

t h e s e t l p r o j e c t - m a s t e r c a t a l o g

- part 1 - major documents.
- part 2 - SETL newsletters.
- part 3 - LITTLE newsletters.
- part 4 - the SETL algorithms library.
- part 5 - the SETL test packages.

subscribers, please note:

up to five back issues of SETL or LITTLE newsletters are available free on request in single copies. those wishing to be added to the regular mailing list maintained for these newsletters should write to:

SETL Project
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, New York 10012

enclosing a check for ten dollars for a one year subscription which should be made out to New York University.

Certain of our longer documents, listed on the next page, are available direct from our publications co-ordinator.

Write to:

Lenora Greene, SETL Publications Co-ordinator
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012

1. On Programming. An interim report on the SETL project.
J. Schwartz

- part 1: generalities.
- part 2: the SETL language, and examples of its use.

(revised) June 1975 xii + 675 pp

- part 3: extended facilities of the SETL language.
(to appear)

2. A SETLB primer.

H. Mullish
M. Goldstein

a step-by-step tutorial with over 100 illustrative programettes.

June 1973

v + 201 pp

3. The SETL run-time library.

This is the run-time support system for SETL. It is written in LITTLE and is well documented internally. It supports all of the main set-theoretic primitives of SETL and is of prime interest to those wishing to develop or modify the SETL system. The run-time library is available in machine readable form.

4. Courant computer sciences reports.

1. ASL: a proposed variant of SETL.

(out of print)

H. Warren

May 1973

xi+326 pp

2. A metalanguage for expressing grammatical restrictions in nodal spans parsing of natural language.

J. Hobbs

January 1974

266 pp

3. Type determination for very high level languages.

(out of print)

A. Tenenbaum

October 1974

v+171 pp

5. Investigations in the theory of descriptive complexity.

W. Gewirtz

October 1974

60 pp

6. Operating system specification using very high level dictions.

(out of print)

P. Markstein

June 1975

152 pp

7. Directions in artificial intelligence: natural language processing.

(out of print)

R. Grishman (ed.)

August 1975

109 pp

8. A survey of syntactic analysis procedures for natural language.
(out of print) R. Grishman
August 1975 95 pp
9. Scene analysis: a survey. C. Weiman
December 1975 62 pp
10. A hierarchical technique for mechanical theorem proving
and its application to programming language design.
(out of print) N. Rubin
November 1976 172 pp
11. Making computational sense of Montague's intensional
logic. J. Hobbs
S. Rosenschein
January 1977 41 pp
12. Correct program technology / extensibility of verifiers.
M. Davis
J. Schwartz
E. Deak (appendix)
September 1977 146 pp
13. Groups with Solvable Word Problems. C. Semeniuk
March 1979 77 pp
14. Automatic Storage Optimization. J. Fabri
June 1979 94 pp
15. Data Structure Choice/Formal Differentiation: Two Papers on
Very High Level Program Optimization. S. Liu
R. Paige
September 1979 658 pp
16. On the Complexity of the Satisfiability Problem.
A. Goldberg
October 1979 85 pp
17. A Design for Optimizations of the Bitvectoring Class.
J. Schwartz
M. Sharir
September 1979 117 pp
18. Automatic Discovery of Heuristics for Nondeterministic
Programs from Sample Execution Traces. S. Stolfo
September 1979 172 pp
19. Computing Chromatic Polynomials for Special Families of
Graphs. B. Loeferinc
February 1980 111 pp

20. The Language REFAL - The Theory of Complication and Metasystem
Analysis. V. Turchin
February 1980 245 pp

5. The LITTLE system manual.

E. Deak
D. Shields

the system manual for the LITTLE language addresses the following needs: a system programmer's reference manual giving details of compiler structure, installation and maintenance; an algorithmic description of the compiler structure; a guide to extending the compiler to support new language features; a case study of an actual implementation and of the problems encountered in constructing machine independent software; a pragmatic reference document for a compiler-writing course.

6. Back issues of newsletters.

Subscribers who want complete, unbound back issues of newsletters should write to SETL Project at the address given on page 1, enclosing a check made out to New York University to cover cost as follows:

SETL newsletters - part 2 of this catalog.	price \$50.00
LITTLE newsletters - part 3 of this catalog.	price \$20.00

7. various machine readable information.

Those interested in acquiring any of the following documents in machine readable form should address an inquiry concerning price to SETL Project at the address given on page 1.

SETL newsletter 49 -
detailed specifications of certain SETL operations.
described in part 6 of this catalog.
part 2 - SETL newsletters.

Following is an inclusive, up-to-date list of the SETL newsletters. Many of our earliest newsletters are not relevant to our current goals and have therefore been marked: obsolete near the right hand margin of the pages on which they appear. Obsolete newsletters are available by special request only - they will not be provided in standard orders for back issues.

