

Analysis of Algorithms: An Example of a Sort Algorithm

Grzegorz Bancerek
Association of Mizar Users
Białystok, Poland

Summary. We analyse three algorithms: exponentiation by squaring, calculation of maximum, and sorting by exchanging in terms of program algebra over an algebra.

MML identifier: AOFA_A01, version: 8.0.01 5.5.1167

The notation and terminology used in this paper have been introduced in the following articles: [37], [1], [2], [17], [3], [4], [13], [18], [34], [23], [29], [19], [20], [15], [5], [33], [6], [27], [38], [28], [30], [14], [7], [8], [31], [16], [24], [26], [35], [9], [21], [32], [39], [36], [10], [11], [25], [12], and [22].

1. EXPONENTIATION BY SQUARING REVISITED

Now we state the propositions:

- (1) (i) $1 \bmod 2 = 1$, and
(ii) $2 \bmod 2 = 0$.
- (2) Let us consider a non empty non void many sorted signature Σ , an algebra \mathfrak{A} over Σ , a subalgebra \mathfrak{B} of \mathfrak{A} , a sort symbol s of Σ , and a set a . Suppose $a \in (\text{the sorts of } \mathfrak{B})(s)$. Then $a \in (\text{the sorts of } \mathfrak{A})(s)$.
- (3) Let us consider a non empty set I , sets a, b, c , and an element i of I . Then $c \in (i\text{-singleton } a)(b)$ if and only if $b = i$ and $c = a$.
- (4) Let us consider a non empty set I , sets a, b, c, d , and elements i, j of I . Then $c \in (i\text{-singleton } a \cup j\text{-singleton } d)(b)$ if and only if $b = i$ and $c = a$ or $b = j$ and $c = d$. The theorem is a consequence of (3).

Let Σ be a boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1 and \mathfrak{A} be a non-empty algebra over Σ . We say that \mathfrak{A} is integer if and only if

(Def. 1) There exists an image \mathfrak{C} of \mathfrak{A} such that \mathfrak{C} is a boolean correct algebra over Σ with integers with connectives from 4 and the sort at 1.

Now we state the propositions:

- (5) Let us consider a non empty non void many sorted signature Σ and a non-empty algebra \mathfrak{A} over Σ . Then $\text{Im id}_\alpha =$ the algebra of \mathfrak{A} , where α is the sorts of \mathfrak{A} .
- (6) Let us consider a non empty non void many sorted signature Σ . Then every non-empty algebra over Σ is an image of \mathfrak{A} . The theorem is a consequence of (5). PROOF: \mathfrak{A} is \mathfrak{A} -image. \square

Let Σ be a boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1. One can verify that there exists a non-empty algebra over Σ which is integer.

Let \mathfrak{A} be an integer non-empty algebra over Σ . Note that there exists an image of \mathfrak{A} which is boolean correct.

Let us note that there exists a boolean correct image of \mathfrak{A} which has integers with connectives from 4 and the sort at 1.

Now we state the proposition:

- (7) Let us consider a non empty non void many sorted signature Σ , a non-empty algebra \mathfrak{A} over Σ , an operation symbol o of Σ , a set a , and a sort symbol r of Σ . Suppose o is of type $a \rightarrow r$. Then
 - (i) $\text{Den}(o, \mathfrak{A})$ is a function from (the sorts of $\mathfrak{A})^\#(a)$ into (the sorts of $\mathfrak{A})(r)$, and
 - (ii) $\text{Args}(o, \mathfrak{A}) = (\text{the sorts of } \mathfrak{A})^\#(a)$, and
 - (iii) $\text{Result}(o, \mathfrak{A}) = (\text{the sorts of } \mathfrak{A})(r)$.

Let Σ be a boolean correct non empty non void boolean signature and \mathfrak{A} be a boolean correct non-empty algebra over Σ . Observe that every non-empty subalgebra of \mathfrak{A} is boolean correct.

Let Σ be a boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1 and \mathfrak{A} be a boolean correct non-empty algebra over Σ with integers with connectives from 4 and the sort at 1. Note that every non-empty subalgebra of \mathfrak{A} has integers with connectives from 4 and the sort at 1.

Let X be a non-empty many sorted set indexed by the carrier of Σ . Let us observe that $\mathfrak{F}_\Sigma(X)$ is integer as a non-empty algebra over Σ .

Now we state the proposition:

- (8) Let us consider a non empty non void many sorted signature Σ , algebras $\mathfrak{A}_1, \mathfrak{A}_2, \mathfrak{B}_1$ over Σ , and a non-empty algebra \mathfrak{B}_2 over Σ . Suppose

- (i) the algebra of $\mathfrak{A}_1 =$ the algebra of \mathfrak{A}_2 , and
- (ii) the algebra of $\mathfrak{B}_1 =$ the algebra of \mathfrak{B}_2 .

Let us consider a many sorted function h_1 from \mathfrak{A}_1 into \mathfrak{B}_1 and a many sorted function h_2 from \mathfrak{A}_2 into \mathfrak{B}_2 . Suppose

- (iii) $h_1 = h_2$, and
- (iv) h_1 is an epimorphism of \mathfrak{A}_1 onto \mathfrak{B}_1 .

Then h_2 is an epimorphism of \mathfrak{A}_2 onto \mathfrak{B}_2 .

Let Σ be a boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1 and X be a non-empty many sorted set indexed by the carrier of Σ . Let us note that there exists an including Σ -terms over X non-empty free variable algebra over Σ which is vf-free and integer.

Let Σ be a non empty non void many sorted signature. Let \mathfrak{T} be an including Σ -terms over X non-empty algebra over Σ . The functor $\text{FreeGenerator}(\mathfrak{T})$ yielding a non-empty generator set of \mathfrak{T} is defined by the term

(Def. 2) $\text{FreeGenerator}(X)$.

Let X_0 be a countable non-empty many sorted set indexed by the carrier of Σ and \mathfrak{T} be an including Σ -terms over X_0 non-empty algebra over Σ . Let us observe that $\text{FreeGenerator}(\mathfrak{T})$ is $\text{Equations}(\Sigma, \mathfrak{T})$ -free and non-empty.

Let X be a non-empty many sorted set indexed by the carrier of Σ , \mathfrak{T} be an including Σ -terms over X algebra over Σ , and G be a generator set of \mathfrak{T} . We say that G is basic if and only if

(Def. 3) $\text{FreeGenerator}(\mathfrak{T}) \subseteq G$.

Let s be a sort symbol of Σ and x be an element of $G(s)$. We say that x is pure if and only if

(Def. 4) $x \in (\text{FreeGenerator}(\mathfrak{T}))(s)$.

Observe that $\text{FreeGenerator}(\mathfrak{T})$ is basic.

Note that there exists a non-empty generator set of \mathfrak{T} which is basic.

Let G be a basic generator set of \mathfrak{T} and s be a sort symbol of Σ . One can check that there exists an element of $G(s)$ which is pure.

Now we state the proposition:

- (9) Let us consider a non empty non void many sorted signature Σ , a non-empty many sorted set X indexed by the carrier of Σ , an including Σ -terms over X algebra \mathfrak{T} over Σ , a basic generator set G of \mathfrak{T} , a sort symbol s of Σ , and a set a . Then a is a pure element of $G(s)$ if and only if $a \in (\text{FreeGenerator}(\mathfrak{T}))(s)$.

Let Σ be a non empty non void many sorted signature, X be a non-empty many sorted set indexed by the carrier of Σ , \mathfrak{T} be an including Σ -terms over X algebra over Σ , and G be a generator system over Σ , X , and \mathfrak{T} . We say that G is basic if and only if

(Def. 5) The generators of G are basic.

Observe that there exists a generator system over Σ , X , and \mathfrak{T} which is basic.

Let G be a basic generator system over Σ , X , and \mathfrak{T} . Note that the generators of G are basic.

In this paper Σ denotes a boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1, X denotes a non-empty many sorted set indexed by the carrier of Σ , \mathfrak{T} denotes a vf-free including Σ -terms over X integer non-empty free variable algebra over Σ , \mathfrak{C} denotes a boolean correct non-empty image of \mathfrak{T} with integers with connectives from 4 and the sort at 1, G denotes a basic generator system over Σ , X , and \mathfrak{T} , \mathfrak{A} denotes a if-while algebra over the generators of G , I denotes an integer sort symbol of Σ , x, y, z, m denote pure elements of (the generators of G)(I), b denotes a pure element of (the generators of G)(the boolean sort of Σ), τ, τ_1, τ_2 denote elements of \mathfrak{T} from I , P denotes an algorithm of \mathfrak{A} , and s, s_1, s_2 denote elements of \mathfrak{C} -States(the generators of G).

Let Σ be a boolean correct non empty non void boolean signature and \mathfrak{A} be a non-empty algebra over Σ . The functor $\text{false}_{\mathfrak{A}}$ yielding an element of \mathfrak{A} from the boolean sort of Σ is defined by the term

(Def. 6) $\neg \text{true}_{\mathfrak{A}}$.

In this paper f denotes an execution function of \mathfrak{A} over \mathfrak{C} -States(the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G).

Now we state the proposition:

$$(10) \quad \text{false}_{\mathfrak{C}} = \text{false}.$$

Let Σ be a boolean correct non empty non void boolean signature, X be a non-empty many sorted set indexed by the carrier of Σ , \mathfrak{T} be an including Σ -terms over X algebra over Σ , G be a generator system over Σ , X , and \mathfrak{T} , b be an element of (the generators of G)(the boolean sort of Σ), \mathfrak{C} be an image of \mathfrak{T} , \mathfrak{A} be a pre-if-while algebra, f be an execution function of \mathfrak{A} over \mathfrak{C} -States(the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G), s be an element of \mathfrak{C} -States(the generators of G), and P be an algorithm of \mathfrak{A} . Note that the functor $f(s, P)$ yields an element of \mathfrak{C} -States(the generators of G). Let Σ be a non empty non void many sorted signature, \mathfrak{T} be a non-empty algebra over Σ , G be a non-empty generator set of \mathfrak{T} , s be a sort symbol of Σ , and x be an element of $G(s)$. The functor ${}^{\textcircled{x}}$ yielding an element of \mathfrak{T} from s is defined by the term

(Def. 7) x .

