\mathfrak{C}$)$(\mathbf{non}) = \mathbf{adj}$ and for every element o of the operation symbols of $\mathfrak{C}$ such that $o \neq *$ and $o \neq \mathbf{non}$ holds (the arity of $\mathfrak{C}$)$(o) \in \{\mathbf{term}\}^*$.

Let us note that every signature which is constructor is also non empty and non void.

The strict signature MinConstrSign is defined by:

(Def. 10) MinConstrSign is constructor and the operation symbols of MinConstrSign = $\{*, \mathbf{non}\}$.

Let us observe that MinConstrSign is constructor.

Let us observe that there exists a signature which is constructor and strict.

Let $\mathfrak{C}$ be a constructor signature and let o be an operation symbol of $\mathfrak{C}$. We say that o is constructor if and only if:

(Def. 11) $o \neq *$ and $o \neq \mathbf{non}$.

We now state the proposition

(37) Let Σ be a constructor signature and o be an operation symbol of Σ. If o is constructor, then $\mathrm{Arity}(o) = \mathrm{len}\,\mathrm{Arity}(o) \mapsto \mathbf{term}$.

Let $\mathfrak{C}$ be a non empty non void signature. We say that $\mathfrak{C}$ is initialized if and only if the condition (Def. 12) is satisfied.

(Def. 12) There exist operation symbols m, α of $\mathfrak{C}$ such that the result sort of m = **type** and $\mathrm{Arity}(m) = \emptyset$ and the result sort of α = **adj** and $\mathrm{Arity}(\alpha) = \emptyset$.

Let $\mathfrak{C}$ be a constructor signature. The functor $\mathbf{type}_{\mathfrak{C}}$ is a sort symbol of $\mathfrak{C}$ and is defined by:

(Def. 13) $\mathbf{type}_{\mathfrak{C}} = \mathbf{type}$.

The functor $\mathbf{adj}_{\mathfrak{C}}$ is a sort symbol of $\mathfrak{C}$ and is defined as follows:

(Def. 14) $\mathbf{adj}_{\mathfrak{C}} = \mathbf{adj}$.

The functor $\mathbf{term}_{\mathfrak{C}}$ is a sort symbol of $\mathfrak{C}$ and is defined by:

(Def. 15) $\mathbf{term}_{\mathfrak{C}} = \mathbf{term}$.

The functor $\mathbf{non}_{\mathfrak{C}}$ yielding an operation symbol of $\mathfrak{C}$ is defined as follows:

(Def. 16) $\mathbf{non}_{\mathfrak{C}} = \mathbf{non}$.

The functor $*_{\mathfrak{C}}$ yielding an operation symbol of $\mathfrak{C}$ is defined as follows:

(Def. 17) $*_{\mathfrak{C}} = *$.

We now state the proposition

(38) Let $\mathfrak{C}$ be a constructor signature. Then $\mathrm{Arity}(\mathbf{non}_{\mathfrak{C}}) = \langle \mathbf{adj}_{\mathfrak{C}} \rangle$ and the result sort of $\mathbf{non}_{\mathfrak{C}} = \mathbf{adj}_{\mathfrak{C}}$ and $\mathrm{Arity}(*_{\mathfrak{C}}) = \langle \mathbf{adj}_{\mathfrak{C}}, \mathbf{type}_{\mathfrak{C}} \rangle$ and the result sort of $*_{\mathfrak{C}} = \mathbf{type}_{\mathfrak{C}}$.

The functor Modes is defined as follows:

(Def. 18) Modes = $\{\mathbf{type}\} \times (\mathrm{QuasiLoci} \times \mathbb{N})$.

The functor Attrs is defined as follows:

(Def. 19) Attrs = $\{\mathbf{adj}\} \times (\mathrm{QuasiLoci} \times \mathbb{N})$.

The functor Funcs is defined by:

(Def. 20) Funcs = $\{\mathbf{term}\} \times (\mathrm{QuasiLoci} \times \mathbb{N})$.

One can verify the following observations:

* Modes is non empty,
* Attrs is non empty, and

∗ Funcs is non empty.

The non empty set Constructors is defined by:

(Def. 21) Constructors = Modes ∪ Attrs ∪ Funcs .

Next we state the proposition

(39) $\{*, \mathbf{non}\}$ misses Constructors.

Let x be an element of QuasiLoci $\times\mathbb{N}$. Then $x_{\mathbf{1}}$ is a quasi-locus sequence. Then $x_{\mathbf{2}}$ is an element of $\mathbb{N}$.

Let c be an element of Constructors. We introduce the kind of c as a synonym of $c_{\mathbf{1}}$.

Let c be an element of Constructors. Then the kind of c is an element of $\{\mathbf{type}, \mathbf{adj}, \mathbf{term}\}$. Then $c_{\mathbf{2}}$ is an element of QuasiLoci $\times\mathbb{N}$.

Let c be an element of Constructors. The loci of c yields a quasi-locus sequence and is defined as follows:

(Def. 22) The loci of $c = (c_{\mathbf{2}})_{\mathbf{1}}$.

The index of c yielding a natural number is defined as follows:

(Def. 23) The index of $c = (c_{\mathbf{2}})_{\mathbf{2}}$.

We now state the proposition

(40) Let c be an element of Constructors. Then

(i) the kind of $c = \mathbf{type}$ iff $c \in$ Modes,

(ii) the kind of $c = \mathbf{adj}$ iff $c \in$ Attrs, and

(iii) the kind of $c = \mathbf{term}$ iff $c \in$ Funcs .

The strict constructor signature MaxConstrSign is defined by the conditions (Def. 24).

(Def. 24)(i) The operation symbols of MaxConstrSign $= \{*, \mathbf{non}\} \cup$ Constructors, and

(ii) for every operation symbol o of MaxConstrSign such that o is constructor holds (the result sort of MaxConstrSign)$(o) = o_{\mathbf{1}}$ and $\overline{\overline{\text{(the arity of MaxConstrSign)}(o)}} = \overline{\overline{(o_{\mathbf{2}})_{\mathbf{1}}}}$.

Let us note that MinConstrSign is non initialized and MaxConstrSign is initialized.

Let us observe that there exists a constructor signature which is initialized and strict.

Let $\mathfrak{C}$ be an initialized constructor signature. One can check that there exists an operation symbol of $\mathfrak{C}$ which is constructor.

4. Mizar Expressions

Let $\mathfrak{C}$ be a constructor signature. The functor Vars $\mathfrak{C}$ yielding a many sorted set indexed by the carrier of $\mathfrak{C}$ is defined as follows:

(Def. 25) $(\mathrm{Vars}\,\mathfrak{C})(\mathbf{type}) = \emptyset$ and $(\mathrm{Vars}\,\mathfrak{C})(\mathbf{adj}) = \emptyset$ and $(\mathrm{Vars}\,\mathfrak{C})(\mathbf{term}) = \mathrm{Vars}$.

Let $\mathfrak{C}$ be a constructor signature. Note that $\mathrm{Vars}\,\mathfrak{C}$ is non empty yielding.

Let $\mathfrak{C}$ be an initialized constructor signature. Observe that $\mathrm{Free}_{\mathfrak{C}}(\mathrm{Vars}\,\mathfrak{C})$ is non-empty.

Let Σ be a non void signature, let X be a non empty yielding many sorted set indexed by the carrier of Σ, and let τ be an element of $\mathrm{Free}_{\Sigma}(X)$. We say that τ is ground if and only if:

(Def. 26) $\bigcup \mathrm{Var}_{\Sigma}\,\tau = \emptyset$.

We say that τ is compound if and only if:

(Def. 27) $\tau(\emptyset) \in$ (the operation symbols of Σ) $\times$ {the carrier of Σ}.

In the sequel $\mathfrak{C}$ denotes an initialized constructor signature, s denotes a sort symbol of $\mathfrak{C}$, o denotes an operation symbol of $\mathfrak{C}$, and c denotes a constructor operation symbol of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$. An expression of $\mathfrak{C}$ is an element of $\mathrm{Free}_{\mathfrak{C}}(\mathrm{Vars}\,\mathfrak{C})$.

Let us consider $\mathfrak{C}$, s. An expression of $\mathfrak{C}$ is called an expression of $\mathfrak{C}$ from s if:

(Def. 28) It $\in$ (the sorts of $\mathrm{Free}_{\mathfrak{C}}(\mathrm{Vars}\,\mathfrak{C})$)$(s)$.