1. BALM-SETL -- a simple implementation of SETL.

November 1970

8 pp

M. Harrison

obsolete

2. No longer available.

3. Modifications and extensions for SETL, part 1.
November 1970 6 pp D. Shields obsolete
4. An APL version of Peter Markstein's McKeeman table routine.
November 1970 2 pp P. Markstein
5. Miscellaneous algorithms written in SETL.
November 1970 7 pp J. Schwartz obsolete
6. A revised SETL version of the McKeeman parse.
November 1970 3 pp P. Markstein obsolete
7. Modifications and extensions for SETL, part 2.
November 1970 8 pp D. Shields obsolete
8. Additional miscellaneous SETL algorithms.
November 1970 4 pp J. Schwartz obsolete
9. Implementation and language design.
December 1970 5 pp M. Harrison obsolete
10. A sorting algorithm.
December 1970 3 pp K. Maly obsolete
11. Modifications and extensions for SETL, part 3.
December 1970 4 pp D. Shields obsolete
12. Recapitulation of the basic parts of the SETL language.
January 1971 14 pp J. Schwartz obsolete
13. Additional miscellaneous algorithms.
January 1971 7 pp J. Schwartz obsolete
14. Additional syntactic extensions.
January 1971 3 pp J. Schwartz
15. A proposed SETL implementation plan through the end of the
bootstrap phase.
February 1971 2 pp J. Schwartz obsolete
16. SETL 64-character set -- 48-character set / 025 keypunch -- cdc

- 6600 64-character set / 023 keypunch. K. Maly
February 1971 2 pp obsolete
17. No longer available.
18. Preliminary specification of BALMSETL conventions.
February 1971 2 pp D. Shields obsolete
19. Lexical description of SETL.
February 1971 5 pp K. Maly obsolete
20. BALMSETL user's guide (in brief).
March 1971 4 pp D. Shields obsolete
21. An outside review: comments on the SETL draft.
April 1971 8 pp anon. (publisher)
22. Some small and large language extensions for consideration.
April 1971 4 pp J. Schwartz
23. Current status of BALMSETL implementation.
April 1971 3 pp D. Shields obsolete
24. Description of a register allocation algorithm.
April 1971 8 pp K. Kennedy
25. A print routine.
April 1971 3 pp B. Loeferinc
26. The currently specified form of SETL from a more fundamental
point of view.
May 1971 8 pp J. Schwartz
27. Code for the postparse setup procedure (postparse metalanguage
analysis).
May 1971 15 pp J. Schwartz obsolete
28. An algorithm for common subexpression elimination and code
motion.
May 1971 10 pp K. Kennedy
29. Some issues connected with subroutine linkage.
May 1971 3 pp J. Schwartz
30. Sinister calls.

- | | | | | |
|-----|--|--------|--------------------------|----------|
| | May 1971 | 32 pp | J. Schwartz | |
| 31. | An additional preliminary remark on the importance of "object types" for SETL, with some reflections on the motion of "data structure language". | | J. Schwartz | |
| | May 1971 | 12 pp | | |
| 32. | Hyper-SETL procedural languages. | | J. Schwartz | |
| | May 1971 | 5 pp | | |
| 33. | What is programming. | | J. Schwartz | |
| | May 1971 | 9 pp | | |
| 34. | Syntax revisions in preparation for implementation. | | J. Schwartz | |
| | May 1971 | 12 pp | | obsolete |
| 35. | A new form for the IFF-statement. | | D. Shields | |
| | May 1971 | 7 pp | | |
| 36. | No longer available. | | | |
| 37. | Initial description of an algorithm for use-definition chaining in optimization. | | P. Owens
K. Kennedy | |
| | July 1971 | 6 pp | | |
| 38. | An algorithm for live-dead analysis including node-splitting for irreducible program graphs. | | K. Kennedy
k. kennedy | |
| | January 1972 | 10 pp. | | |
| 39. | More detailed suggestions concerning "data strategy" elaborations for SETL. | | J. Schwartz | |
| | May 1971 | 54 pp | | |
| 40. | No longer available | | | |
| 41. | Additional planning detail for the current and next phase of SETL implementation. | | J. Schwartz | |
| | June 1971 | 4 pp | | obsolete |
| 42. | Revised conventions concerning tuples. | | J. Schwartz | |
| | June 1971 | 6 pp | | obsolete |
| 43. | A parsing scheme for FORTRAN. | | S. Gruber | |
| | July 1971 | 20 pp | | obsolete |
| 44. | Comprehensive SETL specifications. | | K. Maly | |

- | | | | |
|---|----------------|--------|----------------|
| 45. Semi-local SETL optimization. | July 1971 | 11 pp | D. Shields |
| | July 1971 | 8 pp | |
| 46. Generalized nodal span parse routine - preliminary draft. | | | J. Schwartz |
| | July 1971 | 12 pp | |
| 47. An outline for a parsing scheme for SETL. | | | K. Maly |
| | July 1971 | 3 pp | |
| 48. Toward a documentation of the string project's program for parsing english sentences. | | | J. Hoops |
| | August 1971 | 19 pp | obsolete |
| 49. Specification of the SETL run time library (revision 2). | | | H. Warren |
| | April 1973 | 138 pp | |
| 50. A three-phase parsing scheme for SETL. | | | K. Maly |
| | September 1971 | 3 pp | |
| 51. No longer available. | | | |
| 52. Comments on SETL. | | | J. Earley |
| | September 1971 | 8 pp | |
| 53. SETL to LITTLE translator: a first look. | | | H. Warren |
| | September 1971 | 26 pp | |
| 54. Current status of BALMSETL-4. | | | S. Gruber |
| | September 1971 | 5 pp | |
| 55. SETL suggestions and questions. | | | S. Finkelstein |
| | September 1971 | 5 pp | |
| 56. Additional comments on some basic SETL operations. | | | J. Earley |
| | September 1971 | 4 pp | |
| 56a. More comments on SETL. | | | J. Earley |
| | October 1971 | 5 pp | |
| 56b. More SETL comments. | | | J. Earley |