Let us consider $\Sigma, X, \mathfrak{T}, G, \mathfrak{A}, b, I, \tau_1$, and τ_2 . The functors $b \text{leq}(\tau_1, \tau_2, \mathfrak{A})$ and $b \text{gt}(\tau_1, \tau_2, \mathfrak{A})$ yielding algorithms of \mathfrak{A} are defined by the terms, respectively.

(Def. 8) $b :=_{\mathfrak{A}}(\text{leq}(\tau_1, \tau_2))$.

(Def. 9) $b := \mathfrak{A}(\neg \text{leq}(\tau_1, \tau_2))$.

The functor $2_{\mathfrak{X}}^I$ yielding an element of \mathfrak{X} from I is defined by the term

(Def. 10) $1_{\mathfrak{X}}^I + 1_{\mathfrak{X}}^I$.

Let us consider G , \mathfrak{A} , and b . Let us consider τ . The functors τ is odd(b, \mathfrak{A}) and τ is even(b, \mathfrak{A}) yielding algorithms of \mathfrak{A} are defined by the terms, respectively.

(Def. 11) $b \text{gt}(\tau \text{ mod } 2_{\mathfrak{X}}^I, 0_{\mathfrak{X}}^I, \mathfrak{A})$.

(Def. 12) $b \text{leq}(\tau \text{ mod } 2_{\mathfrak{X}}^I, 0_{\mathfrak{X}}^I, \mathfrak{A})$.

Let us consider \mathfrak{C} . Let us consider s . Let x be an element of (the generators of G)(I). Let us note that $s(I)(x)$ is integer.

Let us consider τ . Let us note that τ value at(\mathfrak{C}, s) is integer.

In the sequel u denotes a many sorted function from FreeGenerator(\mathfrak{X}) into the sorts of \mathfrak{C} .

Let us consider Σ , X , \mathfrak{X} , \mathfrak{C} , I , u , and τ . One can verify that τ value at(\mathfrak{C}, u) is integer.

Let us consider G . Let us consider s . Let τ be an element of \mathfrak{X} from the boolean sort of Σ . One can verify that τ value at(\mathfrak{C}, s) is boolean.

Let us consider u . One can check that τ value at(\mathfrak{C}, u) is boolean.

Let us consider an operation symbol o of Σ . Now we state the propositions:

- (11) Suppose $o = (\text{the connectives of } \Sigma)(1)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $o = (\text{the connectives of } \Sigma)(1)$, and
 - (ii) $\text{Arity}(o) = \emptyset$, and
 - (iii) the result sort of $o = \text{the boolean sort of } \Sigma$.
- (12) Suppose $o = (\text{the connectives of } \Sigma)(2)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $o = (\text{the connectives of } \Sigma)(2)$, and
 - (ii) $\text{Arity}(o) = \langle \text{the boolean sort of } \Sigma \rangle$, and
 - (iii) the result sort of $o = \text{the boolean sort of } \Sigma$.
- (13) Suppose $o = (\text{the connectives of } \Sigma)(3)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $o = (\text{the connectives of } \Sigma)(3)$, and
 - (ii) $\text{Arity}(o) = \langle \text{the boolean sort of } \Sigma, \text{the boolean sort of } \Sigma \rangle$, and
 - (iii) the result sort of $o = \text{the boolean sort of } \Sigma$.
- (14) Suppose $o = (\text{the connectives of } \Sigma)(4)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \emptyset$, and
 - (ii) the result sort of $o = I$.
- (15) Suppose $o = (\text{the connectives of } \Sigma)(5)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \emptyset$, and
 - (ii) the result sort of $o = I$.

- (16) Suppose $o = (\text{the connectives of } \Sigma)(6)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \langle I \rangle$, and
 - (ii) the result sort of $o = I$.
- (17) Suppose $o = (\text{the connectives of } \Sigma)(7)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \langle I, I \rangle$, and
 - (ii) the result sort of $o = I$.
- (18) Suppose $o = (\text{the connectives of } \Sigma)(8)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \langle I, I \rangle$, and
 - (ii) the result sort of $o = I$.
- (19) Suppose $o = (\text{the connectives of } \Sigma)(9)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \langle I, I \rangle$, and
 - (ii) the result sort of $o = I$.
- (20) Suppose $o = (\text{the connectives of } \Sigma)(10)(\in (\text{the carrier' of } \Sigma))$. Then
- (i) $\text{Arity}(o) = \langle I, I \rangle$, and
 - (ii) the result sort of $o = \text{the boolean sort of } \Sigma$.
- (21) Let us consider a non empty non void many sorted signature Σ and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \emptyset$. Let us consider an algebra \mathfrak{A} over Σ . Then $\text{Args}(o, \mathfrak{A}) = \{\emptyset\}$.
- (22) Let us consider a non empty non void many sorted signature Σ , a sort symbol a of Σ , and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \langle a \rangle$. Let us consider an algebra \mathfrak{A} over Σ . Then $\text{Args}(o, \mathfrak{A}) = \prod \langle (\text{the sorts of } \mathfrak{A})(a) \rangle$.
- (23) Let us consider a non empty non void many sorted signature Σ , sort symbols a, b of Σ , and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \langle a, b \rangle$. Let us consider an algebra \mathfrak{A} over Σ . Then $\text{Args}(o, \mathfrak{A}) = \prod \langle (\text{the sorts of } \mathfrak{A})(a), (\text{the sorts of } \mathfrak{A})(b) \rangle$.
- (24) Let us consider a non empty non void many sorted signature Σ , sort symbols a, b, c of Σ , and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \langle a, b, c \rangle$. Let us consider an algebra \mathfrak{A} over Σ . Then $\text{Args}(o, \mathfrak{A}) = \prod \langle (\text{the sorts of } \mathfrak{A})(a), (\text{the sorts of } \mathfrak{A})(b), (\text{the sorts of } \mathfrak{A})(c) \rangle$.
- (25) Let us consider a non empty non void many sorted signature Σ , non-empty algebras $\mathfrak{A}, \mathfrak{B}$ over Σ , a sort symbol s of Σ , an element a of \mathfrak{A} from s , a many sorted function h from \mathfrak{A} into \mathfrak{B} , and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \langle s \rangle$. Let us consider an element p of $\text{Args}(o, \mathfrak{A})$. If $p = \langle a \rangle$, then $h\#p = \langle h(s)(a) \rangle$.

- (26) Let us consider a non empty non void many sorted signature Σ , non-empty algebras \mathfrak{A} , \mathfrak{B} over Σ , sort symbols s_1, s_2 of Σ , an element a of \mathfrak{A} from s_1 , an element b of \mathfrak{A} from s_2 , a many sorted function h from \mathfrak{A} into \mathfrak{B} , and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \langle s_1, s_2 \rangle$. Let us consider an element p of $\text{Args}(o, \mathfrak{A})$. Suppose $p = \langle a, b \rangle$. Then $h\#p = \langle h(s_1)(a), h(s_2)(b) \rangle$.
- (27) Let us consider a non empty non void many sorted signature Σ , non-empty algebras \mathfrak{A} , \mathfrak{B} over Σ , sort symbols s_1, s_2, s_3 of Σ , an element a of \mathfrak{A} from s_1 , an element b of \mathfrak{A} from s_2 , an element c of \mathfrak{A} from s_3 , a many sorted function h from \mathfrak{A} into \mathfrak{B} , and an operation symbol o of Σ . Suppose $\text{Arity}(o) = \langle s_1, s_2, s_3 \rangle$. Let us consider an element p of $\text{Args}(o, \mathfrak{A})$. Suppose $p = \langle a, b, c \rangle$. Then $h\#p = \langle h(s_1)(a), h(s_2)(b), h(s_3)(c) \rangle$.

Let us consider a many sorted function h from \mathfrak{T} into \mathfrak{C} , a sort symbol a of Σ , and an element τ of \mathfrak{T} from a . Now we state the propositions:

- (28) If h is a homomorphism of \mathfrak{T} into \mathfrak{C} , then τ value at $(\mathfrak{C}, h \upharpoonright \text{FreeGenerator}(\mathfrak{T})) = h(a)(\tau)$.
- (29) Suppose h is a homomorphism of \mathfrak{T} into \mathfrak{C} and $s = h \upharpoonright$ the generators of G . Then τ value at $(\mathfrak{C}, s) = h(a)(\tau)$.
- (30) $\text{true}_{\mathfrak{T}}$ value at $(\mathfrak{C}, s) = \text{true}$. The theorem is a consequence of (11) and (21).
- (31) Let us consider an element τ of \mathfrak{T} from the boolean sort of Σ . Then $\neg\tau$ value at $(\mathfrak{C}, s) = \neg(\tau$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (29), (12), (22), and (25).
- (32) Let us consider a boolean set a and an element τ of \mathfrak{T} from the boolean sort of Σ . Then $\neg\tau$ value at $(\mathfrak{C}, s) = \neg a$ if and only if τ value at $(\mathfrak{C}, s) = a$. The theorem is a consequence of (31).
- (33) Let us consider an element a of \mathfrak{C} from the boolean sort of Σ and a boolean set x . Then $\neg a = \neg x$ if and only if $a = x$.
- (34) $\text{false}_{\mathfrak{T}}$ value at $(\mathfrak{C}, s) = \text{false}$. The theorem is a consequence of (31) and (30).
- (35) Let us consider elements τ_1, τ_2 of \mathfrak{T} from the boolean sort of Σ . Then $(\tau_1 \wedge \tau_2)$ value at $(\mathfrak{C}, s) = (\tau_1$ value at $(\mathfrak{C}, s)) \wedge (\tau_2$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (29), (13), (23), and (26).
- (36) $0_{\mathfrak{T}}^I$ value at $(\mathfrak{C}, s) = 0$. The theorem is a consequence of (14) and (21).
- (37) $1_{\mathfrak{T}}^I$ value at $(\mathfrak{C}, s) = 1$. The theorem is a consequence of (15) and (21).
- (38) $(-\tau)$ value at $(\mathfrak{C}, s) = -\tau$ value at (\mathfrak{C}, s) . The theorem is a consequence of (16), (22), and (25).
- (39) $(\tau_1 + \tau_2)$ value at $(\mathfrak{C}, s) = \tau_1$ value at $(\mathfrak{C}, s) + \tau_2$ value at (\mathfrak{C}, s) . The theorem is a consequence of (17), (23), and (26).
- (40) $2_{\mathfrak{T}}^I$ value at $(\mathfrak{C}, s) = 2$. The theorem is a consequence of (37) and (39).

