

A Bibliography of Combinators

Stephen Wolfram

Foundational Documents

M. Schönfinkel (1924), “Über die Bausteine der mathematischen Logik” (in German) [“On the Building Blocks of Mathematical Logic”], *Mathematische Annalen* 92, 305–316. doi: 10.1007/BF01448013. (Translated by S. Bauer-Mengelberg (1967), as “On the Building Blocks of Mathematical Logic”, in *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, J. van Heijenoort, Harvard University Press, 357–366.)

H. B. Curry (1929), “An Analysis of Logical Substitution”, *American Journal of Mathematics* 51, 363–384. doi: 10.2307/2370728.

H. B. Curry (1930), “Grundlagen der kombinatorischen Logik” (in German) [“Foundations of Combinatory Logic”], *American Journal of Mathematics* 52, 509–536. doi: 10.2307/2370619. (Translated by F. Kamareddine and J. Seldin (2016), as *Foundations of Combinatory Logic*, Logic PhDs vol. 1, College Publications.)

Books

H. B. Curry, R. Feys and W. Craig (1958), *Combinatory Logic*, North-Holland.

H. Barendregt (1971), *Some Extensional Term Models for Combinatory Logics and Lambda Calculi*, Amsterdam. Republished 2006.

H. B. Curry, J. R. Hindley and J. P. Seldin (1972), *Combinatory Logic, Volume II*, North Holland.

S. Stenlund (1972), *Combinators, λ -Terms and Proof Theory*, D. Reidel Publishing Co.

F. B. Fitch (1974), *Elements of Combinatory Logic*, Yale University Press.

J. R. Hindley and J. P. Seldin (eds.) (1980), *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, Inc.

J. W. Klop (1980), *Combinatory Reduction Systems*, Mathematisch Centrum.

H. P. Barendregt (1981), *The Lambda Calculus: Its Syntax and Semantics*, North-Holland. Second edition 1984, republished 2006.

R. Smullyan (1985), *To Mock a Mockingbird*, Knopf.

J. R. Hindley and J. P. Seldin (1986), *Introduction to Combinators and Lambda-Calculus*, Cambridge University Press.

G. E. Révész (1988), *Lambda-Calculus, Combinators, and Functional Programming*, Cambridge University Press.

C. L. Hankin (1994), *Lambda Calculi: A Guide for Computer Scientists*, Clarendon Press.

E. Engeler (1995), *The Combinatory Programme*, Springer.

V. E. Wolfengangen (2003), *Combinatory Logic in Programming: Computations with Objects through Examples and Exercises*, Center JurInfoR Ltd. archive.org/details/CLP-2003_780.

J. R. Hindley and J. P. Seldin (2008), *Lambda-Calculus and Combinators: An Introduction*, Cambridge University Press. doi: [10.1017/CBO9780511809835](https://doi.org/10.1017/CBO9780511809835).

K. Bimbó (2011), *Combinatory Logic: Pure, Applied and Typed*, Chapman and Hall/CRC. doi: [10.1201/b11046](https://doi.org/10.1201/b11046).

H. Barendregt (2020), *Some Extensional Term Models for Combinatory Logics and Lambda Calculi 1971 (Series on Term Rewriting and Logic): A 2020 Republication: Motivation, the Making of & Hindsight*, independently published.

S. Wolfram (2021), *Combinators: A Centennial View*, Wolfram Media. wolfram-media.com/products/combinators-a-centennial-view.html.

Surveys & Summaries

P. C. Rosenbloom (1950), “III.4: Combinatory Logics”, in *The Elements of Mathematical Logic*, Dover, 109–133.

H. B. Curry (1955), “La théorie des combinateurs” (in French), *Rendiconti di Matematica e delle sue Applicazioni* 10, 347–359.

H. B. Curry (1957), “Combinatory Logic”, in *Summaries of Talks Presented at the Summer Institute of Symbolic Logic at Cornell University, July 1957*, notes by J. B. Rosser, Communications Research Division, Institute for Defense Analyses, 90–99.

H. B. Curry (1967), “Logic, Combinatory”, *The Encyclopedia of Philosophy*, Volume 4, R. Edwards (ed.), Macmillan and Free Press, 504–509.

W. V. Quine (1967), “Introduction to ‘Bausteine der mathematischen Logik’”, in *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, J. van Heijenoort, Harvard University Press, 355–357.

M. W. Bunder (1972), “An Introduction to, and Trends in Combinatorial Logic”, in *Proceedings of the First Australian Conference on Combinatorial Mathematics*, J. Wallis and W. D. Wallis (eds.), Australian Conference on Combinatorial Mathematics, TUNRA, 183–191.

- G. W. Petznick (1974), “Introduction to Combinatory Logic” in *Theory of Computation*, W. S. Brainerd and L. H. Landweber, Wiley & Sons, 274–323.
- H. B. Curry (1980), “Some Philosophical Aspects of Combinatory Logic”, *Studies in Logic and the Foundations of Mathematics* 101, 85–101. doi: [10.1016/S0049-237X\(08\)71254-0](https://doi.org/10.1016/S0049-237X(08)71254-0).
- A. Rezus (1982), *A Bibliography of Lambda-Calculi, Combinatory Logic and Related Topics*, Mathematisch Centrum. researchgate.net/publication/239844967_A_bibliography_of_lambda-calculi_combinatorial_logics_and_related_topics.
- J. R. Hindley (1986), “Combinators and Lambda-Calculus, a Short Outline”, in *Combinators and Functional Programming Languages*, G. Cousineau, P.-L. Curien and B. Robinet (eds.), LITP Spring School on Theoretical Computer Science 1985, *Lecture Notes in Computer Science* vol. 242, Springer, 104–122. doi: [10.1007/3-540-17184-3_42](https://doi.org/10.1007/3-540-17184-3_42).
- R. D. Knott (1988), “Introduction to Combinators and Lambda-Calculus”, *University Computing* 10, 166–167.
- P. Freyd (1989), “Combinators”, *Categories in Computer Science and Logic*, J. W. Gary and A. Scedrov (eds.), AMS, 63–66. doi: [10.1090/conm/092/1003195](https://doi.org/10.1090/conm/092/1003195).
- J. W. Klop, V. van Oostrom and F. van Raamsdonk (1993), “Combinatory Reduction Systems: Introduction and Survey”, *Theoretical Computer Science* 121, 279–308. doi: [10.1016/0304-3975\(93\)90091-7](https://doi.org/10.1016/0304-3975(93)90091-7).
- F. Cardone and J. R. Hindley (2006), “History of Lambda-Calculus and Combinatory Logic”, Swansea University Mathematics Department Research Report No. MRRS-05-06. cs.vu.nl/~femke/courses/ep/slides/history.pdf.
- K. Bimbó (2009), “Combinatory Logic”, *Stanford Encyclopedia of Philosophy*. plato.stanford.edu/entries/logic-combinatory.
- F. Cardone and J. R. Hindley (2009), “Lambda-Calculus and Combinators in the 20th Century”, in *Logic from Russell to Church*, D. M. Gabbay and J. Woods (eds.), Elsevier, 723–817. doi: [10.1016/S1874-5857\(09\)70018-4](https://doi.org/10.1016/S1874-5857(09)70018-4).
- S. Wolfram (2020, accessed May 19, 2021), “Combinators A Centennial View”. writings.stephenwolfram.com/2020/12/combinators-a-centennial-view.
- S. Wolfram (2020, accessed May 19, 2021), “Combinators and the Story of Computation”. writings.stephenwolfram.com/2020/12/combinators-and-the-story-of-computation.
- S. Wolfram (2020, accessed May 19, 2021), “Where Did Combinators Come From? Hunting the Story of Moses Schönfinkel”. writings.stephenwolfram.com/2020/12/where-did-combinators-come-from-hunting-the-story-of-moses-schonfinkel.

Combinators as Symbolic Expressions

Specific Combinators & Behavior

A. M. Turing (1937), “The β -Function in λ -K-Conversion”, *The Journal of Symbolic Logic* 2, 164. doi: 10.2307/2268281.

F. B. Fitch (1969), “Combinatory Logic and Negative Numbers”, in *The Logical Way of Doing Things, in Honor of Henry S. Leonard*, K. Lambert (ed.), Yale University Press, 265–277.

H. Barendregt and J. B. Rosser (1971), letter correspondence. wolframcloud.com/obj/combinators/Bibliography/Barendregt-Rosser-LetterCorrespondence.

A. Mitschke and G. Mitschke (1975), “A Solution of Böhm’s S-Problem”, Technische Hochschule Darmstadt, unpublished manuscript. wolframcloud.com/obj/combinators/Bibliography/Mitschke-Mitschke-ASolutionofBohmsSProblem.