- | | | | | |
|---|---------------|-------|-------------|----------|
| 57. Minimizing copying in SETL: preliminary observations. | October 1971 | 10 pp | H. Warren | |
| | October 1971 | 3 pp | | obsolete |
| 58. Phase one of the SETL compiler. | | | K. Maly | |
| | October 1971 | 21 pp | | |
| 59. An algebra of assignment. | | | R. Krutar | |
| | October 1971 | 25 pp | | |
| 60. SETL compiled code: calls to SETL procedures. | | | H. Warren | |
| | November 1971 | 48 pp | | |
| 61. Syntactic structure of SETL. | | | K. Maly | |
| | November 1971 | 14 pp | | |
| 62. Final specification of SETL and parser. | | | K. Maly | |
| | December 1971 | 20 pp | | |
| 63. The SETL print routine. | | | G. Fisher | |
| | January 1972 | 5 pp | | |
| 64. SETL compiler with elaborated data structures. | | | K. Maly | |
| | January 1972 | 33 pp | | |
| 65. Some notational suggestions. | | | R. Bonic | |
| | February 1972 | 2 pp | | |
| 66. BALMSETL user's manual version 1.0 (revised 8/72). | | | E. Milgrom | |
| | February 1972 | 71 pp | | obsolete |
| 67. Data structures of the SETL compiler from the LITTLE version. | | | K. Maly | |
| | February 1972 | 16 pp | | |
| 68. Some thoughts on efficient programming in SETLB. | | | S. Brown | |
| | October 1972 | 5 pp | | |
| 69. The SETL project - master catalog (revised 2/79). | | | R. Abes | |
| | February 1973 | 39 pp | | |
| 70. The SETLA user's manual (revised 1/75). | | | J. Schwartz | |

- | | | | |
|--|----------------|-------|---|
| | August 1974 | 70 pp | S. Brown
E. Schonberg |
| 71. Deducing the logical structure of objects occurring in SETL programs. | April 1972 | 14 pp | J. Schwartz |
| 72. An introductory explanation of SETL: a status review and profile of the SETL user group. | April 1972 | 13 pp | D. Shields |
| 73. User's guide to the SETL run-time library.
(revision 3 - May 1973) | April 1972 | 34 pp | K. Maly
H. Warren |
| 74. Project plan for first stage of implementation.
(partial translation from the russian) | May 1972 | 3 pp | L. Gorodnaya
D. Levin
V. Chernobrod |
| 75. Some thoughts on the use of BALM to implement SETL.
(this is also BALM bulletin no. 13) | June 1972 | 7 pp | E. Milgrom |
| 76. Semantic definition matters. | May 1973 | 91 pp | J. Schwartz
G. Jennings |
| 77. Transferring SETLB to other machines. | September 1972 | 1 pp | J. Schwartz |
| 78. Executing BALM and SETLb at NYU Courant. | September 1972 | 2 pp | R. Paige |
| 79. No longer available. | | | |
| 80. Algorithms in the SETLB test package. | September 1972 | 3 pp | K. Curtis |
| 81. Memory size of SETLB runs. | September 1972 | 1 pp | J. Schwartz
s. brown |
| 82. Timing comparison between SETLB and FORTRAN. | October 1972 | 2 pp | E. Desautels |
| 83. User experience and human factors. | November 1972 | 16 pp | J. Schwartz |
| 84. Plan for a library of algorithms. | | | |

- | | | | |
|---|---------------|-------|--------------|
| | November 1972 | 18 pp | J. Schwartz |
| 85. Estimate of minimum running size for the next SETL system (revision 1). | | | H. Warren |
| | April 1973 | 6 pp | |
| 86. Proposal for a temporary, but easily implemented, software paging system. | | | J. Schwartz |
| | November 1972 | 7 pp | |
| 87. Workplan for the next phase of SETL implementation. | | | J. Schwartz |
| | November 1972 | 7 pp | |
| 88. A scheme for BALMSETL measurements. | | | J. Schwartz |
| | November 1972 | 2 pp | |
| 89. User information for lexical scan setup package. | | | E. Guth |
| | November 1972 | 2 pp | t. polacek |
| 90. Preliminary reflections on the use of SETL in a data-base context. | | | J. Schwartz |
| | December 1972 | 18 pp | |
| 91. A grammarless parse and a related method of retrieval by similarity. | | | J. Schwartz |
| | December 1972 | 24 pp | |
| 92. Some experiments with SETL3 programs. | | | K. Curtis |
| | January 1973 | 9 pp | |
| 93. A note on optimization and programming style in SETL. | | | K. Curtis |
| | January 1973 | 2 pp | |
| 94. An algorithm to represent a collection of sets as intervals (on a line). | | | G. Jennings |
| | January 1973 | 9 pp | |
| 95. Generalized nodal span parse routine, corrected version. | | | Y. Feinroth |
| | January 1973 | 10 pp | |
| 96. Pointers and 'very high level languages'. | | | N. Minsky |
| | January 1973 | 3 pp | |
| 97. SETL extensions for operating system description. | | | P. Markstein |
| | February 1973 | 24 pp | |

