

Axioms for multilevel objects

J. A. BERGSTRA, H. J. M. GOEMAN, A. OLLONGREN, G. A. TERPSTRA,
Th. P. VAN DER WEIDE

Department of Mathematics, University of Leiden, The Netherlands

Received February 22, 1978

AMS Categories: 68A05

Abstract. A set of axioms for structured objects of data is presented. In the structured objects components and levels are distinguished. Change of level is the result of a special application operator, components are accessible by successive selections. The set of access paths is also axiomatized. The set of axioms is uniform in the sense that features of various known classes of datastructures are combined.

Key words: multi-level datastructures, programming.

1. Motivation

In a number of papers H. D. Ehrich (1976) has outlined an algebraic theory for datastructures and their semantics. This work can be seen as an extension of earlier work on the so-called Viennese datastructures by P. Lucas et al. (1970), P. Wegner (1972), and one of the present authors (A. Ollongren (1974)).

The datastructures introduced are generalisations over the Viennese objects in two respects:

- (a) the structures are infinite and
- (b) each structure can be represented by a rooted, directed, labelled tree with elementary information associated with each node. A Viennese datastructure can be represented by a finite tree of the same type, but with only non-zero information associated with the leaves.

In the present paper we describe a more general kind of data objects which have components accessible via access paths. The underlying collection of access paths will be a monoid satisfying a scratching property. This condition does not exclude circularities.

On the other hand we intend to describe objects with multilevel structure. This level structure should not show any circularities. The motivation for this regularity is that data items in a data object may themselves be structured data objects and that all objects are built up via successive assignments (ν) from smaller objects starting from the empty object and

elementary objects. We need a mechanism to read data items from a data object A . This is done by the application (a) which reads from A the item that has been placed at the root of the object. To read other items a selection must be performed first.

We will restrict our considerations to an axiomatisation using a first order language. Finiteness of the data objects is expressed in terms of induction axioms. Especially we consider a scheme of domain induction, expressing that only finitely many components of a data object can be nontrivial, and a scheme of depth induction which concerns the multilevel structure. Of course our theories have nonstandard models but we assume that the axioms capture important facts which may play a role in correctness proofs of algorithms using the data objects.

In section 7 we introduce new axioms concerning the monoid of selectors which essentially turn it into a tree. It is then possible to encode applications within the selectors.

2. The class of objects

Consider the structures $\mathcal{O} = (O, (CS, *, \varepsilon), a, \circ, \nu, \mathcal{E}, \Omega)$ in which:

Ω the *empty object*, is a constant in O ,

\mathcal{E} is a predicate over O for *elementary objects*,

a , the *application operation*, is of type $O \rightarrow O$,

\circ , the *selection operation*, is of type $O \times CS \rightarrow O$,

ν , the *assignment operator*, is of type $O \times CS \times O \rightarrow O$,

$(CS, *, \varepsilon)$ are structures in which

ε , the *zero element*, is a constant in CS ,

$*$ is a binary operation of type $CS \times CS \rightarrow CS$,

and for which we require the following axioms

$$A0 \quad (a * \beta) * \gamma = a * (\beta * \gamma)$$

$$a \neq \beta \Rightarrow \gamma * a \neq \gamma * \beta \quad (\text{scratching property})$$

$$\varepsilon * a = a * \varepsilon = a.$$

These structures are described in a language \mathcal{L}_0 and we require \mathcal{O} to satisfy the following axioms

$$A1 \text{ (selection)} \quad A \circ (a * \beta) = (A \circ a) \circ \beta,$$

$$A \circ \varepsilon = A,$$

$$A2 \text{ (elementary objects)} \quad \mathcal{E}(e) \Leftrightarrow [\forall a (a \neq \varepsilon \Rightarrow \varepsilon \circ a = \Omega) \wedge ea = e \wedge e \neq \Omega],$$

$$A3 \text{ (empty object)} \quad \Omega \circ a = \Omega \wedge \Omega a = \Omega,$$

$$A4 \text{ (extensionality)} \quad ((\forall a)[(A \circ a)a = (B \circ a)a]) \Rightarrow A = B,$$

$$A5 \text{ (assignment)} \quad a \neq \beta \Rightarrow [(\nu(A, a, B) \circ \beta)a = (A \circ \beta)a] \wedge (\nu(A, a, B) \circ a)$$

$$a = B.$$

The set of axioms in \mathcal{L}_0 is called the theory \mathcal{T}_0 . Using the theory \mathcal{T}_0 we can formulate a few lemmas and theorems. We give them in the form of a few remarks and we omit the proofs. First we need a definition.