A. S. Kuzičev (1976), “Combinatorially Complete Systems with Operators Ξ , F, Q, Π , \exists , P, \neg , $\&$, \vee , \equiv , \cdot ”, *Moscow University Mathematics Bulletin* 31, 117–122. istina.msu.ru/publications/article/18961167.

S. Zachos (1978), “Kombinatorische Logik und S-Terme” (in German), dissertation, ETH Zürich. doi: 10.3929/ethz-a-000136021.

J. Shultis (1985), “A Note on the S Combinator”, CU-CS-304-85. [core.ac.uk/download-pdf/54846325.pdf](http://core.ac.uk/download/pdf/54846325.pdf).

W. McCune and L. Wos (1987), “A Case Study in Automated Theorem Proving: Finding Sages in Combinatory Logic”. *Journal of Automated Reasoning* 3, 91–108. doi: 10.1007/BF00381147.

R. Legrand (1988), “A Basis Result in Combinatory Logic”, *The Journal of Symbolic Logic* 53, 1224–1226. doi: 10.2307/2274616.

L. Wos and W. McCune (1988), “Searching for Fixed Point Combinators by Using Automated Theorem Proving: A Preliminary Report”, Report ANL-88-10, Argonne National Laboratory, Argonne, IL. doi: 10.2172/6852789.

J. D. Fokker (1989), *The Systematic Construction of a One-Combinator Basis*, University of Utrecht. cs.uu.nl/research/techreps/repo/CS-1989/1989-14.pdf.

R. Statman (1989), “The Word Problem for Smullyan’s Lark Combinator Is Decidable”, *Journal of Symbolic Computation* 7, 103–112. doi: 10.1016/S0747-7171(89)80044-6.

R. Statman (1989), “On Sets of Solutions to Combinator Equations”, *Theoretical Computer Science* 66, 99–104. doi: 10.1016/0304-3975(89)90148-5.

N. Dershowitz and R. Treinen (1991, accessed April 9, 2021), “The RTA List of Open Problems”. win.tue.nl/rtaloop/problems/summary.html.

- W. McCune and L. Wos (1991), “The Absence and the Presence of Fixed Point Combinators”, *Theoretical Computer Science* 87, 221–228. doi: [10.1016/0304-3975\(91\)90034-Y](https://doi.org/10.1016/0304-3975(91)90034-Y).
- R. Statman (1991), “There Is No Hyperrecurrent S, K Combinator”, Carnegie Mellon University. Journal contribution. doi: [10.1184/R1/6480071.v1](https://doi.org/10.1184/R1/6480071.v1).
- J. Fokker (1992), “The Systematic Construction of a One-Combinator Basis for Lambda-Terms”, *Formal Aspects of Computing* 4, 776–780. doi: [10.1007/BF03180572](https://doi.org/10.1007/BF03180572).
- M. Sprenger and M. Wymann-Böni (1993), “How to Decide the Lark”, *Theoretical Computer Science* 110, 419–432. doi: [10.1016/0304-3975\(93\)90015-L](https://doi.org/10.1016/0304-3975(93)90015-L).
- R. Statman (1993), “Some Examples of Non-existent Combinators”, *Theoretical Computer Science* 121, 441–448. doi: [10.1016/0304-3975\(93\)90096-C](https://doi.org/10.1016/0304-3975(93)90096-C).
- L. Wos (1993), “The Kernel Strategy and Its Use for the Study of Combinatory Logic”, *Journal of Automated Reasoning* 10, 287–343. doi: [10.1007/BF00881795](https://doi.org/10.1007/BF00881795).
- P. Trigg, J. R. Hindley and M. W. Bunder (1994), “Combinatory Abstraction Using \mathbf{B} , \mathbf{B}' and Friends”, *Theoretical Computer Science* 135, 405–422. doi: [10.1016/0304-3975\(94\)90114-7](https://doi.org/10.1016/0304-3975(94)90114-7).
- B. Intrigila (1997), “Non-existent Statman’s Double Fixed Point Combinator Does Not Exist, Indeed”, *Information and Computation* 137, 35–40. doi: [10.1006/inco.1997.2633](https://doi.org/10.1006/inco.1997.2633).
- R. Statman (2000), “On the Word Problem for Combinators”, in *Rewriting Techniques and Applications*, L. Bachmair (ed.), International Conference on Rewriting Techniques and Applications 2000, *Lecture Notes in Computer Science* vol. 1833, Springer, 203–213. doi: [10.1007/10721975_14](https://doi.org/10.1007/10721975_14).
- J. Waldmann (2000), “The Combinator \mathbf{S} ”, *Information and Computation* 159, 2–21. doi: [10.1006/inco.2000.2874](https://doi.org/10.1006/inco.2000.2874).
- D. Probst and T. Studer (2001), “How to Normalize the Jay”, *Theoretical Computer Science* 254, 677–681. doi: [10.1016/S0304-3975\(00\)00379-0](https://doi.org/10.1016/S0304-3975(00)00379-0).
- T. Dörges (2002), “Unendliche Reduktionen in der kombinatorischen Logik” (in German). Diplomarbeit, Universität Leipzig. nbn-resolving.org/urn:nbn:de:bsz:15-qucosa2-165217.
- B. J. Maclennan (2002), “Molecular Combinator Reference Manual”, Department of Computer Science, University of Tennessee, Knoxville, UPIM Report 2, Technical Report UT-CS-02-489. [researchgate.net/publication/2854487_Molecular_Combinator_Reference_Manual](https://www.researchgate.net/publication/2854487_Molecular_Combinator_Reference_Manual).
- S. Wolfram (2002), “Symbolic Systems”, etc. in *A New Kind of Science*, Wolfram Media, 102–104. [wolframscience.com/nks/p102--symbolic-systems](https://www.wolframscience.com/nks/p102--symbolic-systems).
- R. Statman (2005), “Two Variables Are Not Enough”, in *Theoretical Computer Science*, M. Coppo, E. Lodi and G. M. Pinna (eds.), Italian Conference on Theoretical Computer Science 2005, *Lecture Notes in Computer Science* vol. 3701, Springer, 406–409. doi: [10.1007/11560586_32](https://doi.org/10.1007/11560586_32).

- J. W. Klop (2007), “New Fixed Point Combinators from Old”, in “Reflections on Type Theory, λ -Calculus, and the Mind. Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday”, E. Barendsen, et al. (eds.), University Nijmegen, 197–210. cs.ru.nl/barendregt60/essays/klop/art16_klop.pdf.
- J. Endrullis, R. de Vrijer and J. Waldmann (2010), “Local Termination: Theory and Practice”, *Logical Methods in Computer Science* 6, 1–37. doi: 10.2168/LMCS-6(3:20)2010.
- P. Cheilaris, J. Ramirez and S. Zachos (2011), “Checking in Linear Time if an S-Term Normalizes”, in *Proceedings of the 8th Panhellenic Logic Symposium*, Panhellenic Logic Symposium 8. 147.102.36.12/~philaris/publications/PLS8_submission_23.pdf.
- H. P. Barendregt, et al. (2017), “Dance of the Starlings”, in *Raymond Smullyan on Self Reference*, M. Fitting and B. Rayman (eds.), *Outstanding Contributions to Logic* vol. 14, Springer, 67–111. doi: 10.1007/978-3-319-68732-2_5.
- M. Ikebuchi and K. Nakano (2018), “On Repetitive Right Application of B-Terms”. [arXiv:1703.10938](https://arxiv.org/abs/1703.10938).
- E. Paul (2020), “Analyzing and Computing with Combinators”, Wolfram Summer Camp 2020. community.wolfram.com/groups/-/m/t/2034214.

Conversions & Notations

- H. B. Curry (1933), “Apparent Variables from the Standpoint of Combinatory Logic”, *Annals of Mathematics Second Series* 34, 381–404. [jstor.org/stable/1968167](https://www.jstor.org/stable/1968167).
- W. V. Quine (1936), “A Reinterpretation of Schönfinkel’s Logical Operators”, *Bulletin of the American Mathematical Society* 42, 87–89. Zbl: 0014.00301.
- R. Feys (1946), “La technique de la logique combinatoire” (in French), *Revue philosophique de Louvain* 44, 74–103. [jstor.org/stable/26332881](https://www.jstor.org/stable/26332881).
- R. Feys (1953), “Peano et Burali-Forti précurseurs de la logique combinatoire” (in French), *Actes du XIème Congrès International de Philosophie* 5, 70–72. doi: 10.5840/wcp1119535131.
- F. B. Fitch (1957), “Combinatory Logic and Whitehead’s Theory of Prehensions”, *Philosophy of Science* 24, 331–335. [jstor.org/stable/185007](https://www.jstor.org/stable/185007).
- H. B. Curry (1964), “The Elimination of Variables by Regular Combinators 1”, in *The Critical Approach to Science and Philosophy: Essays in Honor of Karl Popper*, M. Bunge (ed.), The Free Press, 127–143. Republished 1999 and 2017. doi: 10.4324/9781351313087-9.
- D. Meredith (1974), “Combinatory and Propositional Logic”, *Notre Dame Journal of Formal Logic* 15, 156–160. doi: 10.1305/ndjfl/1093891208.
- M. A. Amer (1975), “Parentheses in Combinatory Logic”, *Studies in Logic and the Foundations of Mathematics* 80, 429–432. doi: 10.1016/S0049-237X(08)71961-X.