Next we state the proposition

(41) z is an expression of $\mathfrak{C}$ from s iff $z \in$ (the sorts of $\mathrm{Free}_{\mathfrak{C}}(\mathrm{Vars}\,\mathfrak{C})$)$(s)$.

Let us consider $\mathfrak{C}$ and let us consider c. Let us assume that $\mathrm{len}\,\mathrm{Arity}(c) = 0$. The functor c_{t} yielding an expression of $\mathfrak{C}$ is defined by:

(Def. 29) $c_{\mathrm{t}} = \langle c$, the carrier of $\mathfrak{C}\rangle$-tree$(\emptyset)$.

The following proposition is true

(42) Let given o. Suppose $\mathrm{len}\,\mathrm{Arity}(o) = 1$. Let α be an expression of $\mathfrak{C}$. Given s such that $s = \mathrm{Arity}(o)(1)$ and α is an expression of $\mathfrak{C}$ from s. Then $\langle o$, the carrier of $\mathfrak{C}\rangle$-tree$(\langle\alpha\rangle)$ is an expression of $\mathfrak{C}$ from the result sort of o.

Let us consider $\mathfrak{C}$, o. Let us assume that $\mathrm{len}\,\mathrm{Arity}(o) = 1$. Let η be an expression of $\mathfrak{C}$. Let us assume that there exists a sort symbol s of $\mathfrak{C}$ such that $s = \mathrm{Arity}(o)(1)$ and η is an expression of $\mathfrak{C}$ from s. The functor $o(\eta)$ yielding an expression of $\mathfrak{C}$ is defined by:

(Def. 30) $o(\eta) = \langle o$, the carrier of $\mathfrak{C}\rangle$-tree$(\langle\eta\rangle)$.

In the sequel α, β are expressions of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$.

One can prove the following two propositions:

(43) $\mathbf{non}_{\mathfrak{C}}(\alpha)$ is an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ and $\mathbf{non}_{\mathfrak{C}}(\alpha) = \langle\,\mathbf{non}$, the carrier of $\mathfrak{C}\rangle$-tree$(\langle\alpha\rangle)$.

(44) If $\mathbf{non}_{\mathfrak{C}}(\alpha) = \mathbf{non}_{\mathfrak{C}}(\beta)$, then $\alpha = \beta$.

Let us consider $\mathfrak{C}$, α. Observe that $\mathbf{non}_{\mathfrak{C}}(\alpha)$ is compound.

Let us consider $\mathfrak{C}$. Note that there exists an expression of $\mathfrak{C}$ which is compound.

Next we state the proposition

(45) Let given o. Suppose $\operatorname{len}\operatorname{Arity}(o) = 2$. Let α, β be expressions of $\mathfrak{C}$. Given sort symbols s_1, s_2 of $\mathfrak{C}$ such that $s_1 = \operatorname{Arity}(o)(1)$ and $s_2 = \operatorname{Arity}(o)(2)$ and α is an expression of $\mathfrak{C}$ from s_1 and β is an expression of $\mathfrak{C}$ from s_2. Then $\langle o$, the carrier of $\mathfrak{C}\rangle$-tree$(\langle \alpha, \beta\rangle)$ is an expression of $\mathfrak{C}$ from the result sort of o.

Let us consider $\mathfrak{C}$, o. Let us assume that $\operatorname{len}\operatorname{Arity}(o) = 2$. Let η_1, η_2 be expressions of $\mathfrak{C}$. Let us assume that there exist sort symbols s_1, s_2 of $\mathfrak{C}$ such that $s_1 = \operatorname{Arity}(o)(1)$ and $s_2 = \operatorname{Arity}(o)(2)$ and η_1 is an expression of $\mathfrak{C}$ from s_1 and η_2 is an expression of $\mathfrak{C}$ from s_2. The functor $o(\eta_1, \eta_2)$ yielding an expression of $\mathfrak{C}$ is defined as follows:

(Def. 31) $o(\eta_1, \eta_2) = \langle o$, the carrier of $\mathfrak{C}\rangle$-tree$(\langle \eta_1, \eta_2\rangle)$.

In the sequel τ, τ_1, τ_2 are expressions of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$.

We now state two propositions:

(46) $*_{\mathfrak{C}}(\alpha, \tau)$ is an expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ and $*_{\mathfrak{C}}(\alpha, \tau) = \langle *$, the carrier of $\mathfrak{C}\rangle$-tree$(\langle \alpha, \tau\rangle)$.

(47) If $*_{\mathfrak{C}}(\alpha, \tau_1) = *_{\mathfrak{C}}(\beta, \tau_2)$, then $\alpha = \beta$ and $\tau_1 = \tau_2$.

Let us consider $\mathfrak{C}$, α, τ. One can check that $*_{\mathfrak{C}}(\alpha, \tau)$ is compound.

Let Σ be a non void signature and let s be a sort symbol of Σ. Let us assume that there exists an operation symbol o of Σ such that the result sort of $o = s$. An operation symbol of Σ is said to be an operation symbol of s if:

(Def. 32) The result sort of it $= s$.

Let $\mathfrak{C}$ be a constructor signature. Then $\mathbf{non}_{\mathfrak{C}}$ is an operation symbol of $\mathbf{adj}_{\mathfrak{C}}$. Then $*_{\mathfrak{C}}$ is an operation symbol of $\mathbf{type}_{\mathfrak{C}}$.

The following proposition is true

(48) Let s_1, s_2 be sort symbols of $\mathfrak{C}$. Suppose $s_1 \neq s_2$. Let τ_1 be an expression of $\mathfrak{C}$ from s_1 and τ_2 be an expression of $\mathfrak{C}$ from s_2. Then $\tau_1 \neq \tau_2$.

5. Quasi-terms

Let us consider $\mathfrak{C}$. The functor QuasiTerms $\mathfrak{C}$ yields a subset of $\operatorname{Free}_{\mathfrak{C}}(\operatorname{Vars}\mathfrak{C})$ and is defined as follows:

(Def. 33) QuasiTerms $\mathfrak{C} = (\text{the sorts of } \operatorname{Free}_{\mathfrak{C}}(\operatorname{Vars}\mathfrak{C}))(\mathbf{term}_{\mathfrak{C}})$.

Let us consider $\mathfrak{C}$. One can check that QuasiTerms $\mathfrak{C}$ is non empty and constituted of decorated trees.

Let us consider $\mathfrak{C}$. A quasi-term of $\mathfrak{C}$ is an expression of $\mathfrak{C}$ from $\mathbf{term}_{\mathfrak{C}}$.

We now state the proposition

(49) z is a quasi-term of $\mathfrak{C}$ iff $z \in$ QuasiTerms $\mathfrak{C}$.

Let x be a variable and let us consider $\mathfrak{C}$. The functor $x_{\mathfrak{C}}$ yields a quasi-term of $\mathfrak{C}$ and is defined by:

(Def. 34) $x_{\mathfrak{C}}$ = the root tree of $\langle x, \mathbf{term} \rangle$.

One can prove the following proposition

(50) For all variables x_1, x_2 and for all initialized constructor signatures $\mathfrak{C}_1$, $\mathfrak{C}_2$ such that $(x_1)_{\mathfrak{C}_1} = (x_2)_{\mathfrak{C}_2}$ holds $x_1 = x_2$.

Let x be a variable and let us consider $\mathfrak{C}$. One can check that $x_{\mathfrak{C}}$ is non compound.

We now state the proposition

(51) Let p be a decorated tree yielding finite sequence. Then $\langle c$, the carrier of $\mathfrak{C}\rangle$-tree(p) is an expression of $\mathfrak{C}$ if and only if $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$ and $p \in (\operatorname{QuasiTerms} \mathfrak{C})^*$.

In the sequel p is a finite sequence of elements of $\operatorname{QuasiTerms} \mathfrak{C}$.

Let us consider $\mathfrak{C}$, c and let us consider p. Let us assume that $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$. The functor $\vec{c}(p)$ yields a compound expression of $\mathfrak{C}$ and is defined as follows:

(Def. 35) $\vec{c}(p) = \langle c$, the carrier of $\mathfrak{C}\rangle$-tree(p).

Next we state several propositions:

(52) If $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$, then $\vec{c}(p)$ is an expression of $\mathfrak{C}$ from the result sort of c.