- (41) $(\tau_1 - \tau_2)$ value at $(\mathfrak{C}, s) = \tau_1$ value at $(\mathfrak{C}, s) - \tau_2$ value at (\mathfrak{C}, s) . The theorem is a consequence of (39) and (38).
- (42) $(\tau_1 \cdot \tau_2)$ value at $(\mathfrak{C}, s) = (\tau_1$ value at $(\mathfrak{C}, s)) \cdot (\tau_2$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (29), (18), (23), and (26).
- (43) $(\tau_1 \text{ div } \tau_2)$ value at $(\mathfrak{C}, s) = \tau_1$ value at $(\mathfrak{C}, s) \text{ div } \tau_2$ value at (\mathfrak{C}, s) . The theorem is a consequence of (19), (23), and (26).
- (44) $(\tau_1 \text{ mod } \tau_2)$ value at $(\mathfrak{C}, s) = \tau_1$ value at $(\mathfrak{C}, s) \text{ mod } \tau_2$ value at (\mathfrak{C}, s) . The theorem is a consequence of (41), (42), and (43).
- (45) $\text{leq}(\tau_1, \tau_2)$ value at $(\mathfrak{C}, s) = \text{leq}(\tau_1$ value at $(\mathfrak{C}, s), \tau_2$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (20), (23), and (26).
- (46) $\text{true}_{\mathfrak{T}}$ value at $(\mathfrak{C}, u) = \text{true}$. The theorem is a consequence of (11) and (21).
- (47) Let us consider an element τ of \mathfrak{T} from the boolean sort of Σ . Then $\neg\tau$ value at $(\mathfrak{C}, u) = \neg(\tau$ value at $(\mathfrak{C}, u))$. The theorem is a consequence of (28), (12), (22), and (25).
- (48) Let us consider a boolean set a and an element τ of \mathfrak{T} from the boolean sort of Σ . Then $\neg\tau$ value at $(\mathfrak{C}, u) = \neg a$ if and only if τ value at $(\mathfrak{C}, u) = a$. The theorem is a consequence of (47).
- (49) $\text{false}_{\mathfrak{T}}$ value at $(\mathfrak{C}, u) = \text{false}$. The theorem is a consequence of (47) and (46).
- (50) Let us consider elements τ_1, τ_2 of \mathfrak{T} from the boolean sort of Σ . Then $(\tau_1 \wedge \tau_2)$ value at $(\mathfrak{C}, u) = (\tau_1$ value at $(\mathfrak{C}, u)) \wedge (\tau_2$ value at $(\mathfrak{C}, u))$. The theorem is a consequence of (28), (13), (23), and (26).
- (51) $0_{\mathfrak{T}}$ value at $(\mathfrak{C}, u) = 0$. The theorem is a consequence of (14) and (21).
- (52) $1_{\mathfrak{T}}$ value at $(\mathfrak{C}, u) = 1$. The theorem is a consequence of (15) and (21).
- (53) $(-\tau)$ value at $(\mathfrak{C}, u) = -\tau$ value at (\mathfrak{C}, u) . The theorem is a consequence of (16), (22), and (25).
- (54) $(\tau_1 + \tau_2)$ value at $(\mathfrak{C}, u) = \tau_1$ value at $(\mathfrak{C}, u) + \tau_2$ value at (\mathfrak{C}, u) . The theorem is a consequence of (17), (23), and (26).
- (55) $2_{\mathfrak{T}}$ value at $(\mathfrak{C}, u) = 2$. The theorem is a consequence of (52) and (54).
- (56) $(\tau_1 - \tau_2)$ value at $(\mathfrak{C}, u) = \tau_1$ value at $(\mathfrak{C}, u) - \tau_2$ value at (\mathfrak{C}, u) . The theorem is a consequence of (54) and (53).
- (57) $(\tau_1 \cdot \tau_2)$ value at $(\mathfrak{C}, u) = (\tau_1$ value at $(\mathfrak{C}, u)) \cdot (\tau_2$ value at $(\mathfrak{C}, u))$. The theorem is a consequence of (28), (18), (23), and (26).
- (58) $(\tau_1 \text{ div } \tau_2)$ value at $(\mathfrak{C}, u) = \tau_1$ value at $(\mathfrak{C}, u) \text{ div } \tau_2$ value at (\mathfrak{C}, u) . The theorem is a consequence of (19), (23), and (26).
- (59) $(\tau_1 \text{ mod } \tau_2)$ value at $(\mathfrak{C}, u) = \tau_1$ value at $(\mathfrak{C}, u) \text{ mod } \tau_2$ value at (\mathfrak{C}, u) . The theorem is a consequence of (56), (57), and (58).

- (60) $\text{leq}(\tau_1, \tau_2)$ value at $(\mathfrak{C}, u) = \text{leq}(\tau_1 \text{ value at } (\mathfrak{C}, u), \tau_2 \text{ value at } (\mathfrak{C}, u))$.
The theorem is a consequence of (20), (23), and (26).
- (61) Let us consider a sort symbol a of Σ and an element x of (the generators of G)(a). Then $\text{@}x$ value at $(\mathfrak{C}, s) = s(a)(x)$. The theorem is a consequence of (29).
- (62) Let us consider a sort symbol a of Σ , a pure element x of (the generators of G)(a), and a many sorted function u from $\text{FreeGenerator}(\mathfrak{T})$ into the sorts of \mathfrak{C} . Then $\text{@}x$ value at $(\mathfrak{C}, u) = u(a)(x)$.

Let us consider integers i, j and elements a, b of \mathfrak{C} from I . Now we state the propositions:

- (63) If $a = i$ and $b = j$, then $a - b = i - j$.
- (64) If $a = i$ and $b = j$ and $j \neq 0$, then $a \bmod b = i \bmod j$.
- (65) Suppose G is \mathfrak{C} -supported and $f \in \mathfrak{C}\text{-Execution}_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$. Then let us consider a sort symbol a of Σ , a pure element x of (the generators of G)(a), and an element τ of \mathfrak{T} from a . Then
- (i) $f(s, x := \mathfrak{A}\tau)(a)(x) = \tau$ value at (\mathfrak{C}, s) , and
 - (ii) for every pure element z of (the generators of G)(a) such that $z \neq x$ holds $f(s, x := \mathfrak{A}\tau)(a)(z) = s(a)(z)$, and
 - (iii) for every sort symbol b of Σ such that $a \neq b$ for every pure element z of (the generators of G)(b), $f(s, x := \mathfrak{A}\tau)(b)(z) = s(b)(z)$.
- (66) Suppose G is \mathfrak{C} -supported and $f \in \mathfrak{C}\text{-Execution}_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$. Then
- (i) τ_1 value at $(\mathfrak{C}, s) < \tau_2$ value at (\mathfrak{C}, s) iff $f(s, b \text{gt}(\tau_2, \tau_1, \mathfrak{A})) \in \text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G), and
 - (ii) τ_1 value at $(\mathfrak{C}, s) \leq \tau_2$ value at (\mathfrak{C}, s) iff $f(s, b \text{leq}(\tau_1, \tau_2, \mathfrak{A})) \in \text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G), and
 - (iii) for every x , $f(s, b \text{gt}(\tau_1, \tau_2, \mathfrak{A}))(I)(x) = s(I)(x)$ and $f(s, b \text{leq}(\tau_1, \tau_2, \mathfrak{A}))(I)(x) = s(I)(x)$, and
 - (iv) for every pure element c of (the generators of G)((the boolean sort of Σ)) such that $c \neq b$ holds $f(s, b \text{gt}(\tau_1, \tau_2, \mathfrak{A}))((\text{the boolean sort of } \Sigma))(c) = s((\text{the boolean sort of } \Sigma))(c)$ and $f(s, b \text{leq}(\tau_1, \tau_2, \mathfrak{A}))((\text{the boolean sort of } \Sigma))(c) = s((\text{the boolean sort of } \Sigma))(c)$.

The theorem is a consequence of (31), (45), and (33).

Let i, j be real numbers and a, b be boolean sets. One can verify that $(i > j \rightarrow a, b)$ is boolean.

Now we state the proposition:

- (67) Suppose G is \mathfrak{C} -supported and $f \in \mathfrak{C}\text{-Execution}_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$. Then
- (i) $f(s, \tau \text{ is odd}(b, \mathfrak{A}))((\text{the boolean sort of } \Sigma))(b) = \tau$ value at $(\mathfrak{C}, s) \bmod 2$, and

- (ii) $f(s, \tau \text{ is even}(b, \mathfrak{A}))(\text{the boolean sort of } \Sigma)(b) = (\tau \text{ value at } (\mathfrak{C}, s) + 1) \bmod 2$, and
- (iii) for every z , $f(s, \tau \text{ is odd}(b, \mathfrak{A}))(I)(z) = s(I)(z)$ and $f(s, \tau \text{ is even}(b, \mathfrak{A}))(I)(z) = s(I)(z)$.