- D. Meredith (1975), “Combinator Operations”, *Studia Logica* 24, 367–385. doi: [10.1007/BF02121666](https://doi.org/10.1007/BF02121666).
- D. Scott (1975), “Some Philosophical Issues Concerning Theories of Combinators”, in *λ -Calculus and Computer Science Theory*, C. Böhm (ed.), International Symposium on Lambda-Calculus and Computer Science Theory 1975, *Lecture Notes in Computer Science* vol. 37, Springer, 346–366. doi: [10.1007/BFb0029537](https://doi.org/10.1007/BFb0029537).
- S. K. Abdali (1976), “An Abstraction Algorithm for Combinatory Logic”, *Journal of Symbolic Logic* 41, 222–224. [jstor.org/stable/2272961](https://www.jstor.org/stable/2272961).
- R. Canal (1978), “Complexité de la réduction en logique combinatoire” (in French), dissertation, Université Paul Sabatier.
- D. A. Turner (1979), “Another Algorithm for Bracket Abstraction”, *The Journal of Symbolic Logic* 44, 267–270. doi: [10.2307/2273733](https://doi.org/10.2307/2273733).
- B. Robinet (1982), “Sur des séquences itératives de combinateurs” (in French) [“On Iterative Sequences of Combinators”], *Comptes Rendus de l'Académie des Sciences de Paris* 295, 29–30.
- E. V. Krishnamurthy and B. P. Vickers (1987), “Compact Numeral Representation with Combinators”, *The Journal of Symbolic Logic* 52, 519–525. doi: [10.2307/2274398](https://doi.org/10.2307/2274398).
- A. Piperno (1989), “Abstraction Problems in Combinatory Logic: A Compositive Approach”, *Theoretical Computer Science* 66, 27–43. doi: [10.1016/0304-3975\(89\)90143-6](https://doi.org/10.1016/0304-3975(89)90143-6).
- S. Broda and L. Damas (1995), “A New Translation Algorithm from Lambda Calculus into Combinatory Logic”, in *Progress in Artificial Intelligence*, C. Pinto-Ferreira and N. J. Mamede (eds.), Portuguese Conference on Artificial Intelligence 1995, *Lecture Notes in Computer Science* vol. 990, Springer, 359–370. doi: [10.1007/3-540-60428-6_30](https://doi.org/10.1007/3-540-60428-6_30).
- D. C. Keenan (1996), “To Dissect a Mockingbird: A Graphical Notation for the Lambda Calculus with Animated Reduction”. dkeen.com/Lambda.
- S. Broda and L. Dama (1997), “Compact Bracket Abstraction in Combinatory Logic”, *The Journal of Symbolic Logic* 62, 729–740. doi: [10.2307/2275570](https://doi.org/10.2307/2275570).
- C. Okasaki (2003), “Flattening Combinators: Surviving without Parentheses”, *Journal of Functional Programming* 13, 815–822. doi: [10.1017/S0956796802004483](https://doi.org/10.1017/S0956796802004483).
- J. Tromp (2004), “Binary Lambda Calculus”, GitHub. [tromp.github.io/cl/Binary_lambda_calculus.html](https://github.com/jtomp/Binary_lambda_calculus.html).
- J. Tromp (2007), “Binary Lambda Calculus and Combinatory Logic”, *Randomness and Complexity, From Leibniz to Chaitin*, 237–260. doi: [10.1142/9789812770837_0014](https://doi.org/10.1142/9789812770837_0014).
- B. Jay and T. Given-Wilson (2011), “A Combinatory Account of Internal Structure”, *The Journal of Symbolic Logic* 76, 807–826. doi: [10.2178/jsl/1309952521](https://doi.org/10.2178/jsl/1309952521).

Combinator Reduction

H. B. Curry (1932), “Some Additions to the Theory of Combinators”, *American Journal of Mathematics* 54, 551–558. doi: [10.2307/2370900](https://doi.org/10.2307/2370900).

K. L. Loewen (1962), “A Study of Strong Reduction in Combinatory Logic”, dissertation, the Pennsylvania State University.

B. Lercher (1963), “Strong Reduction and Recursion in Combinatory Logic”, dissertation, the Pennsylvania State University.

B. Lercher (1967), “Strong Reduction and Normal Form in Combinatory Logic”, *The Journal of Symbolic Logic* 32, 213–223. doi: [10.2307/2271659](https://doi.org/10.2307/2271659).

K. L. Loewen (1968), “Modified Strong Reduction in Combinatory Logic”, *Notre Dame Journal of Formal Logic* 9, 265–270. doi: [10.1305/ndjfl/1093893461](https://doi.org/10.1305/ndjfl/1093893461).

K. L. Loewen (1968), “The Church Rosser Theorem for Strong Reduction in Combinatory Logic”, *Notre Dame Journal of Formal Logic* 9, 299–302. doi: [10.1305/ndjfl/1093893514](https://doi.org/10.1305/ndjfl/1093893514).

G. Mitschke (1972), “Ein algebraischer Beweis für das Church-Rosser-Theorem” (in German), *Archiv für mathematische Logik und Grundlagenforschung* 15, 146–157. doi: [10.1007/BF02008531](https://doi.org/10.1007/BF02008531).

C. Batini and A. Pettorossi (1975), “Some Properties of Subbases in Weak Combinatory Logic”, Istituto di Automatica, University of Rome report 75-04. wolframcloud.com/obj/combinators/Bibliography/Batini-Pettorossi-SomePropertiesofSubbasesinWeakCombinatoryLogic.

A. Pettorossi (1975), “Sulla terminazione in classi subrecursive di algoritmi” (in Italian) [“On Termination in Subrecursive Classes of Algorithms”], in *Proceedings AICA Congress*, International Association for Analog Computation Congress 1975, 62–67. wolframcloud.com/obj/combinators/Bibliography/Pettorossi-OnTerminationinSubrecursiveClassesofAlgorithms.

A. Pettorossi (1975), “Un teorema per la sottobase {B}” (in Italian) [“A Theorem for the Subbase {B}”], Convegno su: “Codici, Complessità di Calcolo e Linguaggi Formali” 1975, Laboratorio di Cibernetica del CNR, 62-67. wolframcloud.com/obj/combinators/Bibliography/Pettorossi-ATheoremfortheSubbaseB.

J. R. Hindley (1977), “Combinatory Reductions and Lambda Reductions Compared”, *Mathematical Logic Quarterly* 23, 169–180. doi: [10.1002/malq.19770230708](https://doi.org/10.1002/malq.19770230708).

R. J. Lipton and L. Snyder (1977), “On the Halting of Tree Replacement Systems”, Department of Computer Science, Yale University, Research Report #99. wolframcloud.com/obj/combinators/Bibliography/Lipton-Snyder-OntheHaltingofTreeReplacementSystems.