(53) Let η be an expression of $\mathfrak{C}$. Then

(i) there exists a variable x such that $\eta = x_{\mathfrak{C}}$, or

(ii) there exists a constructor operation symbol c of $\mathfrak{C}$ and there exists a finite sequence p of elements of $\operatorname{QuasiTerms} \mathfrak{C}$ such that $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$ and $\eta = \vec{c}(p)$, or

(iii) there exists an expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ such that $\eta = \mathbf{non}_{\mathfrak{C}}(\alpha)$, or

(iv) there exists an expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ and there exists an expression τ of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ such that $\eta = *_{\mathfrak{C}}(\alpha, \tau)$.

(54) If $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$, then $\vec{c}(p) \neq \mathbf{non}_{\mathfrak{C}}(\alpha)$.

(55) If $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$, then $\vec{c}(p) \neq *_{\mathfrak{C}}(\alpha, \tau)$.

(56) $\mathbf{non}_{\mathfrak{C}}(\alpha) \neq *_{\mathfrak{C}}(\beta, \tau)$.

In the sequel η is an expression of $\mathfrak{C}$.

Next we state two propositions:

(57) If $\eta(\emptyset) = \langle \mathbf{non}$, the carrier of $\mathfrak{C}\rangle$, then there exists α such that $\eta = \mathbf{non}_{\mathfrak{C}}(\alpha)$.

(58) If $\eta(\emptyset) = \langle *$, the carrier of $\mathfrak{C}\rangle$, then there exist α, τ such that $\eta = *_{\mathfrak{C}}(\alpha, \tau)$.

6. Quasi-adjectives

In the sequel α, α' denote expressions of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$.

Let us consider $\mathfrak{C}$, α. The functor $\operatorname{non}\alpha$ yields an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ and is defined by:

(Def. 36) $\operatorname{non}\alpha = \begin{cases} \alpha\restriction\langle 0\rangle, & \text{if there exists } \alpha' \text{ such that } \alpha = \mathbf{non}_{\mathfrak{C}}(\alpha'), \\ \mathbf{non}_{\mathfrak{C}}(\alpha), & \text{otherwise.} \end{cases}$

Let us consider $\mathfrak{C}$, α. We say that α is positive if and only if:

(Def. 37) It is not true that there exists α' such that $\alpha = \mathbf{non}_{\mathfrak{C}}(\alpha')$.

Let us consider $\mathfrak{C}$. Note that there exists an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ which is positive.

Next we state the proposition

(59) For every positive expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ holds $\operatorname{non}\alpha = \mathbf{non}_{\mathfrak{C}}(\alpha)$.

Let us consider $\mathfrak{C}$, α. We say that α is negative if and only if:

(Def. 38) There exists α' such that α' is positive and $\alpha = \mathbf{non}_{\mathfrak{C}}(\alpha')$.

Let us consider $\mathfrak{C}$ and let α be a positive expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$. Note that $\operatorname{non}\alpha$ is negative and non positive.

Let us consider $\mathfrak{C}$. One can check that there exists an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ which is negative and non positive.

Next we state three propositions:

(60) For every non positive expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ there exists an expression α' of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ such that $\alpha = \mathbf{non}_{\mathfrak{C}}(\alpha')$ and $\operatorname{non}\alpha = \alpha'$.

(61) Let α be a negative expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$. Then there exists a positive expression α' of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ such that $\alpha = \mathbf{non}_{\mathfrak{C}}(\alpha')$ and $\operatorname{non}\alpha = \alpha'$.

(62) For every non positive expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ holds $\mathbf{non}_{\mathfrak{C}}(\operatorname{non}\alpha) = \alpha$.

Let us consider $\mathfrak{C}$ and let α be a negative expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$. Note that $\operatorname{non}\alpha$ is positive.

Let us consider $\mathfrak{C}$, α. We say that α is regular if and only if:

(Def. 39) α is positive or negative.

Let us consider $\mathfrak{C}$. Observe that every expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ which is positive is also regular and non negative and every expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ which is negative is also regular and non positive.

Let us consider $\mathfrak{C}$. Note that there exists an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ which is regular.

Let us consider $\mathfrak{C}$. The functor $\operatorname{QuasiAdjs}\mathfrak{C}$ yields a subset of $\operatorname{Free}_{\mathfrak{C}}(\operatorname{Vars}\mathfrak{C})$ and is defined as follows:

(Def. 40) $\operatorname{QuasiAdjs}\mathfrak{C} = \{\alpha : \alpha \text{ is regular}\}$.

Let us consider $\mathfrak{C}$. Note that QuasiAdjs $\mathfrak{C}$ is non empty and constituted of decorated trees.

Let us consider $\mathfrak{C}$. A quasi-adjective of $\mathfrak{C}$ is a regular expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$.

Next we state two propositions:

(63) z is a quasi-adjective of $\mathfrak{C}$ iff $z \in$ QuasiAdjs $\mathfrak{C}$.

(64) z is a quasi-adjective of $\mathfrak{C}$ if and only if z is a positive expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ or a negative expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$.

Let us consider $\mathfrak{C}$. Note that every quasi-adjective of $\mathfrak{C}$ which is non positive is also negative and every quasi-adjective of $\mathfrak{C}$ which is non negative is also positive.

Let us consider $\mathfrak{C}$. Note that there exists a quasi-adjective of $\mathfrak{C}$ which is positive and there exists a quasi-adjective of $\mathfrak{C}$ which is negative.

The following propositions are true:

(65) Let α be a positive quasi-adjective of $\mathfrak{C}$. Then there exists a constructor operation symbol v of $\mathfrak{C}$ such that the result sort of $v = \mathbf{adj}_{\mathfrak{C}}$ and there exists p such that $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(v)$ and $\alpha = v^{\frown}(p)$.

(66) Let v be a constructor operation symbol of $\mathfrak{C}$. Suppose the result sort of $v = \mathbf{adj}_{\mathfrak{C}}$ and $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(v)$. Then $v^{\frown}(p)$ is a positive quasi-adjective of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$ and let α be a quasi-adjective of $\mathfrak{C}$. One can check that $\operatorname{non} \alpha$ is regular.

We now state three propositions:

(67) For every quasi-adjective α of $\mathfrak{C}$ holds $\operatorname{non} \operatorname{non} \alpha = \alpha$.

(68) For all quasi-adjectives α_1, α_2 of $\mathfrak{C}$ such that $\operatorname{non} \alpha_1 = \operatorname{non} \alpha_2$ holds $\alpha_1 = \alpha_2$.

(69) For every quasi-adjective α of $\mathfrak{C}$ holds $\operatorname{non} \alpha \neq \alpha$.

7. Quasi-types

Let us consider $\mathfrak{C}$ and let ϑ be an expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$. We say that ϑ is pure if and only if:

(Def. 41) It is not true that there exist α, τ such that $\vartheta = *_{\mathfrak{C}}(\alpha, \tau)$.

The following propositions are true:

(70) Let m be an operation symbol of $\mathfrak{C}$. Suppose the result sort of $m = \mathbf{type}$ and $\operatorname{Arity}(m) = \emptyset$. Then there exists τ such that $\tau =$ the root tree of $\langle m,$ the carrier of $\mathfrak{C}\rangle$ and τ is pure.

(71) Let v be an operation symbol of $\mathfrak{C}$. Suppose the result sort of $v = \mathbf{adj}$ and $\operatorname{Arity}(v) = \emptyset$. Then there exists α such that $\alpha =$ the root tree of $\langle v,$ the carrier of $\mathfrak{C}\rangle$ and α is positive.

Let us consider $\mathfrak{C}$. Note that there exists an expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ which is pure.

In the sequel ϑ denotes a pure expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ and A denotes a finite subset of QuasiAdjs $\mathfrak{C}$.

Let us consider $\mathfrak{C}$. The functor QuasiTypes $\mathfrak{C}$ is defined as follows:

(Def. 42) QuasiTypes $\mathfrak{C} = \{\langle A, \tau\rangle : \tau \text{ is pure}\}$.

Let us consider $\mathfrak{C}$. Note that QuasiTypes $\mathfrak{C}$ is non empty.

Let us consider $\mathfrak{C}$. Quasi-type of $\mathfrak{C}$ is defined by:

(Def. 43) It $\in$ QuasiTypes $\mathfrak{C}$.

The following two propositions are true:

(72) z is a quasi-type of $\mathfrak{C}$ iff there exist A, ϑ such that $z = \langle A, \vartheta\rangle$.

