

1. R.M. Burstall, R.A. Leaver and J.E. Sussams. Evaluation of transport costs for alternative factory sites — a case study. *Operational Research Quarterly* 13:345–354 (1962).
2. R.M. Burstall and G.R. Kiss. Information processing language V for the KDF-9 computer. *Automatic Programming Information* 18 (1963).
3. R.M. Burstall. Heuristic and decision tree methods on computers: some operational research applications. Ph.D. thesis, University of Birmingham (1966).
4. R.M. Burstall. Computer design of electricity supply networks by a heuristic method. *Computer Journal* 9(3):263–274 (1966).
5. R.M. Burstall and J.V. Oldfield. A software stack system for the 340/347 display. *Decuscope* 5(3):1–3 (1966).
6. R.M. Burstall. A heuristic method for a job-scheduling problem. *Operational Research Quarterly* 17:291–304 (1966).
7. R.M. Burstall. Tree searching methods with an application to a network design problem. *Machine Intelligence 1* (eds. N.L. Collins and D. Michie). Edinburgh: Oliver and Boyd, 65–85 (1967).
8. R.M. Burstall. Semantics of assignment. *Machine Intelligence 2* (eds. E. Dale and D. Michie). Edinburgh: Oliver and Boyd, 3–20 (1968).
9. R.M. Burstall, J.S. Collins and R.J. Popplestone. *POP-2 Papers*. Edinburgh: Oliver and Boyd (1968).
10. R.M. Burstall and R.J. Popplestone. POP-2 reference manual. *Machine Intelligence 2* (eds. E. Dale and D. Michie). Edinburgh: Oliver and Boyd, 205–246 (1968). Also in [9].
11. D. Michie, A. Ortony and R.M. Burstall. *Computer programming for schools: first steps in Algol*. Edinburgh: Oliver and Boyd (1968).
12. R.M. Burstall and J.S. Collins. An introduction to the POP-2 programming language. In [9] (1968).
13. J.S. Collins, A.P. Ambler, R.M. Burstall, R.D. Dunn, D. Michie, D.J.S. Pullin and R.J. Popplestone. Multi-POP/4120: a cheap on-line system for numerical and non-numerical computing. *Computer Bulletin* 12(5):186–189 (1968).
14. R.M. Burstall. Writing search algorithms in functional form. *Machine Intelligence 3* (ed. D. Michie). Edinburgh University Press, 373–385 (1968).

15. R.M. Burstall. Proving properties of programs by structural induction. *Computer Journal* 12(1):41–48 (1969).
16. R.M. Burstall. A program for solving word sum puzzles. *Computer Journal* 12(1):48–51 (1969).
17. R.M. Burstall and P.J. Landin. Programs and their proofs: an algebraic approach. *Machine Intelligence 4* (eds. B. Meltzer and D. Michie). Edinburgh University Press, 17–43 (1969).
article
18. R.M. Burstall. Formal description of program structure and semantics in first order logic. *Machine Intelligence 5* (eds. B. Meltzer and D. Michie). Edinburgh University Press, 79–98 (1969).
19. R.M. Burstall. Machine intelligence research at Edinburgh University. *Proc. ACM Intl. Computing Symposium, Bonn*, 696–708 (1970).
20. R.M. Burstall, J.S. Collins and R.J. Popplestone. *Programming in POP-2*. Edinburgh University Press (1971). A revision of [9] incorporating much new material.
21. H.G. Barrow, A.P. Ambler and R.M. Burstall. Some techniques for recognising structures in pictures. *Proc. Intl. Conf. on Frontiers of Pattern Recognition*, Honolulu. Academic Press, 1–29 (1972).
22. R.M. Burstall. An algebraic description of programs with assertions, verification and simulation. *Proc. ACM Conf. on Proving Assertions About Programs, Las Cruces. SIGPLAN Notices* 7(1):7–14 (1972).
23. R.M. Burstall. Some techniques for proving correctness of programs which alter data structures. *Machine Intelligence 7* (eds. B. Meltzer and D. Michie). Edinburgh University Press, 23–50 (1972).
24. D. Michie, A.P. Ambler, H.G. Barrow, R.M. Burstall, R.J. Popplestone and K.J. Turner. Vision and manipulation as a programming problem. *Proc. 1st Conf. on Industrial Robot Technology*, Nottingham, 185–190 (1973).
25. A.P. Ambler, H.G. Barrow, C.M. Brown, R.M. Burstall and R.J. Popplestone. A versatile computer-controlled assembly system. *Proc. 3rd Intl. Joint Conf. on Artificial Intelligence*, Stanford, 298–307 (1973).
26. J. Darlington and R.M. Burstall. A system which automatically improves programs. *Proc. 3rd Intl. Joint Conf. on Artificial Intelligence*, Stanford, 479–485 (1973).
27. R.M. Burstall. Program proving as hand simulation with a little induction. Invited paper, *Proc. IFIP Congress '74*, Stockholm, 308–312 (1974).