- G. Mitschke (1977), “Discriminators in Combinatory Logic and λ -Calculus” preprint 336, Darmstadt University. wolframcloud.com/obj/combinators/Bibliography/Mitschke-DiscriminatorsinCombinatoryLogicandLambdacalculus.
- A. Pettorossi (1977), “Combinators as Tree-Transducers”, *Les arbres en algèbre et en programmation*, G. Jacobs (ed.), Université de Lille.
- R. Canal (1978), “Complexité de la réduction en logique combinatoire” (in French), *RAIRO Informatique Théorique* 12, 185–199. doi: [10.1051/ita/1978120403391](https://doi.org/10.1051/ita/1978120403391).
- G. Jacopini and M. Venturini-Zilli (1978), “Equating for Recurrent Terms of Lambda Calculus and Combinatory Logic”, *Pubblicazioni dell’ Istituto per le Applicazioni del Calcolo*, CRN, Ser. III, 85. wolframcloud.com/obj/combinators/Bibliography/Jacopini-Venturini-Zilli-EquatingforRecurrentTermsofLambdaCalculusandCombinatoryLogic.
- M. V. Zilli (1978), “Head Recurrent Terms in Combinatory Logic: A Generalization of the Notion of Head Normal Form”, in *Automata, Languages and Programming*, G. Ausiello and C. Böhm (eds.), International Colloquium on Automata, Languages, and Programming 1978, *Lecture Notes in Computer Science* vol. 62, Springer, 477–493. doi: [10.1007/3-540-08860-1_36](https://doi.org/10.1007/3-540-08860-1_36).
- R. Canal and J. Vignolle (1979), “Calculs finis et infinis dans les termes combinatoires” (in French), in *Lambda Calcul et Sémantique Formelle des langages de programmation*, B. Robinet (ed.), Litp-Ensta 109–130. wolframcloud.com/obj/combinators/Bibliography/Canal-Vignolle-Calculsfinisetinfinisdanslesternescombinatoires.
- G. Mitschke (1979), “The Standardization Theorem for λ -Calculus”, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 25, 29–31. doi: [10.1002/malq.19790250104](https://doi.org/10.1002/malq.19790250104).
- C. Böhm (1980), “An Abstract Approach to (Hereditary) Finite Sequences of Combinators”, in *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, Academic Press, 231–242.
- J. W. Klop (1980), “Reduction Cycles in Combinatory Logic”, in *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. R. Hindley and J. P. Seldin (eds.), Academic Press, Inc., 193–214.
- G. Mitschke (1980), “Infinite Terms and Infinite Reductions”, in *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. R. Hindley and J. P. Seldin (eds.), Academic Press Inc., 243–257.
- A. Pettorossi (1981), “A Property Which Guarantees Termination in Weak Combinatory Logic and Subtree Replacement Systems”, *Notre Dame Journal of Formal Logic* 22, 289–300. doi: [10.1305/ndjfl/1093883514](https://doi.org/10.1305/ndjfl/1093883514).
- L. V. Shabunin (1983), “On the Interpretation of Combinators with Weak Reduction”, *The Journal of Symbolic Logic* 48, 558–563. doi: [10.2307/2273446](https://doi.org/10.2307/2273446).

- L. Wos and W. McCune (1988), “Challenge Problems Focusing on Equality and Combinatory Logic: Evaluating Automated Theorem-Proving Programs”, in *9th International Conference on Automated Deduction*, E. Lusk and R. Overbeek (eds.), Conference on Automated Deduction 1988, *Lecture Notes in Computer Science* vol. 310, Springer, 714–729. doi: [10.1007/BFb0012870](https://doi.org/10.1007/BFb0012870).
- W. M. Farmer, J. D. Ramsdell and R. J. Watro (1990), “A Correctness Proof for Combinator Reduction with Cycles”, *ACM Transactions on Programming Languages and Systems* 12, 123–134. doi: [10.1145/77606.77612](https://doi.org/10.1145/77606.77612).
- A. Berarducci (1994), “Infinite λ -Calculus and Non-sensible Models”, *Logic and Algebra*, Routledge, 339–377. people.dm.unipi.it/berardu/Art/1996Nonsensible/non-sensible.pdf.
- R. Statman (1997), “Effective Reduction and Conversion Strategies for Combinators”, in *Rewriting Techniques and Applications*, H. Coman (ed.), International Conference on Rewriting Techniques and Applications 1997, *Lecture Notes in Computer Science* vol. 1232, Springer, 299–307. doi: [10.1007/3-540-62950-5_79](https://doi.org/10.1007/3-540-62950-5_79).
- B. Intrigila and E. Biasone (2000), “On the Number of Fixed Points of a Combinator in Lambda Calculus”, *Mathematical Structures in Computer Science* 10, 595–615. doi: [10.1017/S0960129500003091](https://doi.org/10.1017/S0960129500003091).
- P. Minari (2009), “A Solution to Curry and Hindley’s Problem on Combinatory Strong Reduction”, *Archive for Mathematical Logic* 48, 159–184. doi: [10.1007/s00153-008-0109-z](https://doi.org/10.1007/s00153-008-0109-z).
- A. Charguéraud (2010), “The Optimal Fixed Point Combinator”, in *Interactive Theorem Proving*, M. Kaufmann and L. C. Paulson (eds.), International Conference on Interactive Theorem Proving 2010, *Lecture Notes in Computer Science* vol. 6172, Springer, 195–210. doi: [10.1007/978-3-642-14052-5_15](https://doi.org/10.1007/978-3-642-14052-5_15).
- J. Endrullis, D. Hendriks and J. W. Klop (2010), “Modular Construction of Fixed Point Combinators and Clocked Böhm Trees”, in *2010 25th Annual IEEE Symposium on Logic in Computer Science*, IEEE Symposium on Logic in Computer Science 2010, IEEE, 111–119. doi: [10.1109/LICS.2010.8](https://doi.org/10.1109/LICS.2010.8).
- P. Dybjer and D. Kuperberg (2012), “Formal Neighbourhoods, Combinatory Böhm Trees, and Untyped Normalization by Evaluation”, *Annals of Pure and Applied Logic* 163, 122–131. doi: [10.1016/j.apal.2011.06.021](https://doi.org/10.1016/j.apal.2011.06.021).
- B. F. Redmond (2016), “Bounded Combinatory Logic and Lower Complexity”, *Information and Computation* 248, 215–226. doi: [10.1016/j.ic.2015.12.013](https://doi.org/10.1016/j.ic.2015.12.013).
- J. Endrullis, et al. (2017), “Clocked Lambda Calculus”, *Mathematical Structures in Computer Science* 27, 782–806. doi: [10.1017/S0960129515000389](https://doi.org/10.1017/S0960129515000389).
- A. Polonsky (2018), “Fixed Point Combinators as Fixed Points of Higher-Order Fixed Point Generators”, *Logical Methods in Computer Science* 16. [arXiv:1810.02239](https://arxiv.org/abs/1810.02239).

A. Bhayat and G. Reger (2020), “A Knuth-Bendix-Like Ordering for Orienting Combinator Equations”, in *Automated Reasoning*, N. Peltier and V. Sofronie-Stokkermans (eds.), International Joint Conference on Automated Reasoning 2020, *Lecture Notes in Computer Science* vol. 12166, Springer, 259–277. doi: 10.1007/978-3-030-51074-9_15.

J. W. Klop (2020), forthcoming, “An Inkling of Infinitary Rewriting”.

Random Combinators

B. J. Maclennan (1997), “Preliminary Investigation of Random SKI-Combinator Trees”, Department of Computer Science, University of Tennessee, Knoxville, Technical Report CS-97-370. researchgate.net/publication/2736515_Preliminary_Investigation_of_Random_SKI-Combinator_Trees.

R. David, et al. (2013), “Asymptotically Almost All Lambda-Terms Are Strongly Normalizing”, *Logical Methods in Computer Science* 9, 1–30. doi: 10.2168/LMCS-9(1:2)2013.

M. Bendkowski, K. Grygiel and M. Zaionc (2015), “Asymptotic Properties of Combinatory Logic”, in *Theory and Applications of Models of Computation*, R. Jain, S. Jain and F. Stephan (eds.), International Conference on Theory and Applications of Models of Computation 2015, *Lecture Notes in Computer Science* vol. 9076, Springer, 62–72. doi: 10.1007/978-3-319-17142-5_7.

M. Bendkowski, K. Grygiel and M. Zaionc (2017), “On the Likelihood of Normalization in Combinatory Logic”, *Journal of Logic and Computation* 27, 2251–2269. doi: 10.1093/log-com/exx005.

E. J. Parfitt (2017), “Patterns in Combinator Evolution”, *Complex Systems* 26, 119–134. doi: 10.25088/ComplexSystems.26.2.119.

Combinators as Mathematical Constructs

Combinatory Logic

H. B. Curry (1931), “The Universal Quantifier in Combinatory Logic”, *Annals of Mathematics Second Series* 32, 154–180. [jstor.org/stable/1968422](https://www.jstor.org/stable/1968422).

H. B. Curry (1934), “Some Properties of Equality and Implication in Combinatory Logic”, *Annals of Mathematics Second Series* 35, 849–860. doi: 10.2307/1968498.

H. B. Curry (1934), “Functionality in Combinatory Logic”, *Proceedings of the National Academy of Sciences of the United States of America* 20, 584–590. doi: 10.1073/pnas.20.11.584.