98. Reflections on P. Markstein's newsletter on SETL extensions for operating system description. J. Schwartz
January 1973 9 pp
99. Paging, the quick and dirty way.
(this is also balm bulletin no. 21) S. Brown
January 1973 4 pp
100. Making SRTL debugging runs. H. Warren
February 1973 11 pp
101. How to program if you must (the SETL style). R. Bonic
March 1973 15 pp
102. Reduction in strength using hashed temporaries. K. Kennedy
March 1973 12 pp
103. Preliminary plan for BALM-to-LITTLE translator. J. Schwartz
April 1973 8 pp
104. An algorithm to represent a collection of sets as a direct product of intervals on the line. G. Jennings
March 1973 9 pp
105. A SETL program for a basic block optimizer and an extended basic block optimizer. S. Marateck
April 1973 11 pp J. Schwartz
106. User variation of the semantics of function and subroutine invocation. G. Jennings
May 1973 3 pp
107. Linear function test replacement. K. Kennedy
May 1973 5 pp
108. APL - SETL, an extension of SETL achieved from user varied semantics. G. Jennings
May 1973 34 pp
109. Faster execution for the LITTLE based balm system. S. Brown
July 1973 4 pp
111. Global dead computation elimination. K. Kennedy
August 1973 8 pp
112. An algorithm to compute compacted use-definition chains. K. Kennedy

- August 1973 8 pp
 113. LITTLE code generation from the BALM compiler.
 S. Brown
- August 1973 10 pp
 114. A SETLB to publication SETL translator.
 A. Getzler
- August 1973 7 pp
 115. A SETL representation of the Maryland GRAAL graph-manipulation
 language.
 G. Weinberger
 A. Tenenbaum
- August 1973 32 pp
 116. Catalog of SETL(c) newsletters as of July 30, 1973.
 (this is also SETL(c) newsletter 11.)
 A. Ershov et al
 (Novosibirsk group)
- August 1973 5 pp
 117. A static debugging system for LITTLE.
 E. Schonberg
- October 1973 12 pp
 118. Revised and extended algorithms for deducing the types of
 objects occurring in SETL programs.
 A. Tenenbaum
- October 1973 21 pp
 119. A suggested generalization and revision of the SETL compound
 operator form.
 J. Schwartz
- October 1973 2 pp
 120. A general-recursive extension of functional application and its
 uses.
 J. Schwartz
- December 1973 3 pp
 121. An algorithm to determine the identity of SETL run-time objects.
 A. Tenenbaum
- January 1974 15 pp
 122. More local and semi-local SETL optimisations.
 J. Schwartz
- January 1974 6 pp
 122A. A few peephole optimisations applicable to iterators.
 J. Schwartz
- July 1974 2 pp
 122b. Still more miscellaneous optimisations.
 J. Schwartz
- July 1974 6 pp
 123. Variable subsumption with constant folding.
 K. Kennedy
- February 1974 13 pp
 124. The VERS2 language of J. Earley considered in relation to SETL.

- | | | | |
|--|---------------|--------|--------------------------|
| | January 1974 | 6 pp | E. Schonberg |
| 125. Schaefer's node splitting algorithm. | | | |
| | February 1974 | 14 pp | K. Kennedy |
| 126. A SETLB specification of EDIT. | | | |
| | February 1974 | 112 pp | M. Brenner |
| 127. Edge-listing data-flow algorithms. | | | |
| | March 1974 | 9 pp | K. Kennedy |
| 128. A LITTLE written translator from SETL to LITTLE. | | | |
| | April 1974 | 9 pp | E. Schonberg
s. brown |
| 129. More on SETL in a data-base environment. | | | |
| | May 1974 | 43 pp | J. Schwartz |
| 130. Deducing relationships of inclusion and membership in SETL programs. | | | |
| | May 1974 | 36 pp | J. Schwartz |
| 130a. Estimates from below the domain of a mapping. | | | |
| | August 1974 | 7 pp | J. Schwartz |
| 130b The use of equalities in the deduction of inclusion/membership relations. | | | |
| | May 1975 | 12 pp | J. Schwartz |
| 131. More on copy optimisation of SETL programs. | | | |
| | June 1974 | 51 pp | J. Schwartz |
| 132. Some optimisations using type information. | | | |
| | June 1974 | 3 pp | J. Schwartz |
| 133. A higher level control diction. | | | |
| | June 1974 | 6 pp | J. Schwartz |
| 133A. Additional pursue block examples. | | | |
| | June 1974 | 5 pp | J. Schwartz |
| 133b. General comments on high level dictions, and specific suggestions concerning !converge! iterators and some related dictions. | | | |
| | January 1975 | 18 pp | J. Schwartz |

