BIBLIOGRAPHY

Addyman, A.M., Brewer,R., Burnett Hall, D.G. De Morgan, R.M., Findlay, W., Jackson M.I., Joslin, D.A., Rees, M.J., Watt, D.A., Welsh, J. and Wichmann, B.A. (1979) A draft description of Pascal, *Software - Practice and Experience*, **9**(5), 381-424.

Aho, A.V., Sethi, R. and Ullman, J.D. (1986) *Compilers: Principles, Techniques and Tools*, Addison- Wesley, Reading, MA.

Alblas, H. and Nymeyer, A. (1996) *Practice and Principles of Compiler Building with C*, Prentice-Hall, Hemel Hempstead, England.

Andrews, G.R. and Schneider, F.B. (1983) Concepts and notation for concurrent programming, *ACM Computing Surveys, **15**, 3-43.*

Backhouse, R.C. (1979) *Syntax of Programming Languages: Theory and Practice*, Prentice-Hall, Hemel Hempstead, England.

Bailes, P.A. (1984) A rational Pascal, *Australian Computer Journal*, **16**(4), 155-176.

Barron, D.W. (ed) (1981) *Pascal - the Language and its Implementation*, Wiley, Chichester, England.

Ben-Ari, M. (1982) *Principles of Concurrent Programming*, Prentice-Hall, Englewood Cliffs, NJ.

Bennett, J.P. (1990) *Introduction to Compiling Techniques: a First Course using ANSI C, LEX and YACC*, McGraw-Hill, London.

Bjorner, D. and Jones, C.B. (eds) (1982) *Formal Specification and Software Development*, Prentice-Hall, Hemel Hempstead, England.

Bratman, H. (1961) An alternate form of the Uncol diagram, *Communications of the ACM*, **4**(3), 142.

Brinch Hansen, P. (1983) *Programming a Personal Computer*, Prentice-Hall, Englewood Cliffs, NJ.

Brinch Hansen, P. (1985) *On Pascal Compilers*, Prentice-Hall, Englewood Cliffs, NJ.

Burns, A. and Davies, G. (1993) *Concurrent Programming*, Addison-Wesley, Wokingham, England.

Burns, A. and Welling, A. (1989) *Real Time Systems and their Programming Languages*, Addison-Wesley, Wokingham, England.

Bustard, D.W., Elder, J. and Welsh, J. (1988) *Concurrent Programming Structures*, Prentice-Hall, Hemel Hempstead, England.

Cailliau, R. (1982) How to avoid getting schlonked by Pascal, *ACM SIGPLAN Notices*, **17**(12),

31-40.

Chomsky, N. (1959) On certain formal properties of grammars, *Information and Control*, **2**(2), 137-167.

Cichelli, R.J. (1979) A class of easily computed, machine independent, minimal perfect hash functions for static sets, *Pascal News* **15**, 56-59.

Cichelli, R.J. (1980) Minimal perfect hash functions made simple, *Communications of the ACM*, **23**(1), 17- 19.

Cooper, D. (1983) *Standard Pascal Reference Manual*, Norton, New York.

Cormack, G.V., Horspool, R.N.S. and Kaiserwerth, M. (1985) Practical perfect hashing, *Computer Journal* **28**(1), 54-58.

Cornelius. B.J., Lowman. I.R. and Robson, D.J. (1984) Steady-state compilers, *Software - Practice and Experience*, **14**(8), 705-709.

Cornelius, B.J. (1988) Problems with the language Modula-2, *Software - Practice and Experience*, **18**(6), 529- 543.

Dobler, H. and Pirklbauer, K. (1990) Coco-2 - a new compiler-compiler, *ACM SIGPLAN Notices*, **25**(5), 82- 90.

Dobler, H. (1991) Top-down parsing in Coco-2, *ACM SIGPLAN Notices*, **26**(3), 79-87.

Earley, J. and Sturgis, H. (1970) A formalism for translator interactions, *Communications of the ACM*, **13**(10), 607-617.