DEFINITION 2.1.

$$\begin{aligned} \alpha \leq \beta &\equiv (\exists \gamma)[\alpha * \gamma = \beta], \\ \alpha < \beta &\equiv (\exists \gamma)[\gamma \neq \varepsilon \wedge \alpha * \gamma = \beta]. \end{aligned}$$

Remarks:

$$\begin{aligned} \alpha < \beta &\Rightarrow (\exists ! \gamma)[\gamma \neq \varepsilon \wedge \alpha * \gamma = \beta], \\ \nu(A, \alpha * \beta, B) \circ \alpha &= \nu(A \circ \alpha, \beta, B), \\ (\sqcap \alpha \leq \beta) &\Rightarrow \nu(A, \beta, B) \circ \alpha = A \circ \alpha, \\ A\alpha = B\alpha \wedge ((\forall a \neq \varepsilon)[A \circ a = B \circ a]) &\Rightarrow A = B, \\ \mathcal{E}(e) &\Rightarrow \nu(\Omega, \varepsilon, e) = e. \end{aligned}$$

For sake of simplicity we write in the sequel $A(\alpha)$ instead of $(A \circ \alpha)\alpha$.
If $\mathcal{E}(e)$ then $e(\varepsilon) = e$.

3. Some models for O

Let E be a set of constants used for elementary objects and let CS be a set of constants used for access paths. We can build a term model (in the logical sense) \mathcal{T}_{CS}^E for the axioms A0–A5 as follows: build terms from the constants, $*$ (applied on terms of access path type), \circ , α , ν and Ω . Divide at the equivalence relation on the terms which is generated by the axioms A0–A5.

Next we shall show consistency of the axioms by considering a model as follows. Let E be a non-empty set of names for elementary objects, and $\Omega \notin E$. $\{\tau_i\}_{i=0}^\infty$ is some enumeration of Σ^* for some countable Σ with $\tau_0 = \varepsilon$. We define the set of expressions \mathcal{A} by

DEFINITION 3.1.

$$\Omega \in \mathcal{A};$$

$$\text{if } e \in E, \text{ then } e \in \mathcal{A};$$

if $A_0, A_1, \dots, A_k \in \mathcal{A}$, then $(A_0)(A_1) \dots (A_k) \in \mathcal{A}$, expressions are only to be formed with these rules.

We introduce the equivalence relation \equiv on \mathcal{A} as the reflexive, symmetric and transitive closure of the relation \sim on \mathcal{A} defined by

DEFINITION 3.2.

$$(\Omega) \sim \Omega,$$

$$(e) \sim e,$$

$$(A_0)(A_1) \dots (A_k)(\Omega) \sim (A_0)(A_1) \dots (A_k) \quad k \geq 0.$$

Finally the operator ν is defined by

DEFINITION 3.3.

$$\nu(\Omega, \tau_i, B) = (A_0)(A_1) \dots (A_{i-1})(B)$$

with $A_0 = A_1 = \dots = A_{i-1} = \Omega$;

$$\nu(e, \tau_i, B) = (A_0)(A_1) \dots (A_{i-1})(B)$$

with $A_0 = e$ and $A_1 = \dots = A_{i-1} = \Omega$;

$$\begin{aligned} \nu((A_0) \dots (A_k), \tau_i, B) = \\ \text{for } i \leq k : (A_0) \dots (A_{i-1})(B)(A_{i+1}) \dots (A_k), \\ \text{for } i > k : (A_0) \dots (A_k)(A_{k+1}) \dots (A_{i-1})(B), \end{aligned}$$

with $A_{k+1} = \dots = A_{i-1} = \Omega$;

$$\begin{aligned} \nu(A_0) \dots (A_k) a = A_0; \\ \nu(A_0) \dots (A_k) \circ \sigma = (B_0) \dots (B_1), \end{aligned}$$

where l is the maximum of the codes j such that for some $t \leq k$ $\sigma * \tau_j = \tau_t$, and for all $j \leq l$ $B_j = A_m$ for m such that $\tau_m = \sigma * \tau_j$.