134. Inter-procedural optimisation.
 July 1974 24 pp J. Schwartz
135. Introductory lecture at the June 28 "informal optimisation symposium".
 July 1974 14 pp J. Schwartz
- 135a. Structureless programming -or- the notion of "rubble" and the reduction of programs to rubble.
 July 1974 25 pp J. Schwartz
- 135b. Additional thoughts on "language level" and optimisation.
 January 1975 2 pp J. Schwartz
- 135c. On the genesis of complex programs.
 August 1975 4 pp J. Schwartz
136. A framework for certain kinds of high-level optimisation.
 July 1974 4 pp J. Schwartz
137. Additional thoughts concerning automatic data structure choice.
 July 1974 12 pp J. Schwartz
138. On Jay Earley's "method of iterator inversion".
 in progress J. Schwartz
 R. Paige
- 138a. Optimisation by set suppression.
 August 1974 3 pp J. Schwartz
- 138b. Updating the lower bound of a set of integers in set-theoretic strength reduction.
 January 1976 4 pp J. Schwartz
140. Use-use chaining as a technique in typefinding.
 September 1974 5 pp J. Schwartz
141. Reflections on some very high level dictions having an english / "automatic programming" flavor.
 January 1975 18 pp J. Schwartz
142. What programmers should know.
 January 1975 5 pp J. Schwartz

143. "Arguments from use" in the proof of relationships of inclusion and membership. J. Schwartz
February 1975 2 pp
144. Interprocedural live-dead analysis. J. Schwartz
February 1975 3 pp
145. GYVE-oriented inter-process coordination and control structures for an extended SETL (SETLG). J. Schwartz
March 1975 17 pp
146. Adaption of GYVE/SETLG to distributed networks of computers. J. Schwartz
March 1975 5 pp
147. A syntactic construct useful for checking parameters. J. Schwartz
March 1975 4 pp
148. Technical and human factors improvements for the fully compiled SETL system. E. Schonberg
April 1975 7 pp a. stein
149. Conventions allowing other languages to be used within GYVE; files; memory heirarchy questions; some suggestions for GYVE extensions. J. Schwartz
April 1975 28 pp
150. What constitutes progress in programming. J. Schwartz
May 1975 5 pp
151. Additional considerations concerning semi-automatic data structure choice. J. Schwartz
July 1975 37 pp
152. An alternative design for a "MIDL"-level language. J. Schwartz
July 1975 4 pp
153. The significance of "backtracking", and its cost. J. Schwartz
July 1975 20 pp
154. Timing considerations for the SETL translator system. E. Schonberg
July 1975 6 pp
155. Intermediate result recording and other techniques for optimizing recursions and backtrack programs. J. Schwartz
October 1975 26 pp

156. The next phase of our work.
August 1975 4 pp J. Schwartz
157. "Whenever" dictions.
August 1975 24 pp J. Schwartz
158. Implementation of reference counts in the SETL system.
October 1975 4 pp E. Schonberg
159. On the "base form" of algorithms.
November 1975 15 pp J. Schwartz
160. An algebra of program events potentially useful
in a debugging language.
November 1975 5 pp J. Schwartz
161. Specifications for a new optimizer-oriented SETL front end.
December 1975 13 pp A. Grand
162. Improved target code forms available in the presence of
global information concerning a SETL program.
January 1976 20 pp
163. Recognizing comparability graphs in SETL.
March 1976 9 pp M. Golumbic
164. Copy optimization in SETL.
April 1976 3 pp J. Schwartz
- 164a. "Copy on assignment" optimization in SETL.
April 1976 9 pp R. Dewar
165. A simple criterion for avoiding basing errors.
April 1976 1 pp J. Schwartz
166. An easy scheme for incorporating backtracking into the
new SETL implementation.
April 1976 13 pp J. Schwartz
R. Dewar
167. A variant SETL implementation incorporating "whenever" dictions.
April 1976 7 pp J. Schwartz

168. The trouble with triples.
 April 1976 7 pp R. Dewar
169. Some changes to the SETL Language in preparation for the optimizer implementation.
 April 1976 14 pp R. Dewar
 A. Grand
 E. Schonberg
 L. Vanek
170. Provisional plan for the SETL optimizer interface.
 April 1976 3 pp R. Dewar
 A. Grand
 E. Schonberg
 L. Vanek
171. More on basings.
 April 1976 29 pp J. Schwartz
- 171a. Still more on basings.
 December 1976 10 pp J. Schwartz
- 171b. Remark on the implementation of the basing scheme.
 December 1976 3 pp J. Schwartz
- 171c. Implementation of base assignments.
 April 1977 9 pp A. Grand
172. A case statement for SETL.
 may 1976 3 pp R. Dewar
 A. Grand
 E. Schonberg
 L. Vanek
173. Simplifying and extending the SETL type calculus.
 April 1976 14 pp L. Vanek
174. Relaxing of basing restrictions.
 July 1976 3 pp R. Dewar
175. More on copy optimization.
 July 1976 4 pp S. Lfu
 E. Schonberg
176. A coarser, but simpler and considerably more efficient copy optimization technique.
 J. Schwartz