The theorem is a consequence of (36), (40), (64), (31), (45), (44), and (1).

Let us consider Σ , X , \mathfrak{T} , G , and \mathfrak{A} . We say that \mathfrak{A} is elementary if and only if

(Def. 13) $\text{rng the assignments of } \mathfrak{A} \subseteq \text{ElementaryInstructions}_{\mathfrak{A}}$.

Now we state the proposition:

- (68) Suppose \mathfrak{A} is elementary. Then let us consider a sort symbol a of Σ , an element x of (the generators of G)(a), and an element τ of \mathfrak{T} from a . Then $x :=_{\mathfrak{A}} \tau \in \text{ElementaryInstructions}_{\mathfrak{A}}$.

Let us consider Σ , X , \mathfrak{T} , and G . One can verify that there exists a strict if-while algebra over the generators of G which is elementary.

Let \mathfrak{A} be an elementary if-while algebra over the generators of G , a be a sort symbol of Σ , x be an element of (the generators of G)(a), and τ be an element of \mathfrak{T} from a . Let us observe that $x :=_{\mathfrak{A}} \tau$ is absolutely-terminating.

Now let Γ denotes the program

```

y :=\mathfrak{A}} 1\mathfrak{T}};
while bgt(\mathfrak{A}m, 0\mathfrak{T}}, \mathfrak{A}) do
  if \mathfrak{A}m is odd(b, \mathfrak{A}) then
    y :=\mathfrak{A}} \mathfrak{A}y · \mathfrak{A}x
  fi;
  m :=\mathfrak{A}} \mathfrak{A}m div 2\mathfrak{T}};
  x :=\mathfrak{A}} \mathfrak{A}x · \mathfrak{A}x
done
```

Then we state the propositions:

- (69) Let us consider an elementary if-while algebra \mathfrak{A} over the generators of G and an execution function f of \mathfrak{A} over \mathfrak{C} -States(the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Suppose
 - (i) G is \mathfrak{C} -supported, and
 - (ii) $f \in \mathfrak{C}$ -Execution $_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$, and
 - (iii) there exists a function d such that $d(x) = 1$ and $d(y) = 2$ and $d(m) = 3$.

Then Γ is terminating w.r.t. f and $\{s : s(I)(m) \geq 0\}$. The theorem is a consequence of (66), (36), (61), (65), (40), and (43). PROOF: Set $ST = \mathfrak{C}$ -States(the generators of G). Set $TV = \text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Set $P = \{s : s(I)(m) \geq 0\}$. Set $W = \text{bgt}(\sup{\mathfrak{A}}m, 0_{\mathfrak{T}}, \mathfrak{A})$. Define \mathcal{F} (element of ST) = $\$_1(I)(m) (\in \mathbb{N})$. Define \mathcal{R} [element of ST] $\equiv \$_1(I)(m) >$

0. Set $K = \text{if } @m \text{ is odd}(b, \mathfrak{A}) \text{ then}(y := \mathfrak{A}(@y \cdot @x))$.

Set $J = (K; m := \mathfrak{A}(@m \text{ div } 2_{\mathfrak{T}}^I)); x := \mathfrak{A}(@x \cdot @x)$. P is invariant w.r.t. W and f . For every element s of ST such that $s \in P$ and $f(f(s, J), W) \in TV$ holds $f(s, J) \in P$. P is invariant w.r.t. $y := \mathfrak{A}(1_{\mathfrak{T}}^I)$ and f . For every s such that $f(s, W) \in P$ holds iteration of f started in $J; W$ terminates w.r.t. $f(s, W)$. \square

- (70) Suppose G is \mathfrak{C} -supported and there exists a function d such that $d(b) = 0$ and $d(x) = 1$ and $d(y) = 2$ and $d(m) = 3$. Then let us consider an element s of \mathfrak{C} -States(the generators of G) and a natural number n . Suppose $n = s(I)(m)$. If $f \in \mathfrak{C}$ -Execution $_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$, then $f(s, \Gamma)(I)(y) = s(I)(x)^n$. The theorem is a consequence of (65), (66), (36), (61), (37), (40), (43), (67), (10), and (42). PROOF: Set $\Sigma = \mathfrak{C}$ -States(the generators of G). Set $W = \mathfrak{T}$. Set $g = f$. Set $\mathfrak{T} = \text{States}_{b \neq \text{false}_{\mathfrak{C}}}(\text{the generators of } G)$. Set $s_0 = f(s, y := \mathfrak{A}(1_W^I))$. Define $\mathcal{R}[\text{element of } \Sigma] \equiv \$_1(I)(m) > 0$. Set $\mathfrak{C} = b \text{ gt}(@m, 0_W^I, \mathfrak{A})$. Define $\mathcal{P}[\text{element of } \Sigma] \equiv s(I)(x)^n = \$_1(I)(y) \cdot \$_1(I)(x)^{\$_1(I)(m)}$ and $\$_1(I)(m) \geq 0$. Define $\mathcal{F}(\text{element of } \Sigma) = \$_1(I)(m) (\in \mathbb{N})$. Set $I = \text{if } @m \text{ is odd}(b, \mathfrak{A}) \text{ then}(y := \mathfrak{A}(@y \cdot @x))$. Set $J = (I; m := \mathfrak{A}(@m \text{ div } 2_W^I)); x := \mathfrak{A}(@x \cdot @x)$. For every element s of Σ such that $\mathcal{P}[s]$ holds $\mathcal{P}[(g(s, \mathfrak{C}) \text{ qua element of } \Sigma)]$ and $g(s, \mathfrak{C}) \in \mathfrak{T}$ iff $\mathcal{R}[(g(s, \mathfrak{C}) \text{ qua element of } \Sigma)]$. Set $s_1 = g(s_0, \mathfrak{C})$. For every element s of Σ such that $\mathcal{R}[s]$ holds $\mathcal{R}[(g(s, J; \mathfrak{C}) \text{ qua element of } \Sigma)]$ iff $g(s, J; \mathfrak{C}) \in \mathfrak{T}$ and $\mathcal{F}((g(s, J; \mathfrak{C}) \text{ qua element of } \Sigma)) < \mathcal{F}(s)$. Set $q = s$. For every element s of Σ such that $\mathcal{P}[s]$ and $s \in \mathfrak{T}$ and $\mathcal{R}[s]$ holds $\mathcal{P}[(g(s, J) \text{ qua element of } \Sigma)]$. \square

2. CALCULATION OF MAXIMUM

Let X be a non empty set, f be a finite sequence of elements of X^ω , and x be a natural number. Let us observe that $f(x)$ is transfinite sequence-like finite function-like and relation-like.

Let us note that every finite sequence of elements of X^ω is function yielding.

Let i be a natural number, f be an i -based finite array, and a, x be sets. Note that $f + \cdot (a, x)$ is i -based finite and segmental.

Let X be a non empty set, f be an X -valued function, a be a set, and x be an element of X . Let us observe that $f + \cdot (a, x)$ is X -valued.

The scheme *Sch1* deals with a non empty set \mathcal{X} and a natural number j and a set \mathfrak{B} and a ternary functor \mathcal{F} yielding a set and a unary functor \mathfrak{A} yielding a set and states that

- (Sch. 1) There exists a finite sequence f of elements of \mathcal{X}^ω such that $\text{len } f = j$ and $f(1) = \mathfrak{B}$ or $j = 0$ and for every natural number i such that $1 \leq i < j$ holds $f(i + 1) = \mathcal{F}(f(i), i, \mathfrak{A}(i))$

provided

- for every 0-based finite array a of \mathcal{X} and for every natural number i such that $1 \leq i < j$ for every element x of \mathcal{X} , $\mathcal{F}(a, i, x)$ is a 0-based finite array of \mathcal{X} and
- \mathfrak{B} is a 0-based finite array of \mathcal{X} and
- for every natural number i such that $i < j$ holds $\mathfrak{A}(i) \in \mathcal{X}$.

Now we state the propositions:

- (71) Let us consider a non empty non void boolean signature Σ with arrays of type 1 with connectives from 11 and integers at 1, sets J, L , and a sort symbol K of Σ . Suppose (the connectives of Σ)(11) is of type $\langle J, L \rangle \rightarrow K$. Then
- (i) $J =$ the array sort of Σ , and
 - (ii) for every integer sort symbol I of Σ , the array sort of $\Sigma \neq I$.
- (72) Let us consider a 1-1-connectives 11-array correct boolean correct non empty non void boolean signature Σ with integers with connectives from 4 and the sort at 1 and arrays of type 1 with connectives from 11 and integers at 1, an integer sort symbol I of Σ , a boolean correct non-empty algebra \mathfrak{A} over Σ with integers with connectives from 4 and the sort at 1 and arrays of type 1 with connectives from 11 and integers at 1, and elements a, b of \mathfrak{A} from I . If $a = 0$, then $\text{init.array}(a, b) = \emptyset$.
- (73) Let us consider an 11-array correct boolean correct non empty non void boolean signature Σ with arrays of type 1 with connectives from 11 and integers at 1 and an integer sort symbol I of Σ . Then
- (i) the array sort of $\Sigma \neq I$, and
 - (ii) (the connectives of Σ)(11) is of type $\langle \text{the array sort of } \Sigma, I \rangle \rightarrow I$, and
 - (iii) (the connectives of Σ)(11 + 1) is of type $\langle \text{the array sort of } \Sigma, I, I \rangle \rightarrow \text{the array sort of } \Sigma$, and
 - (iv) (the connectives of Σ)(11 + 2) is of type $\langle \text{the array sort of } \Sigma \rangle \rightarrow I$, and
 - (v) (the connectives of Σ)(11 + 3) is of type $\langle I, I \rangle \rightarrow \text{the array sort of } \Sigma$.
- (74) Let us consider a 1-1-connectives 11-array correct boolean correct non empty non void boolean signature Σ with arrays of type 1 with connectives from 11 and integers at 1 and integers with connectives from 4 and the sort at 1, an integer sort symbol I of Σ , and a boolean correct non-empty algebra \mathfrak{A} over Σ with arrays of type 1 with connectives from 11 and integers at 1 and integers with connectives from 4 and the sort at 1. Then
- (i) (the sorts of \mathfrak{A})(the array sort of Σ) = \mathbb{Z}^ω , and

- (ii) for every elements i, j of \mathfrak{A} from I such that i is a non negative integer holds $\text{init.array}(i, j) = i \mapsto j$, and
- (iii) for every element a of (the sorts of \mathfrak{A})(the array sort of Σ), $\text{length}_I a = \bar{a}$ and for every element i of \mathfrak{A} from I and for every function f such that $f = a$ and $i \in \text{dom } f$ holds $a(i) = f(i)$ and for every element x of \mathfrak{A} from I , $a_{i \leftarrow x} = f + \cdot (i, x)$.