It is easy to see that the operations and the predicate respect the equivalence relation \equiv . Thus we can consider them in a natural way as operations in c.q. a predicate on A/\equiv . Now A/\equiv forms a model for \mathcal{O} and as a result we have shown consistency.

In sections 4 and 5 we formulate axioms A6 and A7 which are trivially satisfied in our models. Both axioms are induction schemes expressing the finiteness of the data objects in some sense. Of course non-standard models will not be excluded this way.

In terms of programming languages an example of a multilevel structured object can be given as follows. Consider the case expression CST given by

case σ **of** $\sigma_0 : A_0$

—
—
—
—

$\sigma_n : A_n$

esac

where the A 's are expressions and the σ 's are not necessarily the ones of the enumeration mentioned before. In a representation of this statement as a structured object one codes the σ 's by binary numbers using two selectors s and t in Σ . Given some $\sigma_i \in \Sigma^*$ one selects the appropriate expression by CST $(\sigma_i) = A_i$. At this level A_i can be again a case expression.

4. Domain induction

We will give an axiom scheme which expresses the fact that all objects have a finite domain. We need a definition first.

DEFINITION 4.1. $A \leq B \equiv \forall a [A(a) \neq \Omega \Rightarrow A(a) = B(a)],$

$$A < B \equiv A \leq B \wedge A \neq B.$$

\leq is a partial ordering Ω is the unique minimal element. Using this partial order we are able to state the axiom scheme of domain induction (DOI) as follows

$$A6 \quad \forall A [(\forall B < A)[\Phi(B, -)] \Rightarrow \Phi(A, -)] \Rightarrow \forall A [\Phi(A, -)]$$

where $-$ stands for a list of parameters not containing the first argument of Φ .

5. Mutation

We introduce a new symbol, μ , the μ -operator of type $O \times CS \times O \rightarrow O$. For μ we require the following mutation axiom

$$A\mu \quad \varphi(A, a, B, \mu(A, a, B)) \text{ where} \\ \varphi(A, a, B, C) \equiv \forall \beta [\text{if } a \leq \beta \text{ then } \exists \gamma [a * \gamma = \beta \wedge C(\beta) = B(\gamma)] \\ \text{else } C(\beta) = A(\beta) \text{ fi}]$$

$A\mu$ guarantees that in the intended interpretation μ will be the well-known μ -operator. It turns out that $A\mu$ is conservative over $A0$ – $A6$, i.e. μ can be defined in $A0$ – $A6$. This is a consequence of the next theorem:

THEOREM 5.1. $A0$ – $A6 \vdash (\forall A, a, B)(\exists! C)[\varphi(A, a, B, C)]$.

Proof: By extensionality we have $\exists C[\varphi(A, a, B, C)] \Rightarrow (\exists! C)[\varphi(A, a, B, C)]$. Next we define the relation P by $P(A, B) \equiv \forall a \exists C[\varphi(A, a, B, C)]$ and we will first prove $\forall A [P(A, \Omega)]$, using domain induction DOI. The basis $P(\Omega, \Omega)$ is trivial. Suppose $\forall A' < AP[A', \Omega]$. We have

if $A \circ a = \Omega$ then $\varphi(A, a, \Omega, A)$,
if $A \circ a \neq \Omega$ then $\exists \beta [A(a * \beta) \neq \Omega]$.

Let $A' = v(A, a * \beta, \Omega)$, then $A' < A$ and so $P(A', \Omega)$. Let C be such that $\varphi(A', a, \Omega, C)$, then it can be seen easily that $\varphi(A, a, \Omega, C)$. The conclusion is $P(A, \Omega)$ and application of $A6$ yields $\forall A [P(A, \Omega)]$.

Next we use this result as the basis of a new induction in order to prove $\forall A, B [P(A, B)]$. Suppose $(\forall B' < B) \forall A [P(A, B')]$.

If $B = \Omega$ then $\forall A [P(A, \Omega)]$ because of the previous result.

if $B \neq \Omega$ then $\exists \beta [B(\beta) \neq \Omega]$.

Let $B' = v(B, \beta, \Omega)$; then $B' < B$ so $P(A, B')$. Let C' be such that $\varphi(A, a, B', C')$ and let $C = v(C', a * \beta, B \circ \beta)$; then it is easily seen that $\varphi(A, a, B, C)$. The conclusion is $P(A, B)$ and application of $A6$ yields $\forall A, B [P(A, B)]$. ■

The moral is that we can already define the μ -operator in $A0$ – $A6$ and that simple selectors are of no importance for the definition of μ (we will introduce simple selectors in section 7).