177. August 1976 3 pp
Measurement utilities for the optimized SETL system.
J. Schwartz
178. August 1976 2 pp
Motion of range checks out of loops; optimization of
integer arithmetic.
J. Schwartz
179. August 1976 11 pp
Dynamic multiple member basings.
S. Liu
180. October 1976 6 pp
Uncovering profitable basing relations.
S. Liu
E. Schonberg
181. February 1977 15 pp
A reformulation of value-flow analysis.
W. Tsui
182. March 1977 33 pp
Linkage conventions for the SETL optimizer.
L. Vanek
183. November 1976 8 pp
Some revisions of basing semantics and implementation.
R. Dewar
A. Grand
E. Schonberg
J. Schwartz
184. December 1976 3 pp
Extending the notion of basing.
R. Dewar
185. March 1977 2 pp
Using output from the SETL copy optimizer.
L. Vanek
185. March 1977 6 pp
Syntax and semantics of a restricted backtrack implementation.
R. Dewar
J. Schwartz
187. April 1977 9 pp
On inter-procedural flow analysis.
M. Sharir
- 187a. April 1977 14 pp
More on inter-procedural data flow analysis.
M. Sharir
- 187a. May 1977 8 pp

188. A limited form of common subexpression elimination for SETL programs.
L. Vanek
May 1977 15 pp
190. The implementation of backtracking.
A. Grand
May 1977 14 pp
191. More language changes.
A. Grand
May 1977 4 pp
192. 6600, 370, and PUMA microcode nubbins.
A. Grand
J. Schwartz
R. Kenner
May 1977 13 pp
193. On a static scheme to find procedure variables.
M. Sharir
May 1977 5 pp
194. Nondeterminism, backtracking, and pattern matching in SETL.
S. Rapps
June 1977 15 pp
195. An algorithm for copy optimization, based on NL 176.
M. Sharir
July 1977 13 pp
196. Current state of the SETL implementation.
A. Grand
August 1977 2 pp
197. Some comments on extending code motion and expression
availability algorithms for the SETL optimizer.
M. Sharir
September 1977 6 pp
198. String primitives.
R. Dewar
January 1978 5 pp
199. non-propagation of errors - a modified type-finding
algorithm.
M. Sharir
January 1978 5 pp
200. Possible additional reprs for the SETL system.
J. Schwartz
February 1978 7 pp

201. On compaction on re-paths.
February 1978 3 pp M. Sharir
202. Stropping and character-set conventions.
July 1978 3 pp H. Lewis
203. A simplified approach to automatic data structure choice.
January 1978 10 pp M. Sharir
204. Tarjan's fast interval finding algorithm.
March 1978 12 pp M. Sharir
J. Schwartz
205. On disjointness detection in automatic data structure choice.
July 1978 8 pp M. Sharir
206. On name splitting in SETL optimization.
February 1978 16 pp A. Grand
M. Sharir
207. A second simplified approach to automatic data structure choice.
March 1978 21 pp M. Sharir
208. A few cautionary notes on the convergence of iterative data-flow analysis algorithms.
April 1978 9 pp M. Sharir
209. Automatic data structure choice.
March 1978 21 pp E. Schonberg
210. Remarks on debugging.
February 1979 8 pp J. Schwartz
211. The SETL Character Set - The Final Decisions.
March 1979 4 pp R. Dewar
A. Grand
212. A Note on Program Genesis.
September 1979 4 pp J. Schwartz
1. I/O conventions and proposal; quoted strings; octal constants; user information for improved macroprocessor.

- | | | | |
|--|---------------|-------|---|
| | October 1971 | 8 pp | A. Stein
J. Schwartz |
| 2. no longer available. | | | |
| 3. Possible future extensions to LITTLE. | | | J. Schwartz |
| | November 1971 | 16 pp | |
| 4. A LITTLE machine. | | | J. Schwartz |
| | November 1971 | 11 pp | |
| 5. User information concerning the LITTLE-to-FORTRAN translator. | | | J. Schwartz |
| | November 1971 | 3 pp | |
| 6. No longer available. | | | |
| 7. LITTLE for minicomputers. | | | T. Stuart |
| | March 1972 | 26 pp | |
| 8. No longer available. | | | |
| 9. Some suggestions for simplifying the preparation of SETL and LITTLE text: keyboard and lexical macros. D. Shields | | | |
| | March 1972 | 6 pp | |
| 10. Interspersing macros. | | | J. Schwartz |
| | April 1972 | 9 pp | |
| 11. Input / output statements for LITTLE. | | | R. Abes
H. Warren
E. Milgrom obsolete |
| | April 1972 | 21 pp | |
| 12. No longer available. | | | |
| 13. Macro capabilities for structured programming. | | | R. Abes
H. Warren |
| | July 1972 | 15 pp | |
| 14. mass storage utilization in LITTLE. | | | P. Maclean |
| | July 1972 | 9 pp | |
| 15. No longer available. | | | |
| 16. Some timing statistics for LITTLE. | | | D. Shields |
| | October 1972 | 10 pp | |
| 17. Test packages for the LITTLE compiler. | | | |