Elder, J. (1994) *Compiler Construction: a recursive descent model*, Prentice-Hall, Hemel Hempstead, England.

Ellis, M.A. and Stroustrup, B. (1990) *The Annotated C++ Reference Manual*, Addison-Wesley, Reading, MA.

Fischer, C.N. and LeBlanc, R.J. (1988) *Crafting a Compiler*, Benjamin Cummings, Menlo Park, CA.

Fischer, C.N. and LeBlanc, R.J. (1991) *Crafting a Compiler with C*, Benjamin Cummings, Menlo Park, CA.

Gough, K.J. (1988) *Syntax Analysis and Software Tools*, Addison-Wesley, Wokingham, England.

Gough, K.J. and Mohay, G.M. (1988) *Modula-2: A Second Course in Programming*, Prentice-Hall, Sydney, Australia.

Grosch, J. (1988) Generators for high-speed front ends, *Lecture Notes in Computer Science*, 371, 81-92, Springer, Berlin.

Grosch, J. (1989) Efficient generation of lexical analysers, *Software - Practice and Experience*,

19(11), 1089- 1103.

Grosch, J. (1990a) Lalr - a generator for efficient parsers, *Software - Practice and Experience*, **20**(11), 1115- 1135.

Grosch, J. (1990b) Efficient and comfortable error recovery in recursive descent parsers, *Structured Programming*, **11**, 129-140.

Grune, D. and Jacobs, C.J.H. (1988) A programmer-friendly LL(1) parser generator, *Software - Practice and Experience*, **18**(1), 29-38.

Hennessy, J.L. and Mendelsohn, N. (1982) Compilation of the Pascal case statement, *Software - Practice and Experience*, **12**(9), 879-882.

Hennessy, J.L. and Patterson, D.A. (1990) *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, San Mateo, CA.

Hoare, C.A.R. and Wirth, N. (1973) An axiomatic definition of the programming language Pascal, *Acta Informatica*, **2**, 335-355.

Holmes, J. (1995) *Object-Oriented Compiler Construction*, Prentice-Hall, Englewood Cliffs, NJ.

Holub, A.I. (1990) *Compiler Design in C*, Prentice-Hall, Englewood Cliffs, NJ.

Hunter, R.B. (1981) *The Design and Construction of Compilers*, Wiley, New York.

Hunter, R.B. (1985) *The Design and Construction of Compilers with Pascal*, Wiley, New York.

Johnson, S.C. (1975) Yacc - Yet Another Compiler Compiler, *Computing Science Technical Report 32*, AT&T Bell Laboratories, Murray Hill, NJ.

Kernighan, B.W. (1981) Why Pascal is not my favorite programming language, *Computer Science Technical Report 100*, AT&T Bell Laboratories, Murray Hill, NJ.

Kernighan, B.W. and Ritchie, D.M. (1988) *The C Programming Language* (2nd edn), Prentice-Hall, Englewood Cliffs, NJ.

King, K.N. (1996) *C Programming - a modern approach*, W.W. Norton, New York.

Lecarme, O. and Peyrolle-Thomas, M.C. (1973) Self compiling compilers: an appraisal of their implementations and portability, *Software - Practice and Experience*, 8(2), 149-170.

Lee, J.A.N. (1972) The formal definition of the BASIC language, *Computer Journal*, **15**, 37-41.

Lee, J.A.N. and Sammet, J.E. (1978) History of Programming Languages Conference HOPL I, *ACM SIGPLAN Notices*, **13**(8).

Lee, J.A.N. and Sammet, J.E. (1993) History of Programming Languages Conference HOPL II, *ACM SIGPLAN Notices*, **28**(3).

Lesk, M.E. (1975) Lex - a lexical analyser generator, *Computing Science Technical Report* 39,

AT&T Bell Laboratories, Murray Hill, NJ.

Levine, J.R., Mason, T. and Brown D. (1992) *Lex and Yacc* (2nd edn), O'Reilly and Associates, Sebastapol, CA.