- H. B. Curry (1935), “First Properties of Functionality in Combinatory Logic”, *Tohoku Mathematical Journal* 41, 371–401. jstage.jst.go.jp/article/tmj1911/41/0/41_0_371/_article/-char/en.
- J. B. Rosser (1935), “A Mathematical Logic without Variables”, *Annals of Mathematics* 36, 127–150. doi: 10.2307/1968669.
- H. B. Curry (1941), “A Revision of the Fundamental Rules of Combinatory Logic”, *The Journal of Symbolic Logic* 6, 41–53. doi: 10.2307/2266655.
- H. B. Curry (1941), “Consistency and Completeness of the Theory of Combinators”, *The Journal of Symbolic Logic* 6, 54–61. doi: 10.2307/2266656.
- H. B. Curry (1942), “The Combinatory Foundations of Mathematical Logic”, *Journal of Symbolic Logic* 7, 49–64. doi: 10.2307/2266302.
- J. B. Rosser (1942), “New Sets of Postulates for Combinatory Logics”, *The Journal of Symbolic Logic* 7, 18–27. doi: 10.2307/2267551.
- H. B. Curry (1948/1949), “A Simplification of the Theory of Combinators”, *Synthese* 7, 391–399. [jstor.org/stable/20114069](https://www.jstor.org/stable/20114069).
- F. B. Fitch (1960), “A System of Combinatory Logic”, Office of Naval Research, Group Psychology Branch, Contract No. SAR/Nonr-609(16), Technical Report.
- H. B. Curry (1968), “Recent Advances in Combinatory Logic”, *Bulletin de la Société Mathématique Belgique* 20, 288–298.
- H. B. Curry (1969), “Modified Basic Functionality in Combinatory Logic”, *Dialectica* 23, 83–92. doi: 10.1111/j.1746-8361.1969.tb01183.x.
- A. S. Kuzičev (1971), “Certain Properties of Schönfinkel–Curry Combinators”, *Kombinatorny i Anal* 1, 105–119.
- H. P. Barendregt (1973), “Combinatory Logic and the Axiom of Choice”, *Indagationes Mathematicae* 35, 203–221. doi: 10.1016/1385-7258(73)90005-X.
- L. V. Šabunin (1973), “Simple Combinatory Calculi”, *Vestnik Moskovskogo Universiteta . Serija I. Matematika, Mehanika* 28, 30–35.
- H. P. Barendregt (1974), “Combinatory Logic and the ω -Rule”, *Fundamental Mathematicae* 82, 199–215. doi: 10.4064/fm-82-3-199-215.
- C. Batini and A. Pettorossi (1975), “On Subrecursiveness in Weak Combinatory Logic” in *λ -Calculus and Computer Science Theory*, C. Böhm (ed.), International Symposium on Lambda-Calculus and Computer Science Theory 1975, *Lecture Notes in Computer Science* vol. 37, Springer, 297–311. doi: 10.1007/BFb0029533.

L. V. Šabunin (1975), “Combinatory Calculi. I.” (in Russian), *Vestnik Moskovskogo Universiteta. Serija I. Matematika, Mehanika* 30 (1), 12–17.

L. V. Šabunin (1975), “Combinatory Calculi. II.” (in Russian), *Vestnik Moskovskogo Universiteta. Serija I. Matematika, Mehanika* 30 (2), 10–14.

G. Longo (1976), “On the Problem of Deciding Equality in Partial Combinatory Algebras and in a Formal System”, *Studia Logica: An International Journal for Symbolic Logic* 35, 363–375. [jstor.org/stable/20014826](https://www.jstor.org/stable/20014826).

J. P. Seldin (1978), “Recent Advances in Curry’s Program”, *Rendiconti del Seminario Matematico* 35, 77–88.

M. W. Bunder and R. K. Meyer (1985), “A Result for Combinators, BCK Logics and BCK Algebras”, *Logique et Analyse* 28, 33–40. [jstor.org/stable/44084112](https://www.jstor.org/stable/44084112).

H. Barendregt, M. Bunder and W. Dekkers (1993), “Systems of Illative Combinatory Logic Complete for First-Order Propositional and Predicate Calculus”, *The Journal of Symbolic Logic* 58, 769–788. doi: 10.2307/2275096.

D. J. Dougherty (1993), “Higher-Order Unification via Combinators”, *Theoretical Computer Science* 114, 273–298. doi: 10.1016/0304-3975(93)90075-5.

I. Bethke, J. W. Klop and R. de Vrijer (1999), “Extending Partial Combinatory Algebras”, *Mathematical Structures in Computer Science* 9, 483–505. doi: 10.1017/S0960129599002832.

Wolfram Guide Page (accessed March 26, 2021), “Combinatory Logic”. reference.wolfram.com/language/guide/CombinatoryLogic.html.

Models of Combinatory Logic

G. D. Plotkin (1972), “A Set-Theoretical Definition of Application”, Technical Report MiP-R-95, University of Edinburgh. doi: 10.1.1.62.4516.

C. Böhm and M. Dezani-Ciancaglini (1974), “Combinatorial Problems, Combinator Equations and Normal Forms”, in *Automata, Languages and Programming*, J. Loeckx (ed.), International Colloquium on Automata, Languages, and Programming 1974, *Lecture Notes in Computer Science* vol. 14, Springer, 185–199. doi: 10.1007/978-3-662-21545-6_13.

G. Jacopini (1975), “A Condition for Identifying Two Elements of Whatever Model of Combinatory Logic”, in *λ -Calculus and Computer Science Theory*, C. Böhm (ed.), International Symposium on Lambda-Calculus and Computer Science Theory 1975, *Lecture Notes in Computer Science* vol. 37, Springer, 213–219. doi: 10.1007/BFb0029527.

D. Scott (1975), “Combinators and Classes” in *λ -Calculus and Computer Science Theory*, C. Böhm, International Symposium on Lambda-Calculus and Computer Science Theory 1975, *Lecture Notes in Computer Science* vol. 37, Springer, 1–26. doi: 10.1007/BFb0029517.

- D. M. R. Park (1976), “The Y-Combinator in Scott’s Lambda-Calculus Models”, University of Warwick, Department of Computer Science, Theory of Computation Report (unpublished). wrap.warwick.ac.uk/46310/1/WRAP_Park_cs-rr-013.pdf.
- M. W. Bunder (1979), “Scott’s Models and Illative Combinatory Logic”, *Notre Dame Journal of Formal Logic* 20, 609–612. doi: [10.1305/ndjfl/1093882667](https://doi.org/10.1305/ndjfl/1093882667).
- E. Engeler (1981), “Algebras and Combinators”, *Algebra Universalis* 13, 389–392. doi: [10.1007/BF02483849](https://doi.org/10.1007/BF02483849).
- A. Meyer (1982), “What Is a Model of the Lambda Calculus?”, *Information and Control* 52(1), 87–122. doi: [10.1016/S0019-9958\(82\)80087-9](https://doi.org/10.1016/S0019-9958(82)80087-9).
- R. K. Meyer, M. W. Bunder and L. Powers (1991), “Implementing the ‘Fool’s Model’ of Combinatory Logic”, *Journal of Automated Reasoning* 7, 597–630. doi: [10.1007/BF01880331](https://doi.org/10.1007/BF01880331).
- A. Asperti and A. Ciabatonni (1995), “On Completability of Partial Combinatory Algebras” in *Proceedings of the Italian Conference on Theoretical Computer Science*, A. de Santis (ed.), Italian Conference on Theoretical Computer Science, World Scientific, 162–175. logic.at/staff/aga-ta/ictcs.pdf.
- I. Bethke and J. W. Klop (1995), “Collapsing Partial Combinatory Algebras”, in *Higher-Order Algebra, Logic, and Term Rewriting*, G. Dowek, et al. (eds.), International Workshop on Higher-Order Algebra, Logic, and Term Rewriting 1995, *Lecture Notes in Computer Science* vol. 1074, Springer, 57–73. doi: [10.1007/3-540-61254-8_19](https://doi.org/10.1007/3-540-61254-8_19).
- J. M. Dunn and R. K. Meyer (1997), “Combinators and Structurally Free Logic”, *Logic Journal of the IGPL* 4, 505–537. doi: [10.1093/jigpal/5.4.505](https://doi.org/10.1093/jigpal/5.4.505).
- Y. Akama (2004), “Limiting Partial Combinatory Algebras”, *Theoretical Computer Science* 311, 199–220. doi: [10.1016/S0304-3975\(03\)00360-8](https://doi.org/10.1016/S0304-3975(03)00360-8).
- K. Nour (2005), “Classic Combinatory Logic”, *Computational Logic Applications*, Computational Logic and Applications 2005. [arXiv:0905.1100](https://arxiv.org/abs/0905.1100).
- P. Tarau (2015), “On a Uniform Representation of Combinators, Arithmetic, Lambda Terms and Types”, in *Proceedings of the 17th International Symposium in Principles and Practice of Declarative Programming*, International Symposium in Principles and Practice of Declarative Programming 2015, Association for Computing Machinery, 244–255. doi: [10.1145/2790449.2790526](https://doi.org/10.1145/2790449.2790526).