The theorem is a consequence of (71).

Let a be a 0-based finite array. Observe that $\text{length } a$ is finite.

Let Σ be a 1-1-connectives 11-array correct boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1 and arrays of type 1 with connectives from 11 and integers at 1 and \mathfrak{A} be a boolean correct non-empty algebra over Σ with arrays of type 1 with connectives from 11 and integers at 1 and integers with connectives from 4 and the sort at 1. Observe that every non-empty subalgebra of \mathfrak{A} has arrays of type 1 with connectives from 11 and integers at 1.

Let \mathfrak{A} be a non-empty algebra over Σ . We say that \mathfrak{A} is integer array if and only if

- (Def. 14) There exists an image \mathfrak{C} of \mathfrak{A} such that \mathfrak{C} is a boolean correct algebra over Σ with integers with connectives from 4 and the sort at 1 and arrays of type 1 with connectives from 11 and integers at 1.

Let X be a non-empty many sorted set indexed by the carrier of Σ . One can verify that $\mathfrak{F}_\Sigma(X)$ is integer array as a non-empty algebra over Σ .

Note that every non-empty algebra over Σ which is integer array is also integer.

One can check that there exists an including Σ -terms over X non-empty strict free variable algebra over Σ which is vf-free and integer array.

One can check that there exists a non-empty algebra over Σ which is integer array.

Let \mathfrak{A} be an integer array non-empty algebra over Σ . Observe that there exists a boolean correct image of \mathfrak{A} which has integers with connectives from 4 and the sort at 1 and arrays of type 1 with connectives from 11 and integers at 1.

In this paper Σ denotes a 1-1-connectives 11-array correct boolean correct non empty non void boolean signature with integers with connectives from 4 and the sort at 1 and arrays of type 1 with connectives from 11 and integers at 1, X denotes a non-empty many sorted set indexed by the carrier of Σ , \mathfrak{T} denotes a vf-free including Σ -terms over X integer array non-empty free variable algebra over Σ , \mathfrak{C} denotes a boolean correct non-empty image of \mathfrak{T} with arrays of type 1 with connectives from 11 and integers at 1 and integers with connectives from 4 and the sort at 1, G denotes a basic generator system over Σ , X , and \mathfrak{T} , \mathfrak{A} denotes a if-while algebra over the generators of G , I denotes an integer sort symbol of Σ , x, y, m, i denote pure elements of (the generators of G)(I), M, N

denote pure elements of (the generators of G)(the array sort of Σ), b denotes a pure element of (the generators of G)(the boolean sort of Σ), and s, s_1 denote elements of \mathfrak{C} -States(the generators of G).

Let us consider Σ . Let \mathfrak{A} be a boolean correct non-empty algebra over Σ with arrays of type 1 with connectives from 11 and integers at 1. Observe that every element of (the sorts of \mathfrak{A})(the array sort of Σ) is relation-like and function-like.

Note that every element of (the sorts of \mathfrak{A})(the array sort of Σ) is finite and transfinite sequence-like.

Let us consider an operation symbol o of Σ . Now we state the propositions:

- (75) Suppose $o =$ (the connectives of Σ)(11)(\in (the carrier' of Σ)). Then
- (i) Arity(o) = \langle the array sort of Σ, I \rangle , and
 - (ii) the result sort of $o = I$.
- (76) Suppose $o =$ (the connectives of Σ)(12)(\in (the carrier' of Σ)). Then
- (i) Arity(o) = \langle the array sort of Σ, I, I \rangle , and
 - (ii) the result sort of $o =$ the array sort of Σ .
- (77) Suppose $o =$ (the connectives of Σ)(13)(\in (the carrier' of Σ)). Then
- (i) Arity(o) = \langle the array sort of Σ \rangle , and
 - (ii) the result sort of $o = I$.
- (78) Suppose $o =$ (the connectives of Σ)(14)(\in (the carrier' of Σ)). Then
- (i) Arity(o) = $\langle I, I \rangle$, and
 - (ii) the result sort of $o =$ the array sort of Σ .
- (79) Let us consider an element τ of \mathfrak{F} from the array sort of Σ and an element τ_1 of \mathfrak{F} from I .
Then $\tau(\tau_1)$ value at $(\mathfrak{C}, s) = (\tau$ value at $(\mathfrak{C}, s))(\tau_1$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (29), (75), (23), and (26).
- (80) Let us consider an element τ of \mathfrak{F} from the array sort of Σ and elements τ_1, τ_2 of \mathfrak{F} from I . Then $\tau_{\tau_1 \leftarrow \tau_2}$ value at $(\mathfrak{C}, s) = (\tau$ value at $(\mathfrak{C}, s))_{\tau_1$ value at $(\mathfrak{C}, s) \leftarrow \tau_2$ value at $(\mathfrak{C}, s)}$. The theorem is a consequence of (29), (76), (24), and (27).
- (81) Let us consider an element τ of \mathfrak{F} from the array sort of Σ . Then $\text{length}_I \tau$ value at $(\mathfrak{C}, s) = \text{length}_I(\tau$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (29), (77), (22), and (25).
- (82) Let us consider elements τ_1, τ_2 of \mathfrak{F} from I . Then $\text{init.array}(\tau_1, \tau_2)$ value at $(\mathfrak{C}, s) = \text{init.array}(\tau_1$ value at $(\mathfrak{C}, s), \tau_2$ value at $(\mathfrak{C}, s))$. The theorem is a consequence of (29), (78), (23), and (26).

In the sequel u denotes a many sorted function from $\text{FreeGenerator}(\mathfrak{F})$ into the sorts of \mathfrak{C} .

Now we state the propositions:

- (83) Let us consider an element τ of \mathfrak{T} from the array sort of Σ and an element τ_1 of \mathfrak{T} from I .
Then $\tau(\tau_1)$ value at $(\mathfrak{C}, u) = (\tau \text{ value at } (\mathfrak{C}, u))(\tau_1 \text{ value at } (\mathfrak{C}, u))$. The theorem is a consequence of (28), (75), (23), and (26).
- (84) Let us consider an element τ of \mathfrak{T} from the array sort of Σ and elements τ_1, τ_2 of \mathfrak{T} from I .
Then $\tau_{\tau_1 \leftarrow \tau_2}$ value at $(\mathfrak{C}, u) = (\tau \text{ value at } (\mathfrak{C}, u))_{\tau_1 \text{ value at } (\mathfrak{C}, u) \leftarrow \tau_2 \text{ value at } (\mathfrak{C}, u)}$.
The theorem is a consequence of (28), (76), (24), and (27).
- (85) Let us consider an element τ of \mathfrak{T} from the array sort of Σ . Then $\text{length}_I \tau$ value at $(\mathfrak{C}, u) = \text{length}_I(\tau \text{ value at } (\mathfrak{C}, u))$. The theorem is a consequence of (28), (77), (22), and (25).
- (86) Let us consider elements τ_1, τ_2 of \mathfrak{T} from I . Then $\text{init.array}(\tau_1, \tau_2)$ value at $(\mathfrak{C}, u) = \text{init.array}(\tau_1 \text{ value at } (\mathfrak{C}, u), \tau_2 \text{ value at } (\mathfrak{C}, u))$. The theorem is a consequence of (28), (78), (23), and (26).

Let us consider Σ, X, \mathfrak{T} , and I . Let i be an integer. The functor $i_{\mathfrak{T}}^I$ yielding an element of \mathfrak{T} from I is defined by

(Def. 15) There exists a function f from \mathbb{Z} into (the sorts of $\mathfrak{T})(I)$ such that

- (i) $it = f(i)$, and
- (ii) $f(0) = 0_{\mathfrak{T}}^I$, and
- (iii) for every natural number j and for every element τ of \mathfrak{T} from I such that $f(j) = \tau$ holds $f(j + 1) = \tau + 1_{\mathfrak{T}}^I$ and $f(-(j + 1)) = -(\tau + 1_{\mathfrak{T}}^I)$.

Now we state the propositions:

- (87) $0_{\mathfrak{T}}^I = 0_{\mathfrak{T}}^I$.
- (88) Let us consider a natural number n . Then
 - (i) $(n + 1)_{\mathfrak{T}}^I = n_{\mathfrak{T}}^I + 1_{\mathfrak{T}}^I$, and
 - (ii) $-(n + 1)_{\mathfrak{T}}^I = -(n + 1)_{\mathfrak{T}}^I$.
- (89) $1_{\mathfrak{T}}^I = 0_{\mathfrak{T}}^I + 1_{\mathfrak{T}}^I$. The theorem is a consequence of (88) and (87).
- (90) Let us consider an integer i . Then $i_{\mathfrak{T}}^I$ value at $(\mathfrak{C}, s) = i$. The theorem is a consequence of (87), (36), (37), (88), (39), and (38).