(73) $\langle x, y\rangle$ is a quasi-type of $\mathfrak{C}$ if and only if x is a finite subset of QuasiAdjs $\mathfrak{C}$ and y is a pure expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$.

In the sequel θ is a quasi-type of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$. Observe that every quasi-type of $\mathfrak{C}$ is pair.

Next we state four propositions:

(74) There exists a constructor operation symbol m of $\mathfrak{C}$ such that the result sort of $m = \mathbf{type}_{\mathfrak{C}}$ and there exists p such that $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(m)$ and $\vartheta = m^{\neg}(p)$.

(75) Let m be a constructor operation symbol of $\mathfrak{C}$. Suppose the result sort of $m = \mathbf{type}_{\mathfrak{C}}$ and $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(m)$. Then $m^{\neg}(p)$ is a pure expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$.

(76) QuasiTerms $\mathfrak{C}$ misses QuasiAdjs $\mathfrak{C}$ and QuasiTerms $\mathfrak{C}$ misses QuasiTypes $\mathfrak{C}$ and QuasiTypes $\mathfrak{C}$ misses QuasiAdjs $\mathfrak{C}$.

(77) Let η be a set. Then

(i) if η is a quasi-term of $\mathfrak{C}$, then η is not a quasi-adjective of $\mathfrak{C}$,

(ii) if η is a quasi-term of $\mathfrak{C}$, then η is not a quasi-type of $\mathfrak{C}$, and

(iii) if η is a quasi-type of $\mathfrak{C}$, then η is not a quasi-adjective of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$, A, ϑ. We introduce $A * \vartheta$ as a synonym of $\langle A, \vartheta\rangle$.

Let us consider $\mathfrak{C}$, A, ϑ. Then $A * \vartheta$ is a quasi-type of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$, θ. Note that $\theta_{\mathbf{1}}$ is finite.

Let us consider $\mathfrak{C}$, θ. We introduce adjs θ as a synonym of $\theta_{\mathbf{1}}$. We introduce the base of θ as a synonym of $\theta_{\mathbf{2}}$.

Let us consider $\mathfrak{C}$, θ. Then adjs θ is a subset of QuasiAdjs $\mathfrak{C}$. Then the base of θ is a pure expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$.

One can prove the following propositions:

(78) $\operatorname{adjs}(A * \vartheta) = A$ and the base of $A * \vartheta = \vartheta$.

(79) Let A_1, A_2 be finite subsets of QuasiAdjs $\mathfrak{C}$ and ϑ_1, ϑ_2 be pure expressions of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$. If $A_1 * \vartheta_1 = A_2 * \vartheta_2$, then $A_1 = A_2$ and $\vartheta_1 = \vartheta_2$.

(80) $\theta = \operatorname{adjs} \theta *$ the base of θ.

(81) For all quasi-types θ_1, θ_2 of $\mathfrak{C}$ such that $\operatorname{adjs}\theta_1 = \operatorname{adjs}\theta_2$ and the base of θ_1 = the base of θ_2 holds $\theta_1 = \theta_2$.

Let us consider $\mathfrak{C}$, θ and let α be a quasi-adjective of $\mathfrak{C}$. The functor $\alpha * \theta$ yields a quasi-type of $\mathfrak{C}$ and is defined by:

(Def. 44) $\alpha * \theta = \langle \{\alpha\} \cup \operatorname{adjs}\theta$, the base of $\theta \rangle$.

We now state three propositions:

(82) For every quasi-adjective α of $\mathfrak{C}$ holds $\operatorname{adjs}(\alpha * \theta) = \{\alpha\} \cup \operatorname{adjs}\theta$ and the base of $\alpha * \theta$ = the base of θ.

(83) For every quasi-adjective α of $\mathfrak{C}$ holds $\alpha * (\alpha * \theta) = \alpha * \theta$.

(84) For all quasi-adjectives α, β of $\mathfrak{C}$ holds $\alpha * (\beta * \theta) = \beta * (\alpha * \theta)$.

8. Variables in Quasi-types

Let Σ be a non void signature, let s be a sort symbol of Σ, let X be a non-empty many sorted set indexed by the carrier of Σ, and let τ be a term of Σ over X. Note that $(\operatorname{Var}\tau)(s)$ is finite.

Let Σ be a non void signature, let s be a sort symbol of Σ, let X be a non empty yielding many sorted set indexed by the carrier of Σ, and let τ be an element of $\operatorname{Free}_\Sigma(X)$. Observe that $(\operatorname{Var}_\Sigma \tau)(s)$ is finite.

Let Σ be a non void signature, let X be a non empty yielding many sorted set indexed by the carrier of Σ, and let s be a sort symbol of Σ. The functor vars_s^X yielding a function from $\bigcup$(the sorts of $\operatorname{Free}_\Sigma(X)$) into $2^{X(s)}$ is defined by:

(Def. 45) For every element τ of $\operatorname{Free}_\Sigma(X)$ holds $(\operatorname{vars}_s^X)(\tau) = (\operatorname{Var}_\Sigma \tau)(s)$.

Let $\mathfrak{C}$ be an initialized constructor signature and let η be an expression of $\mathfrak{C}$. The functor $\operatorname{Var}\eta$ yielding a subset of Vars is defined by:

(Def. 46) $\operatorname{Var}\eta = (\operatorname{Var}_{\mathfrak{C}}\eta)(\mathbf{term}_{\mathfrak{C}})$.

Let us consider $\mathfrak{C}$, η. Note that $\operatorname{Var}\eta$ is finite.

Let us consider $\mathfrak{C}$, η. The functor $\operatorname{vars}(\eta)$ yielding a finite subset of Vars is defined as follows:

(Def. 47) $\operatorname{vars}(\eta) = \operatorname{varcl}\operatorname{Var}\eta$.

The following propositions are true:

(85) $\operatorname{varcl}\operatorname{vars}(\eta) = \operatorname{vars}(\eta)$.

(86) For every variable x holds $\operatorname{Var}(x_{\mathfrak{C}}) = \{x\}$.

(87) For every variable x holds $\operatorname{vars}(x_{\mathfrak{C}}) = \{x\} \cup \operatorname{vars}(x)$.

(88) Let p be a decorated tree yielding finite sequence. Suppose $\eta = \langle c$, the carrier of $\mathfrak{C}\rangle$-tree(p). Then $\operatorname{Var}\eta = \bigcup\{\operatorname{Var}\tau; \tau$ ranges over quasi-terms of $\mathfrak{C}$: $\tau \in \operatorname{rng} p\}$.

(89) Let p be a decorated tree yielding finite sequence. Suppose $\eta = \langle c$, the carrier of $\mathfrak{C}\rangle$-tree(p). Then $\operatorname{vars}(\eta) = \bigcup\{\operatorname{vars}(\tau); \tau$ ranges over quasi-terms of $\mathfrak{C}$: $\tau \in \operatorname{rng} p\}$.

(90) If $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$, then $\operatorname{Var}(c^\frown(p)) = \bigcup\{\operatorname{Var}\tau; \tau$ ranges over quasi-terms of $\mathfrak{C}$: $\tau \in \operatorname{rng} p\}$.

(91) If $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$, then $\operatorname{vars}(c^\frown(p)) = \bigcup\{\operatorname{vars}(\tau); \tau$ ranges over quasi-terms of $\mathfrak{C}$: $\tau \in \operatorname{rng} p\}$.

(92) For every many sorted signature Σ and for every set o holds $\operatorname{Var}_\Sigma(\langle o$, the carrier of $\Sigma\rangle$-tree$(\emptyset)) = \mathbf{0}_{\text{the carrier of } \Sigma}$.

(93) Let Σ be a many sorted signature, o be a set, and τ be a decorated tree. Then $\operatorname{Var}_\Sigma(\langle o$, the carrier of $\Sigma\rangle$-tree$(\langle\tau\rangle)) = \operatorname{Var}_\Sigma \tau$.

(94) $\operatorname{Var}(\mathbf{non}_{\mathfrak{C}}(\alpha)) = \operatorname{Var}\alpha$.

(95) $\operatorname{vars}(\mathbf{non}_{\mathfrak{C}}(\alpha)) = \operatorname{vars}(\alpha)$.