Relations to Lambda Calculus

- A. Church (1941), “The Calculi of Lambda-Conversion”, *Annals of Mathematics Studies* 6, Princeton University Press. doi: [10.1515/9781400881932](https://doi.org/10.1515/9781400881932).

H. Barendregt (1970), “A Universal Generator for the Lambda Calculus”, unpublished manuscript. wolframcloud.com/obj/combinators/Bibliography/Barendregt-AUniversalGeneratorForTheLambdaCalculus.

B. Robinet (1979), *Lambda Calcul et Sémantique Formelle des langages de programmation* (in French), Litp-Ensta.

G. Pottinger (1985), “Pleasant Beta-Conversion Using Combinators” (abstract), *The Journal of Symbolic Logic* 50, 1101–1102. doi: 10.2307/2274003.

H.-G. Oberhauser (1986), “On the Correspondence of Lambda Style Reduction and Combinator Style Reduction”, in *Graph Reduction*, J. H. Fasel and R. M. Keller (eds.), Workshop on Graph Reduction 1986, *Lecture Notes in Computer Science* vol. 279, Springer, 1–25. doi: 10.1007/3-540-18420-1_47.

G. Huet (1989), *Initiation au λ -calcul* (in French), PN. apps.dtic.mil/dtic/tr/fulltext/u2/a218095.pdf.

S. Gilezan (1993), “A Note on Typed Combinators and Typed Lambda Terms”, *Zbornik Radova Prirodno-Matematichkog Fakulteta Serija za Matematiku Review of Research Faculty of Science Mathematics Series* 23, 319–329. emis.de/journals/NSJOM/Papers/23_1/NSJOM_23_1_319_329.pdf.

M. W. Bunder (1996), “Lambda Terms Definable as Combinators”, *Theoretical Computer Science* 169, 3–21. doi: 10.1016/S0304-3975(96)00111-9.

J. P. Seldin (2011), “The Search for a Reduction in Combinatory Logic Equivalent to $\lambda\beta$ -Reduction”, *Theoretical Computer Science* 412, 4905–4918. doi: 10.1016/j.tcs.2011.02.002.

H. Barendregt, W. Dekkers and R. Statman (2013), *Lambda Calculus with Types*, Perspectives in Logic, Cambridge University Press.

O. Kiselyov (2018), “ λ to SKI, Semantically”, in *Functional and Logic Programming*, J. Gallagher and M. Sulzmann (eds.), International Symposium on Functional and Logic Programming 2018, *Lecture Notes in Computer Science* vol. 10818, Springer, 33–50. doi: 10.1007/978-3-319-90686-7_3.

Relations to Type Theory

L. E. Sanchis (1964), “Types in Combinatory Logic”, *Notre Dame Journal of Formal Logic* 5, 161–180. doi: 10.1305/ndjfl/1093957876.

R. Hindley (1969), “The Principal Type-Scheme of an Object in Combinatory Logic”, *Transactions of the American Mathematical Society* 146, 29–60. doi: 10.2307/1995158.

J. P. Seldin (1969), “General Models for Type Theory Based on Combinatory Logic” (abstract), *Journal of Symbolic Logic* 34, 544. doi:10.2307/2270969.

J. Staples (1973), “Combinator Realizability of Constructive Finite Type Analysis”, *Cambridge Summer School in Mathematical Logic*, A. R. D. Mathias and H. Rogers (eds.), *Lecture Notes in Mathematics* vol. 337, Springer, 253–273. doi: [10.1007/BFb0066777](https://doi.org/10.1007/BFb0066777).

A. Rezus (1986), “Semantics of Constructive Type Theory”, *Libertas Mathematica* 6, 1–82. system.lm-ns.org/index.php/lm/article/view/292.

K. Meinke (1991), “Equational Specification of Abstract Types and Combinators”, in *Computer Science Logic*, E. Börger, et al. (eds.), International Workshop on Computer Science Logic 1991, *Lecture Notes in Computer Science* vol. 626, Springer, 257–271. doi: [10.1007/BFb0023772](https://doi.org/10.1007/BFb0023772).

M. W. Bunder (1992), “Combinatory Logic and Lambda Calculus with Classical Types”, *Logique & Analyse* 137/138, 69–79. jstor.org/stable/44084341.

J. R. Hindley (1997), *Basic Simple Type Theory*, Cambridge University Press. doi: [10.1017/CBO9780511608865](https://doi.org/10.1017/CBO9780511608865).

S. Broda and L. Damas (2000), “On Principal Types of Combinators”, *Theoretical Computer Science* 247, 277–290. doi: [10.1016/S0304-3975\(99\)00086-9](https://doi.org/10.1016/S0304-3975(99)00086-9).

J. Rehof and P. Urzyczyn (2011), “Finite Combinatory Logic with Intersection Types” in *Typed Lambda Calculi and Applications*, L. Ong (ed.), International Conference on Typed Lambda Calculi and Applications 2011, *Lecture Notes in Computer Science* vol. 6690, Springer, 169–183. doi: [10.1007/978-3-642-21691-6_15](https://doi.org/10.1007/978-3-642-21691-6_15).

B. Döder, et al. (2012), “Bounded Combinatory Logic”, in *Computer Science Logic (CSL’12)—26th International Workshop/21st Annual Conference of the EACSL*, P. Cègielski and A. Durand (eds.), Computer Science Logic 2012, Leibniz-Zentrum fuer Informatik, 243–258. doi: [10.4230/LIPIcs.CSL.2012.243](https://doi.org/10.4230/LIPIcs.CSL.2012.243).

J. Bessai (2019), “A Type-Theoretic Framework for Software Component Synthesis”, Dortmund University Library. doi: [10.17877/DE290R-20320](https://doi.org/10.17877/DE290R-20320).

Relations to Recursive Functions

H. B. Curry (1964), “Combinatory Recursive Objects of All Finite Types”, *Bulletin of the American Mathematical Society* 70, 814–817. doi: [10.1090/S0002-9904-1964-11245-3](https://doi.org/10.1090/S0002-9904-1964-11245-3).

W. W. Tait (1967), “Intensional Interpretations of Functionals of Finite Type I”, *Journal of Symbolic Logic* 32, 198–212. doi: [10.2307/2271658](https://doi.org/10.2307/2271658).

G. Mitschke (1972), “Lambda-definierbare Funktionen auf Peanoalgebren” (in German), *Archiv für mathematische Logik und Grundlagenforschung* 15, 31–35. doi: [10.1007/BF02019773](https://doi.org/10.1007/BF02019773).

- J. R. Hindley and G. Mitschke (1977), “Some Remarks about the Connections between Combinatory Logic and Axiomatic Recursion Theory”, *Archiv für mathematische Logik und Grundlagenforschung* 18, 99–103. doi: 10.1007/BF02007262.
- D. G. Skordev (1980), “Combinatory Spaces and Recursiveness in Them”, Sofia.
- J. A. Zashev (1983), “Recursion Theory in Partially Ordered Combinatory Models”, dissertation, Sofia University.
- J. A. Zashev (1987), “Recursion Theory in B -Combinatory Algebras”, *Serdica* 12, 225–237. Zbl: 0654.03034.
- N. Georgieva (1993), “Spaces with Combinators”, *Archive for Mathematical Logic* 32, 321–339. doi: 10.1007/BF01409966.
- M. Goldberg (2005), “On the Recursive Enumerability of Fixed-Point Combinators”, *BRICS Report Series* 12. doi: 10.7146/brics.v12i1.21867.

Relations to Other Mathematical Structures

- H. B. Curry (1934), “Foundations of the Theory of Abstract Sets from the Standpoint of Combinatory Logic” (abstract), *Bulletin of the American Mathematical Society* 40, 654. ams.org/journals/bull/1934-40-09/S0002-9904-1934-05918-4/S0002-9904-1934-05918-4.pdf.
- A. Church (1937), “Combinatory Logic as a Semigroup” (abstract), *Bulletin of the American Mathematical Society* 43, 333. ams.org/journals/bull/1937-43-05/S0002-9904-1937-06526-8/S0002-9904-1937-06526-8.pdf
- E. J. Cogan (1955), “A Formalization of the Theory of Sets from the Point of View of Combinatory Logic”, dissertation, University of Pennsylvania, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 1, 198–240. doi: 10.1002/malq.19550010304.
- M. W. Bunder (1969), “Set Theory Based on Combinatory Logic”, dissertation, University of Amsterdam.
- T. F. Fox (1970), “Combinatory Logic and Cartesian Closed Categories” dissertation, McGill University. escholarship.mcgill.ca/concern/theses/6h440t871?locale=en.
- C. Böhm (1971), “The Combinator Semigroup Has Two Generators”, I. S. I., Turin, unpublished manuscript. wolframcloud.com/obj/combinators/Bibliography/Bohm-TheCombinatorSemigroupHasTwoGenerators.
- M. W. Bunder (1974), “Propositional and Predicate Calculuses Based on Combinatory Logic”, *Notre Dame Journal of Formal Logic* 15, 25–34. doi: 10.1305/ndjfl/1093891196.
- M. W. Bunder (1974), “Various Systems of Set Theory Based on Combinatory Logic”, *Notre Dame Journal of Formal Logic* 15, 192–206. doi: 10.1305/ndjfl/1093891298.