Let us consider $\Sigma, X, \mathfrak{T}, G, I$, and M . Let i be an integer. The functor $M(i, I)$ yielding an element of \mathfrak{T} from I is defined by the term

(Def. 16) $({}^{\circ}M)(i_{\mathfrak{T}}^I)$.

Let us consider \mathfrak{C} and s . Note that $s(\text{the array sort of } \Sigma)(M)$ is function-like and relation-like.

Note that $s(\text{the array sort of } \Sigma)(M)$ is finite transfinite sequence-like and \mathbb{Z} -valued.

Observe that $\text{rng}(s(\text{the array sort of } \Sigma)(M))$ is finite and integer-membered.

Let us consider an integer j . Now we state the propositions:

- (91) Suppose $j \in \text{dom}(s(\text{the array sort of } \Sigma)(M))$ and $M(j, I) \in (\text{the generators of } G)(I)$. Then $s(\text{the array sort of } \Sigma)(M)(j) = s(I)(M(j, I))$.
- (92) Suppose $j \in \text{dom}(s(\text{the array sort of } \Sigma)(M))$ and $({}^{\textcircled{a}}M)({}^{\textcircled{a}}i) \in (\text{the generators of } G)(I)$ and $j = {}^{\textcircled{a}}i$ value at (\mathfrak{C}, s) . Then $(s(\text{the array sort of } \Sigma)(M))({}^{\textcircled{a}}i \text{ value at } (\mathfrak{C}, s)) = s(I)((({}^{\textcircled{a}}M)({}^{\textcircled{a}}i)))$.

Let X be a non empty set. One can verify that X^ω is infinite.

Now we state the propositions:

- (93) Now let Γ denotes the program

```

 $m := \mathfrak{A} 0_{\frac{I}{\mathfrak{X}}};$ 
for  $i := \mathfrak{A} 1_{\frac{I}{\mathfrak{X}}}$  until  $\text{bgt}(\text{length}_I {}^{\textcircled{a}}M, {}^{\textcircled{a}}i, \mathfrak{A})$  step  $i := \mathfrak{A} {}^{\textcircled{a}}i + 1_{\frac{I}{\mathfrak{X}}}$ 
do
  if  $\text{bgt}(({}^{\textcircled{a}}M)({}^{\textcircled{a}}i), ({}^{\textcircled{a}}M)({}^{\textcircled{a}}m), \mathfrak{A})$  then
     $m := \mathfrak{A} {}^{\textcircled{a}}i$ 
  fi
done

```

Let us consider an execution function f of \mathfrak{A} over \mathfrak{C} -States (the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Suppose

- (i) $f \in \mathfrak{C}\text{-Execution}_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$, and
- (ii) G is \mathfrak{C} -supported, and
- (iii) $i \neq m$, and
- (iv) $s(\text{the array sort of } \Sigma)(M) \neq \emptyset$.

Let us consider a natural number n . Suppose $f(s, \Gamma)(I)(m) = n$. Let us consider a non empty finite integer-membered set X . Suppose $X = \text{rng}(s(\text{the array sort of } \Sigma)(M))$. Then $M(n, I)$ value at $(\mathfrak{C}, s) = \max X$. The theorem is a consequence of (65), (36), (37), (74), (71), (66), (81), (61), (39), (79), and (90). **PROOF:** Set $ST = \mathfrak{C}\text{-States}(\text{the generators of } G)$. Define $\mathcal{R}[\text{element of } ST] \equiv s(\text{the array sort of } \Sigma)(M) = \$_1(\text{the array sort of } \Sigma)(M)$. Reconsider $sm = s$ as a many sorted function from the generators of G into the sorts of \mathfrak{C} . Reconsider $z = sm(\text{the array sort of } \Sigma)(M)$ as a 0-based finite array of \mathbb{Z} . Define $\mathcal{P}[\text{element of } ST] \equiv \mathcal{R}[\$_1]$ and $\$_1(I)(i), \$_1(I)(m) \in \mathbb{N}$ and $\$_1(I)(i) \leq \text{len } z$ and $\$_1(I)(m) < \$_1(I)(i)$ and $\$_1(I)(m) < \text{len } z$ and for every integer mx such that $mx = \$_1(I)(m)$ for every natural number j such that $j < \$_1(I)(i)$ holds $z(j) \leq z(mx)$. Define $\mathcal{Q}[\text{element of } ST] \equiv \mathcal{R}[\$_1]$ and $\$_1(I)(i) < \text{length}_I {}^{\textcircled{a}}M$ value at (\mathfrak{C}, s) . Set $s_0 = s$. Set $s_1 = f(s, m := \mathfrak{A}(0_{\frac{I}{\mathfrak{X}}}))$. Set $s_2 = f(s_1, i := \mathfrak{A}(1_{\frac{I}{\mathfrak{X}}}))$. Consider $J1, K1, L1$ being elements of Σ such that $L1 = 1$ and $K1 = 1$ and $J1 \neq L1$ and $J1 \neq K1$ and (the connectives of Σ)(11) is of type $\langle J1, K1 \rangle \rightarrow L1$ and (the connectives of Σ)(11 + 1) is of type $\langle J1, K1, L1 \rangle \rightarrow J1$ and (the connectives of Σ)(11 + 2) is of type $\langle J1 \rangle \rightarrow K1$ and

(the connectives of Σ)(11 + 3) is of type $\langle K1, L1 \rangle \rightarrow J1$. $\mathcal{P}[s_2]$. Define \mathcal{F} (element of ST) = $(\text{len}(s_0(\text{the array sort of } \Sigma)(M)) - \$_1(I)(i))(\in \mathbb{N})$. $f(s_2, W) \in TV$ iff $\mathcal{Q}[f(s_2, W)]$. Now let Γ denotes the program

```
J;
K;
W
```

For every element s of ST such that $\mathcal{Q}[s]$ holds $\mathcal{Q}[f(s, \Gamma)]$ iff $f(s, \Gamma) \in TV$ and $\mathcal{F}(f(s, \Gamma)) < \mathcal{F}(s)$. For every element s of ST such that $\mathcal{P}[s]$ and $s \in TV$ and $\mathcal{Q}[s]$ holds $\mathcal{P}[f(s, J; K)]$. For every element s of ST such that $\mathcal{P}[s]$ holds $\mathcal{P}[f(s, W)]$ and $f(s, W) \in TV$ iff $\mathcal{Q}[f(s, W)]$. $M(n, I)$ value at (\mathfrak{C}, s) is an upper bound of X . For every upper bound x of X , $M(n, I)$ value at $(\mathfrak{C}, s) \leq x$. \square

(94) Now let Γ denotes the program

```
J;
i :=  $\mathfrak{A}^{\textcircled{i}} + 1 \frac{I}{\mathfrak{X}}$ 
```

Now let Δ denotes the program

```
for i :=  $\mathfrak{A}^{\textcircled{i}} \tau_0$  until bgt( $\tau_1, \textcircled{i}, \mathfrak{A}$ ) step i :=  $\mathfrak{A}^{\textcircled{i}} + 1 \frac{I}{\mathfrak{X}}$  do
  J
done
```

Let us consider an elementary if-while algebra \mathfrak{A} over the generators of G and an execution function f of \mathfrak{A} over \mathfrak{C} -States(the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Suppose

- (i) $f \in \mathfrak{C}\text{-Execution}_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$, and
- (ii) G is \mathfrak{C} -supported.

Let us consider elements τ_0, τ_1 of \mathfrak{T} from I , an algorithm J of \mathfrak{A} , and a set P . Suppose

- (iii) P is invariant w.r.t. $i := \mathfrak{A}^{\textcircled{i}} \tau_0$ and f , invariant w.r.t. $\text{bgt}(\tau_1, \textcircled{i}, \mathfrak{A})$ and f , invariant w.r.t. $i := \mathfrak{A}^{\textcircled{i}} + 1 \frac{I}{\mathfrak{X}}$ and f , and invariant w.r.t. J and f , and
- (iv) J is terminating w.r.t. f and P , and
- (v) for every s , $f(s, J)(I)(i) = s(I)(i)$ and $f(s, \text{bgt}(\tau_1, \textcircled{i}, \mathfrak{A}))(I)(i) = s(I)(i)$ and τ_1 value at $(\mathfrak{C}, f(s, \text{bgt}(\tau_1, \textcircled{i}, \mathfrak{A}))) = \tau_1$ value at (\mathfrak{C}, s) and τ_1 value at $(\mathfrak{C}, f(s, \Gamma)) = \tau_1$ value at (\mathfrak{C}, s) .

Then Δ is terminating w.r.t. f and P . The theorem is a consequence of (61), (66), (65), (39), and (37). PROOF: Set $W = \text{bgt}(\tau_1, \textcircled{i}, \mathfrak{A})$. Set $L = i := \mathfrak{A}^{\textcircled{i}} + 1 \frac{I}{\mathfrak{X}}$. Set $K = i := \mathfrak{A}^{\textcircled{i}} \tau_0$. Set $ST = \mathfrak{C}\text{-States}$ (the generators of G). Set $TV = \text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Now let Γ denotes the program

```
J;
L;
W
```

For every s such that $f(s, W) \in P$ holds iteration of f started in Γ terminates w.r.t. $f(s, W)$. \square

(95) Now let Γ denotes the program

```
m :=\mathfrak{A} 0I\mathfrak{Z};
for i :=\mathfrak{A} 1I\mathfrak{Z} until bgt(lengthI@M, @i, \mathfrak{A}) step i :=\mathfrak{A} @i + 1I\mathfrak{Z}
do
  if bgt((@M)(@i), (@M)(@m), \mathfrak{A}) then
    m :=\mathfrak{A} @i
  fi
done
```

Let us consider an elementary if-while algebra \mathfrak{A} over the generators of G and an execution function f of \mathfrak{A} over \mathfrak{C} -States (the generators of G) and States _{$b \neq \text{false}_{\mathfrak{C}}$} (the generators of G). Suppose

- (i) $f \in \mathfrak{C}$ -Execution _{$b \neq \text{false}_{\mathfrak{C}}$} (\mathfrak{A}), and
- (ii) G is \mathfrak{C} -supported, and
- (iii) $i \neq m$.