(96) Let Σ be a many sorted signature, o be a set, and τ_1, τ_2 be decorated trees. Then $\operatorname{Var}_\Sigma(\langle o$, the carrier of $\Sigma\rangle$-tree$(\langle\tau_1, \tau_2\rangle)) = \operatorname{Var}_\Sigma \tau_1 \cup \operatorname{Var}_\Sigma \tau_2$.

(97) $\operatorname{Var}(*_{\mathfrak{C}}(\alpha, \tau)) = \operatorname{Var}\alpha \cup \operatorname{Var}\tau$.

(98) $\operatorname{vars}(*_{\mathfrak{C}}(\alpha, \tau)) = \operatorname{vars}(\alpha) \cup \operatorname{vars}(\tau)$.

(99) $\operatorname{Var}\operatorname{non}\alpha = \operatorname{Var}\alpha$.

(100) $\operatorname{vars}(\operatorname{non}\alpha) = \operatorname{vars}(\alpha)$.

Let us consider $\mathfrak{C}$ and let θ be a quasi-type of $\mathfrak{C}$. The functor $\operatorname{Var}\theta$ yields a subset of Vars and is defined as follows:

(Def. 48) $\operatorname{Var}\theta = \bigcup((\operatorname{vars}^{\operatorname{Vars}\mathfrak{C}}_{\mathbf{term}_{\mathfrak{C}}})^\circ \operatorname{adjs}\theta) \cup \operatorname{Var}(\text{the base of } \theta)$.

Let us consider $\mathfrak{C}$ and let θ be a quasi-type of $\mathfrak{C}$. Note that $\operatorname{Var}\theta$ is finite.

Let us consider $\mathfrak{C}$ and let θ be a quasi-type of $\mathfrak{C}$. The functor $\operatorname{vars}(\theta)$ yields a finite subset of Vars and is defined by:

(Def. 49) $\operatorname{vars}(\theta) = \operatorname{varcl}\operatorname{Var}\theta$.

We now state several propositions:

(101) For every quasi-type θ of $\mathfrak{C}$ holds $\operatorname{varcl}\operatorname{vars}(\theta) = \operatorname{vars}(\theta)$.

(102) For every quasi-type θ of $\mathfrak{C}$ and for every quasi-adjective α of $\mathfrak{C}$ holds $\operatorname{Var}(\alpha * \theta) = \operatorname{Var}\alpha \cup \operatorname{Var}\theta$.

(103) For every quasi-type θ of $\mathfrak{C}$ and for every quasi-adjective α of $\mathfrak{C}$ holds $\operatorname{vars}(\alpha * \theta) = \operatorname{vars}(\alpha) \cup \operatorname{vars}(\theta)$.

(104) $\operatorname{Var}(A*\vartheta) = \bigcup\{\operatorname{Var}\alpha; \alpha$ ranges over quasi-adjectives of $\mathfrak{C}$: $\alpha \in A\} \cup \operatorname{Var}\vartheta$.

(105) $\operatorname{vars}(A * \vartheta) = \bigcup\{\operatorname{vars}(\alpha); \alpha$ ranges over quasi-adjectives of $\mathfrak{C}$: $\alpha \in A\} \cup \operatorname{vars}(\vartheta)$.

(106) $\operatorname{Var}(\emptyset_{\operatorname{QuasiAdjs}\mathfrak{C}} * \vartheta) = \operatorname{Var}\vartheta$.

(107) η is ground iff $\operatorname{Var}\eta = \emptyset$.

Let us consider $\mathfrak{C}$ and let θ be a quasi-type of $\mathfrak{C}$. We say that θ is ground if and only if:

(Def. 50) $\operatorname{Var}\theta = \emptyset$.

Let us consider $\mathfrak{C}$. Observe that there exists an expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ which is ground and pure and there exists a quasi-adjective of $\mathfrak{C}$ which is ground.

Next we state the proposition

(108) For every ground pure expression τ of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ holds $\emptyset_{\operatorname{QuasiAdjs}\mathfrak{C}} * \tau$ is ground.

Let us consider $\mathfrak{C}$ and let τ be a ground pure expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$. Note that $\emptyset_{\operatorname{QuasiAdjs}\mathfrak{C}} * \tau$ is ground.

Let us consider $\mathfrak{C}$. Note that there exists a quasi-type of $\mathfrak{C}$ which is ground.

Let us consider $\mathfrak{C}$, let θ be a ground quasi-type of $\mathfrak{C}$, and let α be a ground quasi-adjective of $\mathfrak{C}$. Observe that $\alpha * \theta$ is ground.

9. Smooth Type Widening

The strict non empty poset VarPoset is defined by:

(Def. 51) $\operatorname{VarPoset} = (\langle \{\operatorname{varcl} A : A \text{ ranges over finite subsets of Vars}\}, \subseteq \rangle)^{\mathrm{op}}$.

One can prove the following propositions:

(109) For all elements x, y of VarPoset holds $x \leq y$ iff $y \subseteq x$.

(110) For every x holds x is an element of VarPoset iff x is a finite subset of Vars and $\operatorname{varcl} x = x$.

One can check that VarPoset has g.l.b.'s and l.u.b.'s.

The following proposition is true

(111) For all elements V_1, V_2 of VarPoset holds $V_1 \sqcup V_2 = V_1 \cap V_2$ and $V_1 \sqcap V_2 = V_1 \cup V_2$.

Let V_1, V_2 be elements of VarPoset. One can verify that functors $V_1 \sqcup V_2$ and $V_1 \cap V_2$ and functors $V_1 \sqcap V_2$ and $V_1 \cup V_2$ can be identified.

One can prove the following proposition

(112) For every non empty subset X of VarPoset holds $\sup X$ exists in VarPoset and $\sup X = \bigcap X$.

One can verify that VarPoset is up-complete.

The following proposition is true

(113) $\top_{\operatorname{VarPoset}} = \emptyset$.

Let us consider C. The functor vars-function C yielding a function from $\operatorname{QuasiTypes} C$ into the carrier of VarPoset is defined by:

(Def. 52) For every quasi-type T of C holds $(\text{vars-function}\, C)(T) = \operatorname{vars}(T)$.

Let L be a non empty poset. We say that L is smooth if and only if the condition (Def. 53) is satisfied.

(Def. 53) There exists an initialized constructor signature C and there exists a function f from L into VarPoset such that

(i) the carrier of $L \subseteq \operatorname{QuasiTypes} C$,

(ii) $f = \text{vars-function}\, C \restriction$the carrier of L, and

(iii) for all elements x, y of L holds f preserves sup of $\{x, y\}$.

Let C be an initialized constructor signature and let T be a ground quasi-type of C. One can check that $\langle \{T\}, \mathrm{id}_{\{T\}} \rangle$ is smooth.

10. Structural Induction

The scheme *StructInd* deals with an initialized constructor signature $\mathcal{A}$, an expression $\mathcal{B}$ of $\mathcal{A}$, and a unary predicate $\mathcal{P}$, and states that:

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

provided the following conditions are satisfied:

- For every variable x holds $\mathcal{P}[x_{\mathcal{A}}]$,
- Let c be a constructor operation symbol of $\mathcal{A}$ and p be a finite sequence of elements of $\operatorname{QuasiTerms} \mathcal{A}$. Suppose $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$ and for every quasi-term τ of $\mathcal{A}$ such that $\tau \in \operatorname{rng} p$ holds $\mathcal{P}[\tau]$. Then $\mathcal{P}[c^{\frown}(p)]$,
- For every expression α of $\mathcal{A}$ from $\mathbf{adj}_{\mathcal{A}}$ such that $\mathcal{P}[\alpha]$ holds $\mathcal{P}[\mathbf{non}_{\mathcal{A}}(\alpha)]$, and
- Let α be an expression of $\mathcal{A}$ from $\mathbf{adj}_{\mathcal{A}}$. Suppose $\mathcal{P}[\alpha]$. Let τ be an expression of $\mathcal{A}$ from $\mathbf{type}_{\mathcal{A}}$. If $\mathcal{P}[\tau]$, then $\mathcal{P}[*_{\mathcal{A}}(\alpha, \tau)]$.

Let Σ be a many sorted signature. We say that Σ has an operation for each sort if and only if:

(Def. 54) The carrier of $\Sigma \subseteq \operatorname{rng}$ (the result sort of Σ).

Let X be a many sorted set indexed by the carrier of Σ. We say that X has missing variables if and only if:

(Def. 55) $X^{-1}(\{\emptyset\}) \subseteq \operatorname{rng}$ (the result sort of Σ).