28. R.M. Burstall and J.W. Thatcher. The algebraic theory of recursive program schemes. *Proc. 1st Intl. Symp. on Category Theory Applied to Computation and Control*, San Francisco, 1974. Springer Lecture Notes in Computer Science, Vol. 25, 126–131 (1975).
29. R.M. Burstall and J. Darlington. Some transformations for developing recursive programs. Invited paper, *Proc. Intl. Conf. on Reliable Software*, Los Angeles, 465–472 (1975).
30. A.P. Ambler, H.G. Barrow, C.M. Brown, R.M. Burstall and R.J. Poplestone. A versatile system for computer-controlled assembly. *Artificial Intelligence* 6(2):129–156 (1975). Based on [??].
31. J. Darlington and R.M. Burstall. A system which automatically improves programs. *Acta Informatica* 6:41–60 (1976). Revised version of [??].
32. H.G. Barrow and R.M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters* 4(4):83–84 (1976).
33. R.M. Burstall. Program proof, program transformation, program synthesis for recursive programs. Lecture notes for course on “Data and program structures: syntax and semantics”. International School on Theory and Application of Computers, Erice, Sicily (1976).
34. R.M. Burstall and J. Darlington. A transformation system for developing recursive programs. *Journal of the Association for Computing Machinery* 24(1):44–67 (1977). Based on [??].
35. R. Burstall and J. Goguen. Putting theories together to make specifications. Invited paper, *Proc. 5th Intl. Joint Conf. on Artificial Intelligence*, Cambridge, Massachusetts, 1045–1058 (1977).
36. R.M. Burstall. Design considerations for a functional programming language. Invited paper, *Proc. Infotech State of the Art Conf. “The Software Revolution”*, Copenhagen, 45–57 (1977).
37. R.M. Burstall and M. Feather. Program development by transformations: an overview. *Les fondements de la programmation. Proc. Toulouse CREST Course on Programming* (eds. M. Amirchahy and D. Neel). Le Chesnay: IRIA-SEFI (1978).
38. R.M. Burstall. Concepts versus code in program design. *Proc. Euro IFIP ’79*, London. North-Holland (1979).
39. R.M. Burstall and J.A. Goguen. The semantics of CLEAR, a specification language. *Proc. 1979 Winter School on Abstract Software Specifications*, Copenhagen. Springer Lecture Notes in Computer Science, Vol. 86, 292–332 (1980).

40. J.A. Goguen and R.M. Burstall. CAT, a system for the structured elaboration of correct programs from structured specifications. Technical Report CSL-118, SRI International, Menlo Park (1980).
41. J.L. Weiner and R.M. Burstall. Making programs more readable. *Proc. 4th Intl. Symp. on Programming*, Paris. Springer Lecture Notes in Computer Science, Vol. 83, 372–341 (1980).
42. R.M. Burstall. Electronic category theory. Invited paper, *Proc. 9th Intl. Symp. on Mathematical Foundations of Computer Science*, Rydzyna. Springer Lecture Notes in Computer Science, Vol. 88, 22–39 (1980).
43. R.M. Burstall, D.B. MacQueen and D.T. Sannella. HOPE: an experimental applicative language. *Proc. 1980 LISP Conference*, Stanford, 136–143 (1980).
44. R.M. Burstall and J.A. Goguen. An informal introduction to CLEAR, a specification language. *The Correctness Problem in Computer Science* (eds. R. Boyer and J. Moore). Academic Press 185–213 (1981). Republished in: *Software Specification Techniques* (eds. N. Gehani and A.D. McGettrick). Addison Wesley, 363–389, 1986.
45. R.M. Burstall and J.A. Goguen. Algebras, theories and freeness: an introduction for computer scientists. *Theoretical Foundations of Programming Methodology: Lecture Notes of an International Summer School*, Marktoberdorf, 1981 (eds. M. Broy and G. Schmidt). D. Reidel, 329–348 (1982).
46. A. Pettorossi and R.M. Burstall. Deriving very efficient algorithms for evaluating linear recurrence relations using the program transformation technique. *Acta Informatica* 18:181–206 (1982).
47. R.M. Burstall and N. Suzuki. Sakura: a VLSI modelling language. *Proc. Conf. on Advanced Research in VLSI*, Cambridge, Massachusetts (1982).
48. D.T. Sannella and R.M. Burstall. Structured theories in LCF. *Proc. 8th Colloq. on Trees in Algebra and Programming*, L’Aquila. Springer Lecture Notes in Computer Science, Vol. 159, 377–391 (1983).
49. J.A. Goguen and R.M. Burstall. Introducing institutions. *Proc. of Logics of Programming Workshop*, Pittsburgh. Springer Lecture Notes in Computer Science, Vol. 164, 221–256 (1983).
50. J.A. Goguen and R.M. Burstall. Some fundamental algebraic tools for the semantics of computation. Part 1: Comma categories, colimits, signatures and theories. *Theoretical Computer Science* 31(1–2):175–209 (1984). Part 2: Signed and abstract theories. *Theoretical Computer Science* 31(3):263–295 (1984).