A. S. Kuzičev (1974) “Deductive Operators of Combinatory Logic”, *Moscow University Mathematics Bulletin* 29, 8–14. wolframcloud.com/obj/combinators/Bibliography/Kuzicev-DeductiveOperatorsofCombinatoryLogic.

A. S. Kuzičev (1974), “On the Expressive Potentialities of Deductive Systems of λ -Conversion and Combinatory Logic”, *Moscow University Mathematics Bulletin* 29, 58–64. wolframcloud.com/obj/combinators/Bibliography/Kuzicev-OnTheExpressivePotentialitiesofDeductiveSystemsofLambdaconversionandCombinatoryLogic.

J. Staples (1974), “Combinator Realizability of a Constructive Morse Set Theory”, *Journal of Symbolic Logic* 39, 226–234. doi: [10.2307/2272635](https://doi.org/10.2307/2272635).

H. B. Curry (1975), “Representation of Markov Algorithms by Combinators” in *The Logical Enterprise*, A. R. Anderson, et al. (eds.), Yale University Press, 109–119.

M. Bel (1977), “An Intuitionistic Combinatory Theory Not Closed under the Rule of Choice”, *Indagationes Mathematicae* 80, 69–72. doi: [10.1016/1385-7258\(77\)90032-4](https://doi.org/10.1016/1385-7258(77)90032-4).

M. W. Bunder (1979), “On the Equivalence of Systems of Rules and Systems of Axioms in Illative Combinatory Logic”, *Notre Dame Journal of Formal Logic* 20, 603–608. doi: [10.1305/nd-jfl/1093882666](https://doi.org/10.1305/nd-jfl/1093882666).

M. R. Holmes (1991), “Systems of Combinatory Logic Related to Quine’s ‘New Foundations’”, *Annals of Pure and Applied Logic* 53, 103–133. doi: [10.1016/0168-0072\(91\)90052-N](https://doi.org/10.1016/0168-0072(91)90052-N).

Combinator Computation

Combinator Evaluation

T. J. W. Clarke, et al. (1980), “SKIM—The S, K, I Reduction Machine”, in *LFP '80: Proceedings of the 1980 ACM Conference on LISP and Functional Programming*, Conference on LISP and Functional Programming 1980, Association for Computing Machinery, 128–135. doi: [10.1145/800087.802798](https://doi.org/10.1145/800087.802798).

N. D. Jones and S. S. Muchnick (1982), “A Fixed-Program Machine for Combinator Expression Evaluation”, in *LFP '82: Proceedings of the 1982 ACM Symposium on LISP and Functional Programming*, D. P. Friedman and D. M. R. Park (eds.), Conference on LISP and Functional Programming 1982, Association for Computing Machinery, 11–20. doi: [10.1145/800068.802130](https://doi.org/10.1145/800068.802130).

D. R. Brownbridge (1985), “Cyclic Reference Counting for Combinator Machines”, in *Functional Programming Languages and Computer Architecture*, J. P. Jouannaud (ed.), Conference on Functional Programming Languages and Computer Architecture 1985, *Lecture Notes in Computer Science* vol. 201, Springer 273–288. doi: [10.1007/3-540-15975-4_42](https://doi.org/10.1007/3-540-15975-4_42).

- C. L. Hankin, P. E. Osmon and M. J. Shute (1985), “Cobweb—A Combinator Reduction Architecture”, in *Functional Programming Languages and Computer Architecture*, J. P. Jouannaud (ed.), Conference on Functional Programming Languages and Computer Architecture 1985, *Lecture Notes in Computer Science* vol. 201, Springer, 99–112. doi: 10.1007/3-540-15975-4_32.
- K. Noshita and T. Hikita (1985), “The BC-Chain Method for Representing Combinators in Linear Space”, *New Generation Computing* 3, 131–144. doi: 10.1007/BF03037065.
- R. Milner (1985), “Parallel Combinator Reduction Machine”, in *The Analysis of Concurrent Systems*, B. T. Denvir, et al. (eds.), *Lecture Notes in Computer Science* vol. 207, Springer, 121–127. doi: 10.1007/3-540-16047-7_41.
- M. Takeichi (1985), “An Alternative Scheme for Evaluating Combinator Expressions”, *Journal of Information Processing* 7, 246–253. id.nii.ac.jp/1001/00059885.
- M. Scheevel (1986), “NORMA: A Graph Reduction Processor”, in *Proceedings of the 1986 ACM Conference on LISP and Functional Programming*, Conference on LISP and Functional Programming 1986, Association for Computing Machinery, 212–219. doi: 10.1145/319838.319864.
- M. Castan, M.-H. Durand and M. Lemaître (1987), “A Set of Combinators for Abstraction in Linear Space”, *Information Processing Letters* 24, 183–188. doi: 10.1016/0020-0190(87)90183-9.
- K. Noshita and X.-X. He (1987), “A Fast Algorithm for Translating Combinator Expressions with BC-Chains”, *New Generation Computing* 5, 249–257. doi: 10.1007/BF03037465.
- A. Piperno (1987), “A Compositive Abstraction Algorithm for Combinatory Logic”, in *TAPSOFT '87*, H. Ehrig, et al. (eds.), International Joint Conference on Theory and Practice of Software Development 1987, *Lecture Notes in Computer Science* vol. 250, Springer, 39–51. doi: 10.1007/BFb0014971.
- P. H. Hartel and A. H. Veen (1988), “Statistics on Graph Reduction of SASL Programs”, *Software: Practice and Experience* 18, 239–253. doi: 10.1002/spe.4380180305.
- A. Contessa, et al. (1989), “MaRS, a Combinator Graph Reduction Multiprocessor”, in *PARLE '89 Parallel Architectures and Languages Europe*, E. Odijk, M. Rem and J. C. Syre (eds.), International Conference on Parallel Architectures and Languages Europe 1989, *Lecture Notes in Computer Science* vol. 365, Springer, 176–192. doi: 10.1007/3540512845_39.
- P. J. Koopman and P. Lee (1989), “A Fresh Look at Combinator Graph Reduction (Or, Having a TIGRE by the Tail)”, in *Proceedings of the SIGPLAN '89 Conference on Programming Language Design and Implementation*, Special Interest Group on Programming Languages 1989, ACM Press, 110–119. doi: 10.1145/74818.74828.

P. Koopman, P. Lee and D. P. Siewiorek (1990), “Cache Performance of Combinator Graph Reduction” in *Proceedings of the International Conference on Computer Languages 1990*, International Conference on Computer Languages 1990, IEEE, 39-48. doi: [10.1109/IC-CL.1990.63759](https://doi.org/10.1109/IC-CL.1990.63759).

P. H. Hartel (1991), “Performance of Lazy Combinator Graph Reduction”, *Software: Practice and Experience* 21, 299–329. doi: [10.1002/spe.4380210306](https://doi.org/10.1002/spe.4380210306).

M. A. Musicante and R. D. Lins (1991), “GM-C: A Graph Multi-combinator Machine”, *Microprocessing and Microprogramming* 31, 81–84. doi: [10.1016/S0165-6074\(08\)80048-8](https://doi.org/10.1016/S0165-6074(08)80048-8).

M. Waite, B. Giddings and S. Lavington (1991), “Parallel Associative Combinator Evaluation”, in *Parle '91 Parallel Architectures and Languages Europe*, E. H. L. Aarts, J. Leeuwen and M. Rem (eds.), *Lecture Notes in Computer Science* vol. 505, Springer, 771–788. doi: [10.1007/978-3-662-25209-3_47](https://doi.org/10.1007/978-3-662-25209-3_47).

A. Dickinson and M. T. Pope (1992), “A Simple Machine (Based on the SK-Combinator Reduction Mechanism)”, in *Proceedings from TENCON'92 - Technology Enabling Tomorrow*, IEEE Region 10 Conference 1992, IEEE, 845–849. doi: [10.1109/TENCON.1992.271851](https://doi.org/10.1109/TENCON.1992.271851).

S. Thompson and R. Linus (1992), “The Categorical Multi-combinator Machine: CMCM”, *The Computer Journal* 35, 170–176. doi: [10.1093/comjnl/35.2.170](https://doi.org/10.1093/comjnl/35.2.170).