The following proposition is true

(114) Let Σ be a non void signature and X be a many sorted set indexed by the carrier of Σ. Then X has missing variables if and only if for every sort symbol s of Σ such that $X(s) = \emptyset$ there exists an operation symbol o of Σ such that the result sort of $o = s$.

Observe that MaxConstrSign has an operation for each sort. Let $\mathfrak{C}$ be a constructor signature. Observe that $\operatorname{Vars} \mathfrak{C}$ has missing variables.

Let Σ be a many sorted signature. Observe that every many sorted set indexed by the carrier of Σ which is non-empty has also missing variables.

Let Σ be a many sorted signature. Observe that there exists a many sorted set indexed by the carrier of Σ which has missing variables.

One can verify that there exists a constructor signature which is initialized and strict and has an operation for each sort.

Let $\mathfrak{C}$ be a many sorted signature with an operation for each sort. Observe that every many sorted set indexed by the carrier of $\mathfrak{C}$ has missing variables.

Let G be a non empty tree construction structure. Then the terminals of G is a subset of G. Then the nonterminals of G is a subset of G.

Next we state a number of propositions:

(115) Let D_1, D_2 be non empty tree construction structures. Suppose the rules of $D_1 \subseteq$ the rules of D_2. Then
  (i) the nonterminals of $D_1 \subseteq$ the nonterminals of D_2,
  (ii) (the carrier of D_1) $\cap$ (the terminals of D_2) $\subseteq$ the terminals of D_1, and
  (iii) if the terminals of $D_1 \subseteq$ the terminals of D_2, then the carrier of $D_1 \subseteq$ the carrier of D_2.

(116) Let D_1, D_2 be non empty tree construction structures. Suppose the terminals of $D_1 \subseteq$ the terminals of D_2 and the rules of $D_1 \subseteq$ the rules of D_2. Then $\mathrm{TS}(D_1) \subseteq \mathrm{TS}(D_2)$.

(117) Let Σ be a many sorted signature and X, Y be many sorted sets indexed by the carrier of Σ. If $X \subseteq Y$, then if X has missing variables, then Y has missing variables.

(118) For every set Σ and for all many sorted sets X, Y indexed by Σ such that $X \subseteq Y$ holds $\bigcup \mathrm{coprod}(X) \subseteq \bigcup \mathrm{coprod}(Y)$.

(119) Let Σ be a non void signature and X, Y be many sorted sets indexed by the carrier of Σ. If $X \subseteq Y$, then the carrier of $\mathrm{DTConMSA}(X) \subseteq$ the carrier of $\mathrm{DTConMSA}(Y)$.

(120) Let Σ be a non void signature and X be a many sorted set indexed by the carrier of Σ. Suppose X has missing variables. Then the nonterminals of $\mathrm{DTConMSA}(X) =$ (the operation symbols of Σ) $\times$ {the carrier of Σ} and the terminals of $\mathrm{DTConMSA}(X) = \bigcup \mathrm{coprod}(X)$.

(121) Let Σ be a non void signature and X, Y be many sorted sets indexed by the carrier of Σ. Suppose $X \subseteq Y$ and X has missing variables. Then the terminals of $\mathrm{DTConMSA}(X) \subseteq$ the terminals of $\mathrm{DTConMSA}(Y)$ and the rules of $\mathrm{DTConMSA}(X) \subseteq$ the rules of $\mathrm{DTConMSA}(Y)$ and $\mathrm{TS}(\mathrm{DTConMSA}(X)) \subseteq \mathrm{TS}(\mathrm{DTConMSA}(Y))$.

(122) For every set τ holds $\tau \in$ the terminals of $\mathrm{DTConMSA}(\mathrm{Vars}\,\mathfrak{C})$ iff there exists a variable x such that $\tau = \langle x, \mathbf{term}_{\mathfrak{C}} \rangle$.

(123) Let τ be a set. Then $\tau \in$ the nonterminals of $\mathrm{DTConMSA}(\mathrm{Vars}\,\mathfrak{C})$ if and only if one of the following conditions is satisfied:
  (i) $\tau = \langle *_{\mathfrak{C}}$, the carrier of $\mathfrak{C} \rangle$, or
  (ii) $\tau = \langle \mathbf{non}_{\mathfrak{C}}$, the carrier of $\mathfrak{C} \rangle$, or
  (iii) there exists a constructor operation symbol c of $\mathfrak{C}$ such that $\tau = \langle c,$

the carrier of $\mathfrak{C}\rangle$.

(124) Let Σ be a non void signature, X be a many sorted set indexed by the carrier of Σ with missing variables, and τ be a set. Suppose $\tau \in \bigcup$(the sorts of $\mathrm{Free}_\Sigma(X)$). Then τ is a term of Σ over $X \cup$((the carrier of Σ) $\longmapsto \{0\}$).

(125) Let Σ be a non void signature, X be a many sorted set indexed by the carrier of Σ with missing variables, and τ be a term of Σ over $X \cup$((the carrier of Σ) $\longmapsto \{0\}$). If $\tau \in \bigcup$(the sorts of $\mathrm{Free}_\Sigma(X)$), then $\tau \in$ (the sorts of $\mathrm{Free}_\Sigma(X)$)(the sort of τ).

(126) Let G be a non empty tree construction structure, s be an element of G, and p be a finite sequence. Suppose $s \Rightarrow p$. Then p is a finite sequence of elements of the carrier of G.

(127) Let Σ be a non void signature, X, Y be many sorted sets indexed by the carrier of Σ, g_1 be a symbol of $\mathrm{DTConMSA}(X)$, g_2 be a symbol of $\mathrm{DTConMSA}(Y)$, p_1 be a finite sequence of elements of the carrier of $\mathrm{DTConMSA}(X)$, and p_2 be a finite sequence of elements of the carrier of $\mathrm{DTConMSA}(Y)$. If $g_1 = g_2$ and $p_1 = p_2$ and $g_1 \Rightarrow p_1$, then $g_2 \Rightarrow p_2$.

(128) Let Σ be a non void signature and X be a many sorted set indexed by the carrier of Σ with missing variables. Then $\bigcup$(the sorts of $\mathrm{Free}_\Sigma(X)$) = $\mathrm{TS}(\mathrm{DTConMSA}(X))$.

Let Σ be a non void signature and let X be a many sorted set indexed by the carrier of Σ. A unary operation on $\bigcup$(the sorts of $\mathrm{Free}_\Sigma(X)$) is said to be a transformation of Σ-terms over X if:

(Def. 56) For every sort symbol s of Σ holds it$^\circ$(the sorts of $\mathrm{Free}_\Sigma(X)$)$(s) \subseteq$ (the sorts of $\mathrm{Free}_\Sigma(X)$)(s).

The following two propositions are true:

(129) Let Σ be a non void signature, X be a non empty many sorted set indexed by the carrier of Σ, and f be a unary operation on $\bigcup$(the sorts of $\mathrm{Free}_\Sigma(X)$). Then f is a transformation of Σ-terms over X if and only if for every sort symbol s of Σ and for every set α such that $\alpha \in$ (the sorts of $\mathrm{Free}_\Sigma(X)$)(s) holds $f(\alpha) \in$ (the sorts of $\mathrm{Free}_\Sigma(X)$)(s).

(130) Let Σ be a non void signature, X be a non empty many sorted set indexed by the carrier of Σ, f be a transformation of Σ-terms over X, s be a sort symbol of Σ, and p be a finite sequence of elements of (the sorts of $\mathrm{Free}_\Sigma(X)$)(s). Then $f \cdot p$ is a finite sequence of elements of (the sorts of $\mathrm{Free}_\Sigma(X)$)(s) and $\overline{\overline{(f \cdot p \textbf{ qua } \mathrm{set})}} = \mathrm{len}\, p$.

Let Σ be a non void signature, let X be a many sorted set indexed by the carrier of Σ, and let τ be a transformation of Σ-terms over X. We say that τ is substitution if and only if the condition (Def. 57) is satisfied.

(Def. 57) Let o be an operation symbol of Σ and p, p' be finite sequences of elements of $\mathrm{Free}_\Sigma(X)$. Suppose $\langle o$, the carrier of $\Sigma\rangle$-tree$(p) \in \bigcup$(the sorts of

Free$_\Sigma(X)$) and $p' = \tau \cdot p$. Then $\tau(\langle\!\langle o$, the carrier of $\Sigma\rangle$-tree$(p)) = \langle\!\langle o$, the carrier of $\Sigma\rangle$-tree(p').