Then Γ is terminating w.r.t. f and $\{s : s(\text{the array sort of } \Sigma)(M) \neq \emptyset\}$. The theorem is a consequence of (74), (73), (65), (61), (81), and (94). PROOF: Set $J = m :=_{\mathfrak{A}} 0^I_{\mathfrak{Z}}}$. Set $K = i :=_{\mathfrak{A}} 1^I_{\mathfrak{Z}}}$. Set $W = \text{bgt}(\text{length}_I^{@M, @i, \mathfrak{A}})$. Set $L = i :=_{\mathfrak{A}} (@i + 1^I_{\mathfrak{Z}})}$. Set $N = \text{bgt}((^{@M})(^{@i}), (^{@M})(^{@m}), \mathfrak{A})$. Set $O = m :=_{\mathfrak{A}} (@i)$. Set $a = \text{the array sort of } \Sigma$. Set $P = \{s : s(a)(M) \neq \emptyset\}$. P is invariant w.r.t. J and f . P is invariant w.r.t. K and f . P is invariant w.r.t. W and f . P is invariant w.r.t. L and f . P is invariant w.r.t. N and f . P is invariant w.r.t. O and f . Set $ST = \mathfrak{C}$ -States (the generators of G). Set $TV = \text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). P is invariant w.r.t. if N then O and f . Now let Γ denotes the program

```
if N then
  O
fi;
L
```

For every s , $f(s, \text{if } N \text{ then } O)(I)(i) = s(I)(i)$ and $f(s, W)(I)(i) = s(I)(i)$ and $\text{length}_I^{@M} \text{value at } (\mathfrak{C}, f(s, W)) = \text{length}_I^{@M} \text{value at } (\mathfrak{C}, s)$ and $\text{length}_I^{@M} \text{value at } (\mathfrak{C}, f(s, \Gamma)) = \text{length}_I^{@M} \text{value at } (\mathfrak{C}, s)$. \square

3. SORTING BY EXCHANGING

In this paper i_1, i_2 denote pure elements of (the generators of G)(I).

Let us consider Σ, X, \mathfrak{F} , and G . We say that G is integer array if and only if

- (Def. 17) (i) $\{({}^{\textcircled{M}})(\tau)$ where τ is an element of \mathfrak{F} from I : not contradiction $\} \subseteq$
(the generators of G)(I), and
- (ii) for every M and for every element τ of \mathfrak{F} from I and for every element g of G from I such that $g = ({}^{\textcircled{M}})(\tau)$ there exists x such that $x \notin$
($\text{vf } \tau$)(I) and $\text{supp-var } g = x$ and ($\text{supp-term } g$)(the array sort of
 Σ)(M) = $({}^{\textcircled{M}})_{\tau \leftarrow \textcircled{x}}$ and for every sort symbol s of Σ and for every
 y such that $y \in (\text{vf } g)(s)$ and if $s =$ the array sort of Σ , then $y \neq M$
holds ($\text{supp-term } g$)(s)(y) = y .

Now we state the proposition:

- (96) If G is integer array, then for every element τ of \mathfrak{F} from I , $({}^{\textcircled{M}})(\tau) \in$
(the generators of G)(I).

The functor $\langle \mathbb{Z}, \leq \rangle$ yielding a strict real non empty poset is defined by the term

- (Def. 18) RealPoset \mathbb{Z} .

Let us consider Σ, X, \mathfrak{F} , and G . Let \mathfrak{A} be an elementary if-while algebra over the generators of G , a be a sort symbol of Σ , and τ_1, τ_2 be elements of \mathfrak{F} from a . Assume $\tau_1 \in$ (the generators of G)(a). The functor $\tau_1 :=_{\mathfrak{A}} \tau_2$ yielding an absolutely-terminating algorithm of \mathfrak{A} is defined by the term

- (Def. 19) (The assignments of \mathfrak{A})($\langle \tau_1, \tau_2 \rangle$).

Now we state the proposition:

- (97) Let us consider a countable non-empty many sorted set X indexed by the carrier of Σ , a vf-free including Σ -terms over X integer array non-empty free variable algebra \mathfrak{F} over Σ , a basic generator system G over Σ, X , and \mathfrak{F} , a pure element M of (the generators of G)(the array sort of Σ), and pure elements i, x of (the generators of G)(I). Then $({}^{\textcircled{M}})(\textcircled{i}) \neq x$. The theorem is a consequence of (73), (79), (61), and (74).

Let Σ be a non empty non void many sorted signature and \mathfrak{A} be a disjoint valued algebra over Σ . Note that the sorts of \mathfrak{A} is disjoint valued.

Let us consider Σ and X . Let \mathfrak{F} be an including Σ -terms over X algebra over Σ . We say that \mathfrak{F} is array degenerated if and only if

- (Def. 20) There exists I and there exists an element M of
(FreeGenerator(\mathfrak{F}))(the array sort of Σ) and there exists an element τ of \mathfrak{F}
from I such that $({}^{\textcircled{M}})(\tau) \neq \text{Sym}((\text{the connectives of } \Sigma)(11)(\in (\text{the carrier}$
of $\Sigma)), X)$ -tree($\langle M, \tau \rangle$).

Observe that $\mathfrak{F}_\Sigma(X)$ is non array degenerated.

Observe that there exists an including Σ -terms over X algebra over Σ which is non array degenerated.

Now we state the propositions:

- (98) Suppose \mathfrak{T} is non array degenerated. Then $\text{vf}((^{\textcircled{M}})^{(\textcircled{i})}) = I$ -singleton $i \cup$ (the array sort of Σ)-singleton M . The theorem is a consequence of (73).
 PROOF: Set $\tau = (^{\textcircled{M}})^{(\textcircled{i})}$. Reconsider $N = M$ as an element of $(\text{FreeGenerator}(\mathfrak{T}))$ (the array sort of Σ). Consider m being a set such that $m \in X$ (the array sort of Σ) and $M =$ the root tree of $\langle m, \text{the array sort of } \Sigma \rangle$. Consider j being a set such that $j \in X(I)$ and $i =$ the root tree of $\langle j, I \rangle$. $\{M\} = (\text{vf } \tau)$ (the array sort of Σ). $\{i\} = (\text{vf } \tau)(I)$. For every sort symbol s of Σ such that $s \neq$ the array sort of Σ and $s \neq I$ holds $\emptyset = (\text{vf } \tau)(s)$. \square
- (99) Let us consider an elementary if-while algebra \mathfrak{A} over the generators of G and an execution function f of \mathfrak{A} over \mathfrak{C} -States(the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Suppose
- (i) G is integer array and \mathfrak{C} -supported, and
 - (ii) $f \in \mathfrak{C}$ -Execution $_{b \neq \text{false}_{\mathfrak{C}}}$ (\mathfrak{A}), and
 - (iii) X is countable, and
 - (iv) \mathfrak{T} is non array degenerated.

Let us consider an element τ of \mathfrak{T} from I . Then $f(s, (^{\textcircled{M}})^{(\textcircled{i})} :=_{\mathfrak{A}} \tau) = f(s, M :=_{\mathfrak{A}} ((^{\textcircled{M}})^{(\textcircled{i})}_{\textcircled{i} \leftarrow \tau}))$. The theorem is a consequence of (96), (98), (97), (4), (3), (62), (73), (61), (84), (65), and (80). PROOF: Reconsider $H = \text{FreeGenerator}(\mathfrak{T})$ as a many sorted subset of the generators of G . Set $v = \tau$ value at (\mathfrak{C}, s) . Reconsider $p = (^{\textcircled{M}})^{(\textcircled{i})}$ as an element of G from I . Reconsider $g = s$ as a many sorted function from the generators of G into the sorts of \mathfrak{C} . Reconsider $g1 = f(s, (^{\textcircled{M}})^{(\textcircled{i})} :=_{\mathfrak{A}} \tau)$, $g2 = f(s, M :=_{\mathfrak{A}} ((^{\textcircled{M}})^{(\textcircled{i})}_{\textcircled{i} \leftarrow \tau}))$ as a many sorted function from the generators of G into the sorts of \mathfrak{C} . Reconsider $Mi = (^{\textcircled{M}})^{(\textcircled{i})}$ as an element of (the generators of G)(I). Reconsider $m = M$ as an element of G from the array sort of Σ . Consider x such that $x \notin (\text{vf } ^{\textcircled{i}})(I)$ and $\text{supp-var } p = x$ and $(\text{supp-term } p)$ (the array sort of Σ)(M) = $(^{\textcircled{M}})^{(\textcircled{i})}_{\textcircled{i} \leftarrow \textcircled{x}}$ and for every sort symbol s of Σ and for every y such that $y \in (\text{vf } p)(s)$ and if $s =$ the array sort of Σ , then $y \neq M$ holds $(\text{supp-term } p)(s)(y) = y$. $g1 = g2$. \square

Let us consider $\Sigma, X, \mathfrak{T}, G, \mathfrak{C}, s,$ and b . Let us observe that $s((\text{the boolean sort of } \Sigma))(b)$ is boolean.