- November 1972 1 pp R. Abes
18. A new array optimization for basic blocks.
November 1972 4 pp J. Schwartz
19. No longer available.
20. Remarks on the structure of the SETL run time library.
November 1972 5 pp D. Shields
21. Some proposals for improving the accessibility of the LITTLE compiler.
December 1972 9 pp D. Shields
22. Examples of LITTLE-generated code.
December 1972 5 pp D. Shields
23. Namesets: a new way to handle global variables in LITTLE.
January 1973 6 pp D. Shields
24. Proposals for the next stage of LITTLE development.
March 1973 6 pp D. Shields
25. Proposed extensions to LITTLE.
June 1973 21 pp D. Shields
26. Plan for the development of a LITTLE compiler for the BESM/6.
August 1973 3 pp L. Chernobrod
27. Multiple word items in LITTLE.
October 1973 13 pp R. Abes
28. An intermediate language for the LITTLE compiler.
October 1973 6 pp S. Brown
29. A medium-level semantic environment based on LITTLE.
September 1973 20 pp J. Schwartz
30. Interrupt handling facilities in LITTLE.
December 1973 17 pp J. Schwartz
31. Representation of BALM in LITTLE.

- | | | | |
|--|---------------|-------|---|
| | January 1974 | 4 pp | J. Schwartz
A. Stein |
| 32. Interrupt handling in LITTLE: possible revisions. | | | P. Shaw |
| | February 1974 | 20 pp | |
| 32a. Realisation of the -LITTLE- interrupt system described in newsletters nbr. 30 and 32. | | | C. McDonald |
| | January 1975 | 79 pp | |
| 33. Guide to the LITTLE language. | | | D. Shields |
| | March 1974 | 90 pp | |
| 34. Input/output statements for -LITTLE-. | | | R. Abes |
| (this will replace LITTLE newsletter 11.) | | | to appear shortly. |
| 35a. Design of a -LITTLE- code generator for the ADAGE agt-30. | | | I. Chakravarty H. Jacobs
M. Marks E. McGovern |
| | January 1975 | 39 pp | |
| 35b. -LITTLE- code generator for the IBM 1130. | | | R. Aronson M. Macias
D. Patel D. Reilly |
| | January 1975 | 64 pp | |
| 35c. -LITTLE- code generator for the UNIVAC 1108 - preliminary remarks. | | | S. Gold A. Carduso
G. Lucans |
| | January 1975 | 26 pp | |
| 35d. -LITTLE- code generator for the PDP-8. | | | R. Rosenthal A. Eng
M. Potmesil A. Fogel |
| | January 1975 | 36 pp | |
| 35e. -LITTLE- code generator for the PDP-11. | | | R. Colle D. Farkas
J. Farrelly |
| | January 1975 | 99 pp | |
| 36. Run-time considerations for MIDL. | | | E. Deak |
| | November 1974 | 16 pp | |
| 36a. Illustrative examples for MIDL. | | | M. Shimasaki |
| | November 1974 | 17 pp | |
| 37. Proposal for MIDL (GLITTLE). | | | E. Deak |

- | | | | |
|-----|--|-------|-----------|
| | September 1974 | 31 pp | |
| 38. | Guide to the MIDL language. | | G. Story |
| | August 1975 | 98 pp | |
| 39. | Post-partum reflections on the honeywell minicomputer implementation of LITTLE. minicomputer software. | | T. Stuart |
| | November 1975 | 5 pp | |
| 40. | Standardised and more efficient communication with LITTLE code generators. | | A. Grand |
| | November 1975 | 35 pp | |
| 41. | Dynamic arrays in LITTLE. | | R. Dewar |
| | December 1975 | 6 pp | |
| 42. | A parser-code generator interface. | | T. Stuart |
| | July 1976 | 26 pp | |

A library of substantial, important algorithms coded in SETLb exists on a machine readable file. the addition of other algorithms to this library, and the improvement of the documentation and performance of the algorithms that have been established, is an ongoing project. contents are as follows.