6. Depth induction

We state the axiom scheme of depth induction (DEI) at once:

$$A7 \quad \left. \begin{array}{l} \Phi(A, -) \text{ for } A = \Omega \text{ or } \mathcal{E}(A) \\ \forall B [\forall a [\Phi(B(a), -)] \Rightarrow \Phi(B, -)] \end{array} \right\} \Rightarrow \forall B [\Phi(B, -)]$$

This axiom scheme expresses the fact that in the models all objects have finite depth. Using DOI and DEI we can prove another useful scheme of induction. This is expressed by the next theorem.

THEOREM 6.1 (structural induction).

$$\left. \begin{array}{l} P(A, -) \text{ for } A = \Omega \text{ or } \mathcal{E}(A) \\ \forall A, B, a [P(A, -) \wedge P(B, -) \Rightarrow P(\nu(A, a, B), -)] \end{array} \right\} \Rightarrow \forall A [P(A, -)]$$

Proof: we use depth induction on A . The basis is obvious and so suppose $\forall a(PA(a), -)$. We must prove then $P(A, -)$, but we will prove the stronger statement $(\forall A' \leq A) [P(A', -)]$. In order to do this we use the notation $P_A(A', -)$ for $[A' \leq A \Rightarrow P(A', -)]$ and we show $\forall A' [P_A(A', -)]$ using domain induction as follows:

Suppose $(\forall X < Y) [P_A(X, -)]$ so that we must prove $P_A(Y, -)$. Since $P(\Omega, -)$ holds and hence $P_A(\Omega, -)$, we may assume $Y \neq \Omega$. Take a such that $Y(a) \neq \Omega$ and consider $Y' = \nu(Y, a, \Omega)$.

As $Y' < Y$ we know $P_A(Y', -)$ holds. Assume $Y \leq A$ so that $Y' \leq A$ and hence $P(Y', -)$ holds. We know by the hypothesis of the depth induction that $P(A(a), -)$ holds. And therefore, as $Y \leq A$, we have $P(Y(a), -)$ as well. By the assumption of the theorem we know $P(\nu(Y', a, Y(a)), -)$. Extensionality proves $\nu(Y', a, Y(a)) = Y$, so that $P(Y, -)$ holds. ■

7. The need for extra axioms

In this section we intend to look at structures in which the set of access paths can be seen as Σ^* for some alphabet Σ . So, given Σ and E , we generate the model $\mathcal{T}_{\Sigma^*}^E$ for \mathcal{O} by taking for CS the set Σ^* , for $*$ the concatenation operator for strings, and for ε the empty string.

Now consider the following sentence

$$\psi \equiv \forall a \exists A \forall \beta [A(\beta) \neq \Omega \Leftrightarrow \beta \leq a]$$

which is true in $\mathcal{T}_{\Sigma^*}^E$ as every $a \in \Sigma^*$ has only finitely many predecessors. However,

LEMMA 7.1.

$$A0, A1, \dots, A7 \vdash \text{non}\psi.$$

Proof: Consider the interpretation $\mathcal{T}_{\text{CS}}^E$ for the structure \mathcal{O} , where CS is the set of integers, $+$ is the addition operator, $\varepsilon = 0$ and $E \neq \emptyset$. Because of the following properties

$$\forall A \exists \beta [A(\beta) = \Omega],$$

$$\forall a \forall \beta [\beta \leq a],$$

$$\psi \Rightarrow \exists A \forall a [A(a) \neq \Omega],$$

we may conclude $\mathcal{T}_{\text{CS}}^E \models \neg \psi$. ■

We shall prove now that we need extra axioms in order to be able to derive ψ .

DEFINITION 7.1. A sentence φ is *admissible* if it is the universal closure of a formula not containing quantifiers over objects,

the set of admissible sentences is denoted by Ad ,

the set of all sentences which are valid in \mathcal{O}_{Σ}^E is denoted by $\text{Th}(\mathcal{O}_{\Sigma}^E)$ and $\text{Th}(\mathcal{O}_{\Sigma}^E) \cap \text{Ad}$ is denoted by T' .

Note that ψ is not admissible and that T' does not contain statements expressing the existence of objects.