The scheme *StructDef* deals with an initialized constructor signature $\mathcal{A}$, two unary functors $\mathcal{F}$ and $\mathcal{G}$ yielding expressions of $\mathcal{A}$, and two binary functors $\mathcal{H}$ and $\mathcal{I}$ yielding expressions of $\mathcal{A}$, and states that:

There exists a transformation f of $\mathcal{A}$-terms over Vars $\mathcal{A}$ such that

(i) for every variable x holds $f(x_\mathcal{A}) = \mathcal{F}(x)$,

(ii) for every constructor operation symbol c of $\mathcal{A}$ and for all finite sequences p, p' of elements of QuasiTerms $\mathcal{A}$ such that len $p =$ len Arity(c) and $p' = f \cdot p$ holds $f(c^\frown(p)) = \mathcal{H}(c, p')$,

(iii) for every expression α of $\mathcal{A}$ from $\mathbf{adj}_\mathcal{A}$ holds $f(\mathbf{non}_\mathcal{A}(\alpha)) = \mathcal{G}(f(\alpha))$, and

(iv) for every expression α of $\mathcal{A}$ from $\mathbf{adj}_\mathcal{A}$ and for every expression τ of $\mathcal{A}$ from $\mathbf{type}_\mathcal{A}$ holds $f(*_\mathcal{A}(\alpha, \tau)) = \mathcal{I}(f(\alpha), f(\tau))$

provided the parameters meet the following requirements:

- For every variable x holds $\mathcal{F}(x)$ is a quasi-term of $\mathcal{A}$,
- Let c be a constructor operation symbol of $\mathcal{A}$ and p be a finite sequence of elements of QuasiTerms $\mathcal{A}$. Suppose len $p =$ len Arity(c). Then $\mathcal{H}(c, p)$ is an expression of $\mathcal{A}$ from the result sort of c,
- For every expression α of $\mathcal{A}$ from $\mathbf{adj}_\mathcal{A}$ holds $\mathcal{G}(\alpha)$ is an expression of $\mathcal{A}$ from $\mathbf{adj}_\mathcal{A}$, and
- Let α be an expression of $\mathcal{A}$ from $\mathbf{adj}_\mathcal{A}$ and τ be an expression of $\mathcal{A}$ from $\mathbf{type}_\mathcal{A}$. Then $\mathcal{I}(\alpha, \tau)$ is an expression of $\mathcal{A}$ from $\mathbf{type}_\mathcal{A}$.

11. Substitution

Let A be a set, let x, y be sets, and let α, β be elements of A. Then IFIN(x, y, α, β) is an element of A.

Let $\mathfrak{C}$ be an initialized constructor signature. A valuation of $\mathfrak{C}$ is a partial function from Vars to QuasiTerms $\mathfrak{C}$.

Let $\mathfrak{C}$ be an initialized constructor signature and let f be a valuation of $\mathfrak{C}$. We say that f is irrelevant if and only if:

(Def. 58) For every variable x such that $x \in \operatorname{dom} f$ there exists a variable y such that $f(x) = y_\mathfrak{C}$.

Let $\mathfrak{C}$ be an initialized constructor signature and let f be a valuation of $\mathfrak{C}$. We introduce f is relevant as an antonym of f is irrelevant.

Let X, Y be sets. Observe that there exists a partial function from X to Y which is empty.

Let $\mathfrak{C}$ be an initialized constructor signature. Observe that every valuation of $\mathfrak{C}$ which is empty is also irrelevant.

Let $\mathfrak{C}$ be an initialized constructor signature. Note that there exists a valuation of $\mathfrak{C}$ which is empty, irrelevant, and one-to-one.

Let $\mathfrak{C}$ be an initialized constructor signature and let X be a subset of Vars. The functor $\mathrm{idval}_{\mathfrak{C}} X$ yielding a valuation of $\mathfrak{C}$ is defined by:

(Def. 59) $\mathrm{idval}_{\mathfrak{C}} X = \{\langle x, x_{\mathfrak{C}}\rangle; x \text{ ranges over variables: } x \in X\}$.

Next we state the proposition

(131) For every subset X of Vars holds $\mathrm{dom}\,\mathrm{idval}_{\mathfrak{C}} X = X$ and for every variable x such that $x \in X$ holds $(\mathrm{idval}_{\mathfrak{C}} X)(x) = x_{\mathfrak{C}}$.

Let $\mathfrak{C}$ be an initialized constructor signature and let X be a subset of Vars. One can check that $\mathrm{idval}_{\mathfrak{C}} X$ is irrelevant and one-to-one.

Let $\mathfrak{C}$ be an initialized constructor signature and let X be an empty subset of Vars. One can check that $\mathrm{idval}_{\mathfrak{C}} X$ is empty.

Let us consider $\mathfrak{C}$ and let f be a valuation of $\mathfrak{C}$. The functor $f^{\#}$ yielding a transformation of $\mathfrak{C}$-terms over $\mathrm{Vars}\,\mathfrak{C}$ is defined by the conditions (Def. 60).

(Def. 60)(i) For every variable x holds if $x \in \mathrm{dom}\, f$, then $f^{\#}(x_{\mathfrak{C}}) = f(x)$ and if $x \notin \mathrm{dom}\, f$, then $f^{\#}(x_{\mathfrak{C}}) = x_{\mathfrak{C}}$,

(ii) for every constructor operation symbol c of $\mathfrak{C}$ and for all finite sequences p, p' of elements of $\mathrm{QuasiTerms}\,\mathfrak{C}$ such that $\mathrm{len}\, p = \mathrm{len}\,\mathrm{Arity}(c)$ and $p' = f^{\#} \cdot p$ holds $f^{\#}(c^{\frown}(p)) = c^{\frown}(p')$,

(iii) for every expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ holds $f^{\#}(\mathbf{non}_{\mathfrak{C}}(\alpha)) = \mathbf{non}_{\mathfrak{C}}(f^{\#}(\alpha))$, and

(iv) for every expression α of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$ and for every expression τ of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$ holds $f^{\#}(*_{\mathfrak{C}}(\alpha, \tau)) = *_{\mathfrak{C}}(f^{\#}(\alpha), f^{\#}(\tau))$.

Let us consider $\mathfrak{C}$ and let f be a valuation of $\mathfrak{C}$. Observe that $f^{\#}$ is substitution.

In the sequel f denotes a valuation of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$, f, η. The functor $\eta[f]$ yielding an expression of $\mathfrak{C}$ is defined as follows:

(Def. 61) $\eta[f] = f^{\#}(\eta)$.

Let us consider $\mathfrak{C}$, f and let p be a finite sequence. Let us assume that $\mathrm{rng}\, p \subseteq \bigcup(\text{the sorts of } \mathrm{Free}_{\mathfrak{C}}(\mathrm{Vars}\,\mathfrak{C}))$. The functor $p[f]$ yields a finite sequence and is defined as follows:

(Def. 62) $p[f] = f^{\#} \cdot p$.

Let us consider $\mathfrak{C}$, f and let p be a finite sequence of elements of $\mathrm{QuasiTerms}\,\mathfrak{C}$. Then $p[f]$ is a finite sequence of elements of $\mathrm{QuasiTerms}\,\mathfrak{C}$ and it can be characterized by the condition:

(Def. 63) $p[f] = f^{\#} \cdot p$.

In the sequel x is a variable.

The following propositions are true:

(132) If $x \notin \mathrm{dom}\, f$, then $x_{\mathfrak{C}}[f] = x_{\mathfrak{C}}$.

(133) If $x \in \operatorname{dom} f$, then $x_{\mathfrak{C}}[f] = f(x)$.

(134) If $\operatorname{len} p = \operatorname{len} \operatorname{Arity}(c)$, then $\vec{c}(p)[f] = \vec{c}(p[f])$.

(135) $\mathbf{non}_{\mathfrak{C}}(\alpha)[f] = \mathbf{non}_{\mathfrak{C}}(\alpha[f])$.

(136) $*_{\mathfrak{C}}(\alpha, \tau)[f] = *_{\mathfrak{C}}(\alpha[f], \tau[f])$.

(137) For every subset X of Vars holds $\eta[\operatorname{idval}_{\mathfrak{C}} X] = \eta$.