Now we state the proposition:

- (100) Now let Γ denotes the program

```

while  $J$  do
   $y := \mathfrak{A}(@M)(@i_1)$ ;
   $(@M)(@i_1) := \mathfrak{A}(@M)(@i_2)$ ;
   $(@M)(@i_2) := \mathfrak{A}^@y$ 
done

```

Let us consider an elementary if-while algebra \mathfrak{A} over the generators of G and an execution function f of \mathfrak{A} over \mathfrak{C} -States (the generators of G) and $\text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Suppose

- (i) G is integer array and \mathfrak{C} -supported, and
- (ii) $f \in \mathfrak{C}\text{-Execution}_{b \neq \text{false}_{\mathfrak{C}}}(\mathfrak{A})$, and
- (iii) \mathfrak{T} is non array degenerated, and
- (iv) X is countable.

Let us consider an algorithm J of \mathfrak{A} . Suppose

- (v) $f(s, J)$ (the array sort of Σ)(M) = s (the array sort of Σ)(M), and
- (vi) for every array D of $\langle \mathbb{Z}, \leq \rangle$ such that $D = s$ (the array sort of Σ)(M) holds if $D \neq \emptyset$, then $f(s, J)(I)(i_1)$, $f(s, J)(I)(i_2) \in \text{dom } D$ and if inversions $D \neq \emptyset$, then $\{f(s, J)(I)(i_1), f(s, J)(I)(i_2)\} \in \text{inversions } D$ and $f(s, J)$ ((the boolean sort of Σ))(b) = *true* iff inversions $D \neq \emptyset$.

Let us consider a 0-based finite array D of $\langle \mathbb{Z}, \leq \rangle$. Suppose

- (vii) $D = s$ (the array sort of Σ)(M), and
- (viii) $y \neq i_1$, and
- (ix) $y \neq i_2$.

Then

- (x) $f(s, \Gamma)$ (the array sort of Σ)(M) is an ascending permutation of D , and
- (xi) if J is absolutely-terminating, then Γ is terminating w.r.t. f and $\{s_1 : s_1 \text{ (the array sort of } \Sigma \text{)(} M \text{)} \neq \emptyset\}$.

The theorem is a consequence of (73), (10), (61), (65), (99), (80), (74), and (79). PROOF: Define \mathcal{F} (natural number, element of \mathfrak{C} -States (the generators of G)) = $f(\$_2, ((J; y := \mathfrak{A}(@M)(@i_1)); (@M)(@i_1) := \mathfrak{A}(@M)(@i_2)); (@M)(@i_2) := \mathfrak{A}^@y))$. Set $ST = \mathfrak{C}\text{-States}$ (the generators of G). Consider g being a function from \mathbb{N} into ST such that $g(0) = s$ and for every natural number i , $g(i+1) = \mathcal{F}(i, (g(i) \text{ qua element of } ST))$. Define \mathcal{G} (element) = $g(\$_1 \in \mathbb{N})$ (the array sort of Σ)(M). Consider h being a function from \mathbb{N} into \mathbb{Z}^ω such that for every element i such that $i \in \mathbb{N}$ holds $h(i) = \mathcal{G}(i)$. For every ordinal number a such that $a \in \text{dom } g$ holds $h(a)$ is an array of $\langle \mathbb{Z}, \leq \rangle$. Set $TV = \text{States}_{b \neq \text{false}_{\mathfrak{C}}}$ (the generators of G). Consider s_1 such that $s = s_1$ and s_1 (the array sort of Σ)(M) $\neq \emptyset$. Reconsider

$D = s(\text{the array sort of } \Sigma)(M)$ as a 0-based finite non empty array of $\langle \mathbb{Z}, \leq \rangle$. Consider g being a function from \mathbb{N} into ST such that $g(0) = s$ and for every natural number i , $g(i + 1) = \mathcal{F}(i, (g(i) \text{ qua element of } ST))$. Define $\mathcal{G}(\text{element}) = g(\$_1(\in \mathbb{N}))(\text{the array sort of } \Sigma)(M)$. Consider h being a function from \mathbb{N} into \mathbb{Z}^ω such that for every element i such that $i \in \mathbb{N}$ holds $h(i) = \mathcal{G}(i)$. For every ordinal number a such that $a \in \text{dom } g$ holds $h(a)$ is an array of $\langle \mathbb{Z}, \leq \rangle$. Define $\mathfrak{T}[\text{natural number}] \equiv h(\$_1) \neq \emptyset$. For every natural number i such that $\mathfrak{T}[i]$ holds $\mathfrak{T}[i + 1]$. For every natural number a and for every array R of $\langle \mathbb{Z}, \leq \rangle$ such that $R = h(a)$ for every s such that $g(a) = s$ there exist sets x, y such that $x = f(s, J)(I)(i_1)$ and $y = f(s, J)(I)(i_2)$ and $x, y \in \text{dom } R$ and $h(a + 1) = \text{Swap}(R, x, y)$. Define $\mathcal{Q}[\text{natural number}] \equiv h(\$_1)$ is a permutation of D . Define $\mathcal{P}[\text{natural number}] \equiv g(\$_1)(\text{the array sort of } \Sigma)(M)$ is an ascending permutation of D . There exists a natural number i such that $\mathcal{P}[i]$. Consider \mathfrak{B} being a natural number such that $\mathcal{P}[\mathfrak{B}]$ and for every natural number i such that $\mathcal{P}[i]$ holds $\mathfrak{B} \leq i$. Reconsider $c = h \upharpoonright \text{succ } \mathfrak{B}$ as an array of \mathbb{Z}^ω . Set $TV = \text{States}_{b \neq \text{false}_c}$ (the generators of G). Define $\mathcal{H}(\text{natural number}) = f(g(\$_1 - 1), J)$. Consider r being a finite sequence such that $\text{len } r = \mathfrak{B} + 1$ and for every natural number i such that $i \in \text{dom } r$ holds $r(i) = \mathcal{H}(i)$. $\text{rng } r \subseteq ST$. Reconsider $R = g(\mathfrak{B})(\text{the array sort of } \Sigma)(M)$ as an ascending permutation of D . Now let Γ denotes the program

$$\begin{aligned}
&y := \mathfrak{a}^{(@M)}(@i_1); \\
&(@M)(@i_1) := \mathfrak{a}^{(@M)}(@i_2); \\
&(@M)(@i_2) := \mathfrak{a}^y; \\
&J
\end{aligned}$$

For every natural number i such that $1 \leq i < \text{len } r$ holds $r(i) \in TV$ and $r(i + 1) = f(r(i), \Gamma)$. \square

REFERENCES

- [1] Grzegorz Bancerek. Mizar analysis of algorithms: Preliminaries. *Formalized Mathematics*, 15(3):87–110, 2007. doi:10.2478/v10037-007-0011-x.
- [2] Grzegorz Bancerek. Program algebra over an algebra. *Formalized Mathematics*, 20(4):309–341, 2012. doi:10.2478/v10037-012-0037-6.
- [3] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(2):377–382, 1990.
- [4] Grzegorz Bancerek. Sorting by exchanging. *Formalized Mathematics*, 19(2):93–102, 2011. doi:10.2478/v10037-011-0015-4.
- [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. Complete lattices. *Formalized Mathematics*, 2(5):719–725, 1991.
- [7] Grzegorz Bancerek. Free term algebras. *Formalized Mathematics*, 20(3):239–256, 2012. doi:10.2478/v10037-012-0029-6.
- [8] Grzegorz Bancerek. Terms over many sorted universal algebra. *Formalized Mathematics*, 5(2):191–198, 1996.
- [9] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(1):91–96, 1990.
- [10] Grzegorz Bancerek. König’s lemma. *Formalized Mathematics*, 2(3):397–402, 1991.

- [11] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(1):77–82, 1993.
- [12] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(1):93–107, 1997.
- [13] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(1):107–114, 1990.
- [14] Grzegorz Bancerek and Artur Korniłowicz. Yet another construction of free algebra. *Formalized Mathematics*, 9(4):779–785, 2001.
- [15] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(4):485–492, 1996.
- [16] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(1):47–54, 1996.
- [17] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(1):175–180, 1990.
- [18] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(3):529–536, 1990.
- [19] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(1):55–65, 1990.
- [20] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(1):153–164, 1990.
- [21] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(2):357–367, 1990.
- [22] Czesław Byliński. Galois connections. *Formalized Mathematics*, 6(1):131–143, 1997.
- [23] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(1):165–167, 1990.
- [24] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(1):61–65, 1996.
- [25] Jarosław Kotowicz, Beata Madras, and Małgorzata Korolkiewicz. Basic notation of universal algebra. *Formalized Mathematics*, 3(2):251–253, 1992.
- [26] Rafał Kwiatek. Factorial and Newton coefficients. *Formalized Mathematics*, 1(5):887–890, 1990.
- [27] Takashi Mitsuishi and Grzegorz Bancerek. Lattice of fuzzy sets. *Formalized Mathematics*, 11(4):393–398, 2003.
- [28] Beata Perkowska. Free many sorted universal algebra. *Formalized Mathematics*, 5(1):67–74, 1996.
- [29] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(2):329–334, 1990.
- [30] Andrzej Trybulec. A scheme for extensions of homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(2):205–209, 1996.
- [31] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(1):37–42, 1996.
- [32] Andrzej Trybulec. Many sorted sets. *Formalized Mathematics*, 4(1):15–22, 1993.
- [33] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(3):501–505, 1990.
- [34] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(3):575–579, 1990.
- [35] Wojciech A. Trybulec and Grzegorz Bancerek. Kuratowski – Zorn lemma. *Formalized Mathematics*, 1(2):387–393, 1990.
- [36] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(1):67–71, 1990.
- [37] Tetsuya Tsunetou, Grzegorz Bancerek, and Yatsuka Nakamura. Zero-based finite sequences. *Formalized Mathematics*, 9(4):825–829, 2001.
- [38] Edmund Woronowicz. Many argument relations. *Formalized Mathematics*, 1(4):733–737, 1990.
- [39] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(1):73–83, 1990.

Received November 9, 2012