THEOREM 7.1. *Let Σ and E be non-empty sets. Then*

$$T', A0, A1, \dots, A7 \vdash \text{non}\psi.$$

Proof: Let p and q be new constants for access paths and let T^* be the theory resulting from the axioms $T', A0, A1, \dots, A7$, augmented with all axioms

$$p^k < q \quad \text{for } k \in N$$

where

$$p^0 = \varepsilon \quad \text{and} \quad p^{k+1} = p^k * p \quad \text{for } k \in N.$$

Using the well-known compactness theorem it can be shown that T^* is a consistent set of axioms, so that there exists a model \mathcal{A} for T^* . We can define a substructure \mathcal{A}' of \mathcal{A} by restricting \mathcal{A} to objects which can be built by finitely many applications of the ν operator, starting from $E \cup \{\Omega\}$ and all access paths of \mathcal{A} . Now we have

$$q^1 : \mathcal{A}' \text{ is a model for } T \quad q^2 : \mathcal{A}' \text{non} \models \psi.$$

Concerning q^1 : let φ be an admissible sentence in T ; it has the form

$$\forall A_1, \dots, A_n [\varphi'(A_1, \dots, A_n)]$$

where φ' contains no quantifiers over objects and the A 's range over A' . As $\mathcal{A} \models \varphi$ we have $\mathcal{A}' \models \varphi$ as well.

Concerning q^2 : the constant q has infinitely many predecessors in \mathcal{A} and hence in \mathcal{A}' . But all objects in \mathcal{A}' have a finite domain. Therefore no object A exists in \mathcal{A}' with the property $\forall \beta < q [A(\beta) \neq \Omega]$. ■

In order to be able to derive ψ we evidently need extra axioms (cf. lemma 7.1).

Theorem 7.1 says that if we add admissible extra axioms (i.e. axioms which do not say anything about the existence of objects, such as all facts which have to do with sequences), then ψ still cannot be derived. We could use ψ itself as an extra axiom because it is concerned with the existence of objects, but this is an unsatisfactory solution. Instead we use an axiom scheme for sequence induction together with axioms about atomic selectors.

DEFINITION 7.2 (atomic selector a).

$$\text{At}(a) \equiv [a \neq \varepsilon \wedge \forall \beta \forall \gamma [\beta * \gamma = a \Rightarrow \beta = a \vee \gamma = a]].$$

$$\text{A8} \quad (1) \text{ At}(a) \Rightarrow a * \beta \neq \varepsilon \wedge \beta * a \neq \varepsilon,$$

$$(2) \forall a \neq \varepsilon \exists \beta \exists \gamma [\text{At}(\gamma) \wedge a = \beta * \gamma],$$

$$(3) \text{ At}(s) \wedge \text{At}(t) \Rightarrow \forall a \forall \beta [a * s = \beta * t \Rightarrow a = \beta].$$

(2) together with (3) enable us to choose for every non-trivial a a unique predecessor of it by deleting the last atomic selector.

A9 (sequence induction, SI)

$$\forall a [(\forall \beta < a P(\beta, -)) \Rightarrow P(a, -)] \Rightarrow \forall a P(a, -).$$

THEOREM 7.2.

$$\text{A0}, \text{A1}, \dots, \text{A7}, \text{A8}(1), \text{A9} \vdash \psi.$$

Proof: via a straight forward application of SI and A8(1). ■

8. Encoding application in selection

Consider \mathcal{O}_{Σ}^E for some non-empty sets E and Σ . In this case all objects can be considered as trees and the application operation may be regarded as just another selector. One possibility to express this is to extend the space of composite selectors to $(\Sigma \cup \{a\})^*$. Another possibility will be given in detail here. The idea is to encode $(\Sigma \cup \{a\})^*$ in Σ^* itself. Therefore we introduce a new selection operator $\theta : \mathcal{O} \times \text{CS} \rightarrow \mathcal{O}$ which regards its second element as a code of a string in $(\Sigma \cup \{a\})^*$ and corresponds to a succession of applications and usual selections. We choose a fixed selector s_0 and use it as an indicator whether to apply or select. Besides s_0 we assume that there is at least one other simple selector:

$$\text{A10} \quad \text{At}(s_0) \wedge \text{At}(s_1) \wedge s_0 \neq s_1.$$

Recursion equations for θ are as follows

$$\text{A11} \quad A\theta\varepsilon = A$$

$$\text{At}(a) \Rightarrow A\theta a = \Omega$$

$$\text{At}(s) \wedge \text{At}(t) \Rightarrow A\theta(a * s * t) =$$

$$\text{if } s = s_0 \text{ then } (A\theta a)(\varepsilon) \text{ else } (A\theta a) \circ t \text{ fi}$$

Now it is easy to prove that the relation $A\theta a = B$ is *not* definable in the language used for \mathcal{O} , using induction on the number of application symbols a occurring in a formula. Therefore we add θ to our language, as well as the constant symbols s_0 and s_1 , and A11 is added as an axiom as well as A10.

