

Catalog of SETL(C) Newsletters as of July 30, 1973

1. L.V. Gorodnyaya, D.Y. Levin, L.V. Chernobrod.
Phase 1 implementation Project. (9 pages).
A point of view concerning the aim of an implementation of a set-theoretic language is outlined. Certain interconnected problems which arise in such an implementation are developed.
2. L.V. Chernobrod. A conceptual review of the set-theoretic language SETL (41 pages). (Published in the Collection: System and Theoretical Programming, Novosibirsk, 1972.)
Description of a preliminary version of the SETL language. Examples of simple algorithms are given.
3. L.V. Chernobrod. LISPSETL user's manual Version 1, (25 pages + 21). LISPSETL is an extension of LISP which includes a large part of the semantics of the language SETL but is syntactically limited by the framework of the LISP language. LISPSETL consists of a library of routines which implement the basic operations of SETL.

LISPSETL'S limitations with respect to SETL include: the absence of macros and some differences in the description in the semantics of transfer operations and iterations.

4. D.Y. Levin. The BALM Input language of the system BALM-BESM/6 (29 pages). A syntactic description of BALM is given with semantic commentaries. The description is stated in terms of "expressions". A syntactically (semantically) correct program is a quasi-expression; executing the program is the same as computing the value of this expression, which may also result in certain side effects.

5. D.Y. Levin. The system BALM-BESM/6 (19 pages + 37). A BESM/6 implementation of the language BALM is described. The system consists of two modules: The BALM-machine, (An interpreter of an internal language) and a program for a BALM translator, which maps BALM into the internal language. Both modules are written in the language EPSILON: This leads to an efficient but not machine independent variant of the overall system.

6. D.Y. Levin. SETL: Source language of an experimental implementation (22 pages). This report describes a set-theoretic programming language which can be regarded as a variant of SETL. Differences include: Lack of compound operators, no complex left parts in assignments, extended looping constructs, introduction of variable declarations, etc. This description corresponds to a minimal source language since it provides primitive concepts which will be treated in special reports and whose exclusion from the language does not affect other constructions.

7. D.Y. Levin. Internal representation of sets and an experimental implementation of set-theoretic operations. (34 pages). A classification of sets is suggested, and also a strategy using this classification. The strategy associates a manner of description with each set, and using this information chooses a specific method for performing set theoretic operations in a particular case. The type of sets can be given directly or deduced from the context in which they appear. Since a mechanism is provided for the introduction of new types, there exist a potentially infinite number of types of sets.

The classification of the sets is a necessary preliminary to the development of a library of methods of programming sets and operations on them. This classification is also studied as part of the problem of optimizing SETL programs.

8. D.Y. Levin. The SETLBALM system and a preliminary implementation of SETL (10 pages). We describe the extension of the BALM system to a system translating BALM as well as SETL-programs into a general internal language in which one is able to express the semantics of SETL sets. In the extended system, a program of the internal language are executed on a 'SETL-machine'. It is also suggested that a module mapping programs of the internal language into EPSILON-programs be made part of the system.

9. D.Y. Levin. Decomposition of SETL programs (15 pages). A system of precedences for SETL operations is suggested. This is compatible with the system of precedences used in BALM in the sense that it supports a parsing scheme that at the same time includes both BALM and SETL. The system of precedences is based on a classification of SETL operations, as described in the report.

10. L.V. Gorodnaya. Interaction of data types (30 pages). A conceptual memorandum on the design of a system allowing convenient treatment of data.

11. Catalog of SETL(C) Newsletters (1-12) (3 pages).

12. A. S. Narinyani. Parallel programming and high level languages (6 pages). This report considers the possibility of including means for the description of parallel calculations in an algorithmic language.

It is suggested that SETL should be extended by including into SETL a semantic construct called an *A-program*.

An A-program is a set of pairs $\langle x, y \rangle$, where x is an "operator" and y is a two-valued "trigger" function. The notion of A-program execution is defined. Results of the theory of parallel programming show that A-programs may serve to express processes of maximal asynchronity.

The A-program notion is exemplified by showing a fragment of the proposed extended SETL language as well as an example indicating how an algorithm can be written as an A-program.