FUNCTIONAL COMPLETENESS IN ITERATIVE META-ALGEBRAS

G. E. Tseitlin

We consider the problem of functional completeness in iterative meta-algebras with generator systems containing various loop constructs. Meta-algebras are intended for the design of whole classes of algebras of algorithms associated with structured programming, goto programming, and the combination of these programming methods. Families of maximal subalgebras of the Dijkstra meta-algebra and its generalizations are constructed. Functional completeness theorems for the proposed meta-algebras are stated and proved in terms of these families.†

1. STATEMENT OF THE PROBLEM

Functional completeness is one of the most topical and important problems in the algebra of logic. It suffices to mention the seminal research of Post [1, 2] for two-valued algebra of logic, the work of the Moscow mathematics school headed by S. V. Yablonskii for k-valued logics (k ≥ 3) [3, 4], the studies of Rosenberg [5], and the research of Kaluzhnin and his students [6, 7] on lattices of subalgebras of n-relations and algebra of logic, etc. The signature of the algebra of logic consists of superposition, which is interpreted following Mal’tsev [8] by binary substitution and the unary operations of renaming and identification of renamed functions.

Promising directions of research by the Kiev algebraic-cybernetic school include basic research into the algebraic foundations of algorithmics — the applied theory of algorithms that has important applications for the solution of various problems, including, in particular, problems in the theory and practice of modern programming. The problem of constructing an algebra of algorithms was posed by Kaluzhnin [9]. An algebra of algorithms that anticipated the structured programming paradigm was first constructed by Glushkov in his seminal work on the theory of systems of algorithmic algebras (SAA) [10, 11]. Functional completeness was investigated in [12] for algebras of periodically defined transformations in the context of further development of the theory of SAA and their modifications intended for formalization of nondeterministic and parallel computations. The completeness problem for SAA, program algebras, and systems is discussed in [13-15].

In this article we continue the series of our studies in the algebra of algorithmics [16-19] that involve the construction of meta-algebras associated with various methods of algorithm and program design, as well as methods for the construction of classes of algorithms and programs. A specific feature of meta-algebras is that their signature contains only superposition. Meta-algebras thus can be used to construct program algebras with various signatures of operations consisting of sets that are functionally complete in the corresponding meta-algebras. The meta-algebras of algorithms are two-sorted algebraic systems that constitute a natural generalization of the algebra of logic, which they include as the logic sort.

The present article solves the functional completeness problem for generalized Dijkstra meta-algebras (DMA) whose generator systems include loops with several exits and for other iterative meta-algebras whose generator systems contain various structured, nonstructured, and loop constructs [20-22], as well as the prediction function introduced in Glushkov’s SAA [12].

†The research received financial support from the State Foundation of Basic Research of the Ukrainian Ministry of Science and Technology, grant 14/264, and also from the science education support program of the International Renaissance Foundation.

The article is organized as follows. Section 2 presents some preliminary remarks that place our results in the context of earlier research. Section 3 generalizes the concepts and constructs from [18]. These concepts are then applied to develop a simplified proof of the functional completeness theorem for the algebra of structured schemes (ASS) — the operator sort of the DMA. Section 4 extends the results to generalized DMA whose generator system contains loops with up to $k$ exits (where $k = 2, 3, \ldots$). Section 5 proves the functional completeness theorem for the algebra of nonstructured schemes (ANS) — the operator sort of Yanov's meta-algebra (YMA). In Section 6, the functional completeness problem is investigated for the operator and logic sorts of the Glushkov meta-algebra (GMA) whose generator system contains loops and the prediction function. In conclusion we examine some future directions in the development of the algebraic principles of algorithmics.

2. PRELIMINARY REMARKS

This is a review section. The main results of the article are interpreted in the context of the general paradigm of the meta-algebra of algorithmics intended for the construction of the class of algebras of algorithms associated with a specific method of algorithm and program design. A meta-algebra (MA) in general is a many-sorted algebraic system $MA = (O; SUPER)$, where $O = \{O_i | i \in I\}$ are the MA sorts (different sets of functions dependent on variables that take values in the corresponding sorts), SUPER is the MA signature consisting of operations that include superposition of functions, as well as identification and renaming of the variables [8]. The Post algebra (PA), in particular, is an example of a two-sorted meta-algebra: this is a two-valued algebra of logic $AP = (L(2); SUPER)$, where $L(2)$ is the set of all Boolean functions.

The fundamental problems of the algebra of logic include construction of the lattice of its subalgebras and investigation of the main properties of this lattice. For the two-valued case this problem has been solved by Post [1, 2] with the following main results:

- the set of all subalgebras (closed classes of the algebra of logic) is countable;
- these subalgebras are finitely generated;
- the inclusion diagram has been constructed for the subalgebras of the algebra $L(2)$.

A solution of the functional completeness problem thus has been obtained for the algebra of logic as a whole by constructing the family of its maximal subalgebras (precomplete closed classes) and for each subalgebra (closed class) separately by describing the maximal subalgebras with respect to the particular subalgebra.

**THEOREM 1** (functional completeness criterion for PA) [1, 2]. The system $SYST \subseteq L(2)$ is complete in PA if and only if it is not included in any of the following maximal subalgebras of PA: $T_0$, $T_1$, $S$, $M$, $L$. Here $T_0$ and $T_1$ are sets of all functions preserving the constant 0 and 1, respectively; $S$ is the set of self-dual Boolean functions; $M$ is the set of monotone Boolean functions, and $L$ is the set of linear Boolean functions.

Important chapters of the algebra of logic include construction and investigation of various algebras of Boolean functions: Boolean algebra, Zhigalkin algebra, and also algebras whose signatures constitute functionally complete systems of the algebra of logic.

The variety of algebras of Boolean functions is determined by the following factors:

- these algebras have many important applications for the design of combinational networks from various sets of functional elements;
- many different methods and language tools are available for the design of combinational networks;
- there is a rich variety of existing and expected technological environments, etc.

The fundamental importance of Post's theorem is in that it presents the functional completeness criterion for the design of sought functionally complete systems and embeds them in signatures of operations of the corresponding algebras of Boolean functions. These algebras require further study in connection with the axiomatization problem, construction of the theory of normal forms, normal form minimization, etc.

Note that the apparatus of the algebra of logic has important applications for the description of the logical conditions associated with the execution of various algorithmic processes, in particular, writing programs in modern programming languages. We use the algebra of logic and Post's theory as a specimen and a composite component of various meta-algebras of algorithmics intended for the construction of families of algebras of algorithms associated with modern algorithm and program design methods. Our study is limited to the two-sorted case, which involves analysis of operator and logical constructs. Note, however, that the results can be recast for the description of data structures (see, e.g., [23-25]). This indi-