51. R.M. Burstall and B. Lampson. A kernel language for abstract data types and modules. Invited paper, *Proc. Intl. Symp. on Semantics of Data Types*, Sophia-Antipolis. Springer Lecture Notes in Computer Science, Vol. 173, 1–50 (1984).
52. R.M. Burstall. Programming with modules as typed functional programming. Invited paper, *Proc. Intl. Conf. on 5th Generation Computer Systems*, Tokyo, 103–112 (1984).
53. D.E. Rydeheard and R.M. Burstall. Monads and theories: a survey for computation. Seminar on the Application of Algebra to Language Definition and Compilation, Fontainebleau, 1982. *Algebraic Methods in Semantics* (eds. M. Nivat and J.C. Reynolds). Cambridge University Press, 575–605 (1985).
54. R.M. Burstall. Inductively defined functions. Invited paper, *Proc. Intl. Joint Conf. on Theory and Practice of Software Development*, Berlin. Springer Lecture Notes in Computer Science, Vol. 185, 92–96 (1985).
55. J.A. Goguen and R.M. Burstall. A study in the foundations of programming methodology: specifications, institutions, charters and parchments. *Proc. Summer Workshop on Category Theory and Computer Programming*, Guildford, 1985. Springer Lecture Notes in Computer Science, Vol. 240, 313–333 (1986).
56. D.E. Rydeheard and R.M. Burstall. A categorical unification algorithm. *Proc. Summer Workshop on Category Theory and Computer Programming*, Guildford, 1985. Springer Lecture Notes in Computer Science, Vol. 240, 493–505 (1986).
57. R.M. Burstall and D.E. Rydeheard. Computing with categories. *Proc. Summer Workshop on Category Theory and Computer Programming*, Guildford, 1985. Springer Lecture Notes in Computer Science, Vol. 240, 506–519 (1986).
58. R.M. Burstall. Inductively defined functions in functional programming languages. *Journal of Computer and System Sciences* 34:409–421 (1987). Based on [??].
59. R.M. Burstall. Research in interactive theorem proving at Edinburgh University. Invited paper, *Proc. 20th IBM Computer Science Symposium*, Shizuoka, Japan (1987).
60. D.E. Rydeheard and R.M. Burstall. *Computational Category Theory*. Prentice-Hall (1988).
61. B. Lampson and R.M. Burstall. Pebble, a kernel language for modules and abstract data types. *Information and Computation* 76:278–346 (1988). Based on [??].

62. R.M. Burstall and F. Honsell. A natural deduction treatment of operational semantics. Invited paper, *Proc. 8th Conf. on Foundations of Software Technology and Theoretical Computer Science*, Pune. Springer Lecture Notes in Computer Science, Vol. 338, 250–269 (1988).
63. R.M. Burstall. Computer assisted proof for mathematics: an introduction, using the LEGO proof system. *Proc. IAM Conference on The Revolution in Mathematics Caused by Computing*, Brighton (1990).
64. A. Tarlecki, R.M. Burstall and J.A. Goguen. Some fundamental algebraic tools for the semantics of computation. Part 3: Indexed categories. *Theoretical Computer Science* 91:239–264 (1991).
65. R.M. Burstall. Computing: yet another reality construction. Invited paper, *Software Development and Reality Construction* (eds. C. Floyd, H. Züllighoven, R. Budde and R. Keil-Slawik). Springer Verlag (1992).
66. J.A. Goguen and R.M. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the Association for Computing Machinery* 39(1):95–146 (1992).
67. J. McKinna and R.M. Burstall. Deliverables: a categorical approach to program development in type theory. Invited paper, *Proc. 18th Intl. Symp. on Mathematical Foundations of Computer Science*, Gdansk. Springer Lecture Notes in Computer Science, Vol. 711, 32–67 (1993).
68. R.M. Burstall and R. Diaconescu. Hiding and behaviour: an institutional approach. *A Classical Mind: Essays in Honour of C.A.R. Hoare*. Prentice-Hall, 75–92 (1994).
69. R.M. Burstall. Terms, proofs and refinement. Invited paper, *Proc. 9th IEEE Symp. on Logic in Computer Science*, Paris, 2–7 (1994).
70. M. Sato, T. Sakurai and R.M. Burstall. Explicit environments. *Proc. 4th Intl. Conf. on Typed Lambda Calculi and Applications*, L’Aquila. Springer Lecture Notes in Computer Science, Vol. 1581, 340–354 (1999).
71. R.M. Burstall. ProveEasy: helping people learn to do proofs. Invited paper, *Computing: the Australasian Theory Symposium (CATS 2000)*, Canberra. *Electronic Notes in Theoretical Computer Science* 31 (2000).
72. R.M. Burstall. Christopher Strachey — understanding programming languages. *Higher-Order and Symbolic Computation* 13:51–55 (2000).
73. H. Goguen, R. Brooksby and R.M. Burstall. Memory management: an abstract formulation of incremental tracing. *Types for Proofs and Programs: Intl. Workshop TYPES’99 — Selected Papers*. Springer Lecture Notes in Computer Science, Vol. 1956, 148–161 (2000).