MacCabe, A.B. (1993) *Computer Systems: Architecture, Organization and Programming*, Irwin, Boston, MA.

Mak, R. (1991) *Writing Compilers and Interpreters: an Applied Approach*, John Wiley, New York.

McGettrick, A.D. (1980) *The Definition of Programming Languages*, Cambridge University Press, Cambridge, England.

Meek, B.M. (1990) The static semantics file, *ACM SIGPLAN Notices*, **25**(4), 33-42.

Mody, R.P. (1991) C in education and software engineering, *ACM SIGCSE Bulletin*, **23**(3), 45-56.

Mössenböck, H. (1986) Alex: a simple and efficient scanner generator, *ACM SIGPLAN Notices*, **21**(12), 139- 148.

Mössenböck, H. (1990a) *Coco/R: A generator for fast compiler front ends*, Report 127, Departement Informatik, Eidgenössische Technische Hochschule, Zürich.

Mössenböck, H. (1990b) *A generator for production quality compilers*, in Proceedings of the Third International Workshop on Compiler-Compilers, Lecture Notes in Computer Science 471, Springer, Berlin.

Naur, P. (1960) Report on the algorithmic language Algol 60, *Communications of the ACM*, **3**, 299-314.

Naur, P. (1963) Revised report on the algorithmic language Algol 60, *Communications of the ACM*, **6**(1), 1- 17.

Nori, K.V., Ammann, U., Jensen, K. *et al.* (1981) Pascal-P implementation notes, in *Pascal - the Language and its Implementation*, Barron, D.W. (ed) Wiley, Chichester, England.

Panti, M. and Valenti, S. (1992) A modulus oriented hash function for the construction of minimal perfect tables, *ACM SIGPLAN Notices*, **27**(11), 33-38.

Parr, T.J., Dietz, H.G. and Cohen W.E. (1992) PCCTS 1.00: The Purdue Compiler Construction Tool Set, *ACM SIGPLAN Notices*, **27**(2), 88-165.

Parr, T.J. and Quong, R.W. (1995) ANTLR: A predicated-LL(k) parser generator, *Software - Practice and Experience*, **25**(7), 789-810.

Parr, T.J. and Quong, R.W. (1996) LL and LR Translators need k > 1 lookahead, *ACM SIGPLAN Notices*, **31**(2), 27-34.

Parr, T.J. (1996) *Language translation using PCCTS and C++ (a Reference Guide)*, Automata Publishing, San Jose, CA.

Pemberton, S. (1980) Comments on an error-recovery scheme by Hartmann, *Software - Practice and Experience*, **10**(3), 231-240.

Pemberton, S. and Daniels, M. (1982) *Pascal Implementation - the P4 Compiler*, Ellis Horwood, Chichester.

Pittman, T. and Peters, J. (1992) *The Art of Compiler Design*, Prentice-Hall, Englewood Cliffs, NJ.

Rechenberg, P. and Mössenböck, H. (1989) *A Compiler Generator for Microcomputers*, Prentice-Hall, Hemel Hempstead, England.

Rees, M. and Robson, D. (1987) *Practical Compiling with Pascal-S*, Addison-Wesley, Wokingham, England.

Sakkinen, M. (1992) The darker side of C++ revisited, *Structured Programming*, **13**(4), 155-178.

Sale, A.H.J. (1979) A note on scope, one-pass compilers, and Pascal, *Australian Computer Science Communications*, **1**(1), 80-82. Reprinted in *Pascal News*, **15**, 62-63.

Sale, A.H.J. (1981) The implementation of case statements in Pascal, *Software - Practice and Experience*, **11**(9), 929-942.

Schreiner, A.T. and Friedman, H.G. (1985) *Introduction to Compiler Construction with UNIX*, Prentice-Hall, Englewood Cliffs, NJ.

Sebesta, R.W. and Taylor, M.A. (1985) Minimal perfect hash functions for reserved word lists, *ACM SIGPLAN Notices*, **20**(12), 47-53.