- | | | |
|----|---|---|
| 1. | deck binres (413 cards)
deck hypres (133 cards)
coder: e. schonberg | an automatic theorem prover operating on statements in the sentential calculus. produces shortest proofs via a breadth first tree search when binary resolution is used, or longer proofs in less time when hyper-resolution is employed. |
| 2. | deck typevar (543 cards)
coder: k. abdali | given the graph of a program and some information about its assignment statements, this algorithm finds the types that a variable can assume during the execution of the program. |
| 3. | deck matchup (147 cards)
coder: g. whitehead | a modification of marshall hall's algorithm for the marriage problem which will yield a maximal system of |

- p
4. deck topdata (112 cards)
deck topdwna (286 cards)
deck topdwnb (248 cards)
coder: s. marateck
s. brown
distinct representatives.
a top down parser and its input data, complete with a bootstrapping meta-compiler that operates on an extended backus normal form description of the language.
 5. deck mckeman (531 cards)
coder: i. kaye
generates the mckeeman tables (a series of generalized precedence tables) using backus normal form grammar as its input. sample input data is included.
 6. deck chomsnf (184 cards)
coder: n. anthony
removes null variables from a context free input grammar (described by its productions) and puts it into chomsky normal form. sample input data is included.
 7. deck gennspp (478 cards)
deck gennspd (7 cards)
coder: y. feinroth
a generalized nodal span parser, with attributes. sample input data is included.
 8. deck eulerqr (117 cards)
coder: h. mullish
a setlb coding of l. euler's graph tracing algorithm usually associated with the bridges of koenigsberg.
 9. deck lexgena (477 cards)
deck lexgenb (65 cards)
deck lexgenc (43 cards)
coder: t. polacek
the inputs to this mini-system are tables describing the character set, character types, and actions to be taken during the lexical scan of an arbitrary language. the output is a fortran program which is a working lexical scanner for the described language, complete with token file and error message generators.
 10. deck heurmac (30 cards)
deck heura (202 cards)
deck heurb (376 cards)
deck heurc (91 cards)
deck heurd (92 cards)
deck heure (82 cards)
deck heurf (257 cards)
coder: l. welber
six complete independent heuristic search procedures, mostly due to nilsson, preceded by a deck of macros which they all use. the algorithms are: a general path finder, a tree search, a uniform cost search, a breadth first search, a depth first search, an and/or tree search for a game strategy.
 11. deck gps (439 cards)
coder: a. getzler
a stripped down version of ernst, newell and shaw's general problem
- F

- solver program, with a sample specification for the !monkey and bananas! problem.
12. deck scgraph (100 cards) two separate algorithms to find the
 deck partree (124 cards) strongly connected regions of a di-
 deck balance (289 cards) rected graph. a program to generate
 coder: w. tsui the partitions of a number; a program
 to generate all binary trees. two
 separate algorithms for assembly line
 balancing.
13. deck poly (283 cards) a collection of routines for the
 coder: e. guth standard algebraic manipulations of
 polynomials. test input is included.
14. deck graal (651 cards) a SETL representation of the univer-
 coder: g. weinberger sity of maryland graph manipulation
 a. tenenbaum language (graal) of rheinhold, basili
 and mesztyenyi as explained in SETL
 newsletter 115. sample input data is
 included.
15. deck heurgra (209 cards) a heuristic graph search based on the
 coder: l. welber algorithm of chang and slagle.
 sample input data is included.

part 5 - the SETL test packages.

a library of test programs, coded in SETL, currently resides on a machine readable file. this library is frequently used to spot bugs in compiler modifications, and to provide some standards for timing studies. the algorithms vary widely in size, content, and coding style.

1. huffman (75 cards) - produces a huffman tree and table for unique bit string encoding given a set of characters and a frequency of use function over that set.
2. miscperm (50 cards) - contains short programs to make a sequence out of a tuple; compose two functions into one; obtain the inverse of a function; obtain the cycle form of a permutation; obtain the inverse of a permutation; obtain the inverse of a permutation given in cycle form.

3. perm (49 cards) - generates all permutations of n objects in lexicographic order.
4. median (159 cards) - finds the k-th number (in ascending order) of a given set of numbers. this algorithm, due to floyd, et al in 1971, runs in linear time as a function of the number of items in the given set.
5. pocksort (59 cards) - a radix sort in wich the items to be sorted and the radix are input parameters.
6. treeprint (198 cards) - prints binary or ordered trees in a tree-like format.
7. fordj (157 cards) - the ford-johnson tournament sort algorithm (a complicated minimum comparison sort).
8. setup and dsetup (190 cards) - reads SETL code, and prepares a string and some tables for the lexical scanner.
9. intprint (290 cards) - prints the flow-graph of a program in flowchart-like format given a set of paths and a set defining the order in which to print the nodes.
10. tweroc (35 cards) - the natural two-way merge for fast in-core sorting.
11. primes (72 cards) - contains short programs to generate primes by the seive method; generate primes directly from their definition; find the prime factors of a given number.
12. piglatin (27 cards) - string breakup and translation via table lookup or a programmed english-piglatin dictionary.
13. insanity (29 cards) - a backtracking algorithm to solve the instant insanity (colored cubes) puzzle.
14. nodspan (91 cards) - a nodal span parse routine which can apply any production grammar in chomsky normal form to an input string
15. pascal (16 cards) - a string manipulation and formatting program which prints pascal's triangle neatly.
16. erraut (104 cards) - calculates the structure of the automaton associated with error detection in !lr! parsing.
17. splash (138 cards) - solves all the old bucket problems (e.g. how to get 4 gallons of water given a 3 and a 5 gallon bucket).
18. maxflow (103 cards) - a package to find a path in an ordered graph; determine the maximum flow in a network; and apply the maximum flow algorithm to the matching problem.