(138) For every subset X of Vars holds $(\operatorname{idval}_{\mathfrak{C}} X)^{\#} = \operatorname{id}_{\bigcup (\text{the sorts of } \operatorname{Free}_{\mathfrak{C}}(\operatorname{Vars} \mathfrak{C}))}$.

(139) For every empty valuation f of $\mathfrak{C}$ holds $\eta[f] = \eta$.

(140) For every empty valuation f of $\mathfrak{C}$ holds $f^{\#} = \operatorname{id}_{\bigcup (\text{the sorts of } \operatorname{Free}_{\mathfrak{C}}(\operatorname{Vars} \mathfrak{C}))}$.

Let us consider $\mathfrak{C}$, f and let τ be a quasi-term of $\mathfrak{C}$. Then $\tau[f]$ is a quasi-term of $\mathfrak{C}$.

Let us consider $\mathfrak{C}$, f and let α be an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$. Then $\alpha[f]$ is an expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$.

Let us consider $\mathfrak{C}$, f and let α be a positive expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$. Note that $\alpha[f]$ is positive.

Let us consider $\mathfrak{C}$, f and let α be a negative expression of $\mathfrak{C}$ from $\mathbf{adj}_{\mathfrak{C}}$. Observe that $\alpha[f]$ is negative.

Let us consider $\mathfrak{C}$, f and let α be a quasi-adjective of $\mathfrak{C}$. Then $\alpha[f]$ is a quasi-adjective of $\mathfrak{C}$.

We now state the proposition

(141) $(\operatorname{non} \alpha)[f] = \operatorname{non}(\alpha[f])$.

Let us consider $\mathfrak{C}$, f and let τ be an expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$. Then $\tau[f]$ is an expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$.

Let us consider $\mathfrak{C}$, f and let τ be a pure expression of $\mathfrak{C}$ from $\mathbf{type}_{\mathfrak{C}}$. Observe that $\tau[f]$ is pure.

One can prove the following two propositions:

(142) Let f be an irrelevant one-to-one valuation of $\mathfrak{C}$. Then there exists an irrelevant one-to-one valuation g of $\mathfrak{C}$ such that for all variables x, y holds $x \in \operatorname{dom} f$ and $f(x) = y_{\mathfrak{C}}$ if and only if $y \in \operatorname{dom} g$ and $g(y) = x_{\mathfrak{C}}$.

(143) Let f, g be irrelevant one-to-one valuations of $\mathfrak{C}$. Suppose that for all variables x, y such that $x \in \operatorname{dom} f$ and $f(x) = y_{\mathfrak{C}}$ holds $y \in \operatorname{dom} g$ and $g(y) = x_{\mathfrak{C}}$. Let given η. If $\operatorname{Var} \eta \subseteq \operatorname{dom} f$, then $\eta[f][g] = \eta$.

Let us consider $\mathfrak{C}$, f and let A be a subset of $\operatorname{QuasiAdjs} \mathfrak{C}$. The functor $A[f]$ yielding a subset of $\operatorname{QuasiAdjs} \mathfrak{C}$ is defined as follows:

(Def. 64) $A[f] = \{\alpha[f]; \alpha \text{ ranges over quasi-adjectives of } \mathfrak{C}\colon \alpha \in A\}$.

The following three propositions are true:

(144) For every subset A of $\operatorname{QuasiAdjs} \mathfrak{C}$ and for every quasi-adjective α of $\mathfrak{C}$ such that $A = \{\alpha\}$ holds $A[f] = \{\alpha[f]\}$.

(145) For all subsets A, B of $\operatorname{QuasiAdjs} \mathfrak{C}$ holds $(A \cup B)[f] = A[f] \cup B[f]$.

(146) For all subsets A, B of $\operatorname{QuasiAdjs} \mathfrak{C}$ such that $A \subseteq B$ holds $A[f] \subseteq B[f]$.

Let $\mathfrak{C}$ be an initialized constructor signature, let f be a valuation of $\mathfrak{C}$, and let A be a finite subset of QuasiAdjs $\mathfrak{C}$. One can check that $A[f]$ is finite.

Let $\mathfrak{C}$ be an initialized constructor signature, let f be a valuation of $\mathfrak{C}$, and let θ be a quasi-type of $\mathfrak{C}$. The functor $\theta[f]$ yields a quasi-type of $\mathfrak{C}$ and is defined by:

(Def. 65) $\theta[f] = (\mathrm{adjs}\,\theta)[f] * (\text{the base of } \theta)[f]$.

Next we state two propositions:

(147) For every quasi-type θ of $\mathfrak{C}$ holds $\mathrm{adjs}(\theta[f]) = (\mathrm{adjs}\,\theta)[f]$ and the base of $\theta[f] = (\text{the base of } \theta)[f]$.

(148) For every quasi-type θ of $\mathfrak{C}$ and for every quasi-adjective α of $\mathfrak{C}$ holds $(\alpha * \theta)[f] = \alpha[f] * \theta[f]$.

Let $\mathfrak{C}$ be an initialized constructor signature and let f, g be valuations of $\mathfrak{C}$. The functor $f[g]$ yields a valuation of $\mathfrak{C}$ and is defined by:

(Def. 66) $\mathrm{dom}(f[g]) = \mathrm{dom}\, f \cup \mathrm{dom}\, g$ and for every variable x such that $x \in \mathrm{dom}(f[g])$ holds $f[g](x) = x_{\mathfrak{C}}[f][g]$.

Let $\mathfrak{C}$ be an initialized constructor signature and let f, g be irrelevant valuations of $\mathfrak{C}$. One can check that $f[g]$ is irrelevant.

The following three propositions are true:

(149) For all valuations f_1, f_2 of $\mathfrak{C}$ holds $\eta[f_1][f_2] = \eta[f_1[f_2]]$.

(150) For every subset A of QuasiAdjs $\mathfrak{C}$ and for all valuations f_1, f_2 of $\mathfrak{C}$ holds $A[f_1][f_2] = A[f_1[f_2]]$.

(151) For every quasi-type θ of $\mathfrak{C}$ and for all valuations f_1, f_2 of $\mathfrak{C}$ holds $\theta[f_1][f_2] = \theta[f_1[f_2]]$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek. Introduction to trees. *Formalized Mathematics*, 1(**2**):421–427, 1990.
[4] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[5] Grzegorz Bancerek. Tarski's classes and ranks. *Formalized Mathematics*, 1(**3**):563–567, 1990.
[6] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[7] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.
[8] Grzegorz Bancerek. Sets and functions of trees and joining operations of trees. *Formalized Mathematics*, 3(**2**):195–204, 1992.
[9] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(**1**):77–82, 1993.
[10] Grzegorz Bancerek. Terms over many sorted universal algebra. *Formalized Mathematics*, 5(**2**):191–198, 1996.
[11] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[12] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[13] Grzegorz Bancerek. On semilattice structure of Mizar types. *Formalized Mathematics*, 11(**4**):355–369, 2003.

[14] 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.
[15] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[16] Grzegorz Bancerek and Artur Korniłowicz. Yet another construction of free algebra. *Formalized Mathematics*, 9(**4**):779–785, 2001.
[17] Grzegorz Bancerek and Yatsuka Nakamura. Full adder circuit. Part I. *Formalized Mathematics*, 5(**3**):367–380, 1996.
[18] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.
[19] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[20] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[21] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[22] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[23] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[24] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[25] Patricia L. Carlson and Grzegorz Bancerek. Context-free grammar – part 1. *Formalized Mathematics*, 2(**5**):683–687, 1991.
[26] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[27] Adam Grabowski and Robert Milewski. Boolean posets, posets under inclusion and products of relational structures. *Formalized Mathematics*, 6(**1**):117–121, 1997.
[28] Yatsuka Nakamura. Determinant of some matrices of field elements. *Formalized Mathematics*, 14(**1**):1–5, 2006.
[29] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[30] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[31] Beata Perkowska. Free many sorted universal algebra. *Formalized Mathematics*, 5(**1**):67–74, 1996.
[32] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[33] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[34] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[35] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[36] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[37] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[38] Andrzej Trybulec. On the sets inhabited by numbers. *Formalized Mathematics*, 11(**4**):341–347, 2003.
[39] Andrzej Trybulec and Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.
[40] Wojciech A. Trybulec and Grzegorz Bancerek. Kuratowski – Zorn lemma. *Formalized Mathematics*, 1(**2**):387–393, 1990.
[41] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[42] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[43] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received April 21, 2008