For θ we have the property

LEMMA 8.1. *Let $\varphi(a) \equiv \forall A, s, t [At(s) \wedge At(t) \Rightarrow A\theta(s*t*a) = (A\theta(s*t)) \times \theta a]$, then $\forall a[\varphi(a)]$.*

Proof: straightforward using sequence induction over a and A9. ■

DEFINITION 8.1.

$$\Delta(\beta) \equiv \forall A, \gamma [(A\theta\beta)\theta\gamma = A\theta(\beta*\gamma)].$$

THEOREM 8.1.

$$\forall A \forall B [\forall \tau, e [A\theta\tau = e \Leftrightarrow B\theta\tau = e] \Rightarrow A = B].$$

Proof: By using depth induction and the following lemmas:

$$\forall \beta \exists \gamma [\Delta(\gamma) \wedge \forall A [A \circ \beta = A\theta\gamma]]$$

(proved by using sequence induction over β and lemma 8.1);

$$\forall A [A \neq \Omega \Rightarrow \exists \tau, e [A\theta\tau = e]]$$

(proved by using depth induction over A , axiom A9 and lemma 8.1);

$$\forall B [\forall \tau, e [B\theta\tau \neq e] \Rightarrow B = \Omega]$$

(proved by the previous lemma). ■

9. Viennese objects and multilevel structures

The objects used in the Vienna method for the definition of the semantics of programming languages, and axiomatized for the first time in A. Ollongren (1974), can be interpreted in our theory as a substructure of \mathcal{O} . The following formula defines the Viennese objects

$$\forall a [A(a) \neq \Omega \Rightarrow \mathcal{E}(A \circ a)].$$

If A is a Viennese object one can prove

$$\mathcal{E}(A \circ e) \wedge \beta < a \Rightarrow A(\beta) = \Omega.$$

The objects proposed by H. D. Ehrich (1976) are generalizations of the Viennese objects, and the subclass of finite objects can be interpreted in our theory as a substructure of \mathcal{O} (but then an addition operation on elementary objects must be introduced). The following formula defines them

$$\forall a [A(a) \neq \Omega \Rightarrow \mathcal{E}(A(a))].$$

Concerning multilevel structures we note the following. Let \mathcal{P} be a set of sets. Then — according to A. L. Rosenberg and J. W. Thatcher (1975) — the set $\mathcal{F}_{\mathcal{P}}^E$ of multilevel arrays over \mathcal{P} , E is defined as the minimal solution of

$$F = E \cup \bigcup_{p \in \mathcal{P}} [P \rightarrow F].$$

If the $P \in \mathcal{P}$ all have a suitable similar selector space structure, it is possible to embed $\mathcal{F}_{\mathcal{P}}^E$ in \mathcal{O}_{CS}^E for some CS in a natural way. If, moreover, the sets $P \in \mathcal{P}$ are definable one can define $\mathcal{M}_{\mathcal{P}}^E$ in \mathcal{O}_{CS}^E . An advantage of this interpretation is, that we may single out \perp in E and identify all objects in which only \perp occurs in elementary information with Ω .

References

- [1] Ehrich, H. D., *Outline of an algebraic theory of structured objects*, Automata, Languages and Programming, Third International Colloquium, Edinburgh University Press, 1976, pp. 508-530.
 - [2] Lucas, P., Lauer, P., Stigleitner, H., *Method and notation for the formal definition of programming languages*, TR 25.087, IBM Laboratory Vienna 1970.
 - [3] Ollongren, A., *Definition of programming languages by interpreting automata*, Academic Press, APIC Series No 11, 1974.
 - [4] Rosenberg, A. L., Thatcher, J. W., *What is a multilevel array?*, IBM Journal of Research and Development 19 (1975), 163-169.
 - [5] Wegner, P., *The Vienna definition language*, ACM Computing Surveys 4 (1972), 5-63.
-