Stirling, C. (1985) Follow set error recovery, *Software - Practice and Experience*, **15**(8), 239-257.

Stroustrup, B. (1990) *The C++ Programming Language* (2nd edn), Addison-Wesley, Reading, MA.

Stroustrup, B. (1993) *The Design and Evolution of C++*, Addison-Wesley, Reading, MA.

Terry, P.D. (1986) *Programming Language Translation*, Addison-Wesley, Wokingham, England.

Terry, P.D. (1995) Umbriel: another minimal programming language, *ACM SIGPLAN Notices*, **30**(5), 11- 17.

Topor, R.W. (1982) A note on error recovery in recursive descent parsers, *ACM SIGPLAN Notices*, **17**(2), 37- 40.

Tremblay, J.P. and Sorenson, P.G. (1985) *Theory and Practice of Compiler Writing*, McGraw-Hill, New York.

Trono, J.A. (1995) A comparison of three strategies for computing letter-oriented, minimal perfect hashing functions, *ACM SIGPLAN Notices*, **30**(4), 29-35.

Ullmann, J.R. (1994) *Compiling in Modula-2 - a First Introduction to Classical Recursive Descent Compiling*, Prentice-Hall, Hemel Hempstead, England.

van den Bosch, P.N. (1992) A bibliography on syntax error handling in context free languages, *ACM SIGPLAN Notices*, **27**(4), 77-86.

Waite, W.M. and Goos, G. (1984) *Compiler Construction*, Springer, New York.

Wakerly, J.F. (1981) *Microcomputer Architecture and Programming*, Wiley, New York.

Watson, D. (1989) *High-level Languages and their Compilers*, Addison-Wesley, Wokingham, England.

Watt, D.A. (1991) *Programming Language Syntax and Semantics*, Prentice-Hall, Hemel Hempstead, England.

Watt, D.A. (1993) *Programming Language Processors*, Prentice-Hall, Hemel Hempstead, England.

Welsh, J. and Hay, A. (1986) *A Model Implementation of Standard Pascal*, Prentice-Hall, Hemel Hempstead, England.

Welsh, J. and McKeag, M. (1980) *Structured System Programming*, Prentice-Hall, Hemel Hempstead, England.

Welsh, J. and Quinn, C. (1972) A Pascal compiler for ICL 1900 series computers, *Software - Practice and Experience*, **2**(1), 73-78.

Welsh, J., Sneeringer, W.J. and Hoare, C.A.R. (1977) Ambiguities and Insecurities in Pascal, *Software - Practice and Experience*, **7**(6), 685-696. (Reprinted in Barron (1981))

Wilson, I.P. and Addyman, A.M. (1982) *A Practical Introduction to Pascal - with BS6192*, MacMillan, London.

Wirth, N. (1974) *On the design of programming languages*, Proc IFIP Congress 74, 386-393, North-Holland, Amsterdam.

Wirth, N. (1976a) *Programming languages - what to demand and how to assess them*, Proc. Symposium on Software Engineering, Queen's University, Belfast.

Wirth, N. (1976b) *Algorithms + Data Structures = Programs*, Prentice-Hall, Englewood Cliffs, NJ.

Wirth, N. (1977) What can we do about the unnecessary diversity of notation for syntactic definitions? *Communications of the ACM*, **20**(11), 822-823.

Wirth, N. (1981) Pascal-S: A subset and its implementation, in *Pascal - the Language and its Implementation*, Barron, D.W. (ed), Wiley, Chichester, England.

Wirth, N. (1985) *Programming in Modula-2* (3rd edn), Springer, Berlin.

Wirth, N. (1986) *Compilerbau*, Teubner, Stuttgart.

Wirth, N. (1988a) From Modula to Oberon, *Software - Practice and Experience*, **18**(7), 661-670.

Wirth, N. (1988b) The programming language Oberon, *Software - Practice and Experience*, **18**(7),

671-690.

Wirth, N. (1996) *Compiler Construction*, Addison-Wesley, Wokingham, England.