S. Kahrs (1993), “Compilation of Combinatory Reduction Systems”, in *Higher-Order Algebra, Logic, and Term Rewriting*, J. Heering, et al. (eds.), International Workshop on Higher-Order Algebra, Logic, and Term Rewriting 1993, *Lecture Notes in Computer Science* vol. 816, Springer, 169–188. doi: [10.1007/3-540-58233-9_9](https://doi.org/10.1007/3-540-58233-9_9).

S. D. Swierstra, P. R. A. Alcocer and J. Saraiva (1998), “Designing and Implementing Combinator Languages”, in *Advanced Functional Programming*, S. D. Swierstra, J. N. Oliveira, and P. R. Henriques (eds.), International School on Advanced Functional Programming 1998, *Lecture Notes in Computer Science* vol. 1608, Springer, 150–206. doi: [10.1007/10704973_4](https://doi.org/10.1007/10704973_4).

jjpt4 (2017), “A Relational SKI Combinator Calculus Interpreter”, GitHub. github.com/jjpt4/skio.

Wolfram Guide Page (accessed March 10, 2021), “Combinator Functions”. wolframcloud.com/obj/wolframphysics/Tools/combinators.

Compilation to Combinators

S. L. Peyton Jones (1982), “An Investigation of the Relative Efficiencies of Combinators and Lambda Expressions”, in *Proceedings of the 1982 ACM Symposium on LISP and Functional Programming*, Conference on LISP and Functional Programming 1982, Association for Computing Machinery, 150–158. doi: [10.1145/800068.802145](https://doi.org/10.1145/800068.802145).

- F. W. Burton (1983), “A Linear Space Translation of Functional Programs to Turner Combinators”, *Information Processing Letters* 14, 201–204. doi: 10.1016/0020-0190(82)90014-X.
- T. Hikita (1984), “On the Average Size of Turner’s Translation to Combinator Programs”, *Journal of Information Processing* 7, 164–169. Zbl: 0386.68009.
- S. Hirokawa (1985), “Complexity of the Combinator Reduction Machine”, *Theoretical Computer Science* 41, 289–303. doi: 10.1016/0304-3975(85)90076-3.
- M. S. Joy, V. J. Rayward-Smith and F. W. Burton (1985), “Efficient Combinator Code”, *Computer Languages* 10, 211–224. doi: 10.1016/0096-0551(85)90017-7.
- K. Noshita (1985), “Translation of Turner Combinators in $O(n \log n)$ Space”, *Information Processing Letters* 20, 71–74. doi: 10.1016/0020-0190(85)90066-3.
- R. S. Bird (1987), “A Formal Development of an Efficient Supercombinator Compiler”, *Science of Computer Programming* 8, 113–137. doi: 10.1016/0167-6423(87)90017-7.
- H. Nielson and F. Nielson (1990), “Functional Completeness of the Mixed λ -Calculus and Combinatory Logic”, *Theoretical Computer Science* 70, 99–126. doi: 10.1016/0304-3975(90)90155-B
- M. S. Joy and V. J. Rayward-Smith (1995), “NP-Completeness of a Combinator Optimization Problem”, *Notre Dame Journal of Formal Logic* 36, 319–335. doi: 10.1305/ndjfl/1040248462.
- O. Kiselyov (1999, accessed April 15, 2021), “Many Faces of the Fixed-Point Combinator”. okmij.org/ftp/Computation/fixed-point-combinators.html.
- A. Hoogewijs and P. Audenaert (2003), “Combinatory Logic, a Bridge to Verified Programs”, *Logic Colloquium*, Proceedings, Association of Symbolic Logic. hdl.handle.net/1854/LU-802976.
- B. Lynn (2019, accessed April 15, 2021), “An Award-Winning Compiler”. crypto.stanford.edu/~blynn/compiler/ioccc.html.
- B. Lynn (accessed April 15, 2021), “LC to CL, Semantically”. crypto.stanford.edu/~blynn/compiler/sem.html.

Combinators in Functional Programming

- C. Böhm (1972), “Combinatory Foundation of Functional Programming”, in *LFP ’82: Proceedings of the 1982 ACM Symposium on LISP and Functional Programming*, Conference on LISP and Functional Programming 1982, Association for Computing Machinery, 29–36. doi: 10.1145/800068.802132.
- S. K. Abdali (1974), “A Combinatory Logic Model of Programming Languages”, thesis, University of Wisconsin. geomete.com/abdali/papers/phdthesis.pdf.
- P. M. Maurer and A. E. Oldehoeft (1983), “The Use of Combinators in Translating a Purely Functional Language to Low-Level Data-Flow Graphs”, *Computer Languages* 8, 27–45. doi: 10.1016/0096-0551(83)90004-8.

- M. Wand (1983), “Loops in Combinator-Based Compilers”, *Information and Control* 57, 148–164. doi: [10.1016/S0019-9958\(83\)80041-2](https://doi.org/10.1016/S0019-9958(83)80041-2).
- P. Hudak and D. Kranz (1984), “A Combinator-Based Compiler for a Functional Language”, *Proceedings of the 11th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, Principles of Programming Languages, Association of Computing Machinery, 122–132. doi: [10.1145/800017.800523](https://doi.org/10.1145/800017.800523).
- P.-L. Curien (1985), “Categorical Combinatory Logic”, in *Automata, Languages and Programming*, W. Brauer (ed.), International Colloquium on Automata, Languages, and Programming 1985, *Lecture Notes in Computer Science* vol. 194, Springer, 130–139. doi: [10.1007/BFb0015738](https://doi.org/10.1007/BFb0015738).
- J. Gibert (1985), “The J-Machine: Functional Programming with Combinators”, in *EUROCAL '85*, B. F. Caviness (ed.), European Conference on Computer Algebra 1985, *Lecture Notes in Computer Science* vol. 204, Springer, 197–198. doi: [10.1007/3-540-15984-3_262](https://doi.org/10.1007/3-540-15984-3_262).
- P. Hudak and B. Goldberg (1985), “Serial Combinators: ‘Optimal’ Grains of Parallelism”, in *Functional Programming Languages and Computer Architecture*, J. P. Jouannaud (ed.), Conference on Functional Programming Languages and Computer Architecture 1985, *Lecture Notes in Computer Science* vol. 201, Springer, 382–399. doi: [10.1007/3-540-15975-4_49](https://doi.org/10.1007/3-540-15975-4_49).
- P. Bellot (1986), “Gaal: A Functional Programming System with Uncurryfied Combinators and Its Reduction Machine”, in *ESOP 86*, B. Robinet and R. Wilhelm (eds.), European Symposium on Programming 1986, *Lecture Notes in Computer Science* vol. 213, Springer, 82–98. doi: [10.1007/3-540-16442-1_6](https://doi.org/10.1007/3-540-16442-1_6).
- C. L. Hankin, G. L. Burn and S. L. Peyton Jones (1986), “A Safe Approach to Parallel Combinator Reduction (Extended Abstract)”, in *ESOP 86*, B. Robinet and R. Wilhelm (eds.), European Symposium on Programming 1986, *Lecture Notes in Computer Science* vol. 213, Springer, 99–110. doi: [10.1007/3-540-16442-1_7](https://doi.org/10.1007/3-540-16442-1_7).
- I. Toyn and C. Runciman (1986), “Adapting Combinator and SECD Machines to Display Snapshots of Functional Computations”, *New Generation Computing* 4, 339–363. doi: [10.1007/BF03037389](https://doi.org/10.1007/BF03037389).
- A. Cheese (1987), “Combinatory Code and a Parallel Packet-Based Computational Model”, *ACM SIGPLAN Notices* 22, 49–58. doi: [10.1145/24714.24720](https://doi.org/10.1145/24714.24720).
- J. Gibert (1987), “Functional Programming with Combinators”, *Journal of Symbolic Computation* 4, 269–293. doi: [10.1016/S0747-7171\(87\)80009-3](https://doi.org/10.1016/S0747-7171(87)80009-3).
- S. R. D. Meira (1987), “Strict Combinators”, *Information Processing Letters* 24, 255–258. doi: [10.1016/0020-0190\(87\)90144-X](https://doi.org/10.1016/0020-0190(87)90144-X).
- A. D. Robinson (1987), “The Illinois Functional Programming Interpreter”, in *Papers of the Symposium on Interpreters and Interpretive Techniques*, SIGPLAN Symposium, Association for Computing Machinery, 64–73. doi: [10.1145/29650.29657](https://doi.org/10.1145/29650.29657).

- C. Böhm (1988), “Functional Programming and Combinatory Algebras”, in *Mathematical Foundations of Computer Science 1988*, M. P. Chytil, L. Janiga and V. Koubek (eds.), International Symposium on Mathematical Foundations of Computer Science 1988, *Lecture Notes in Computer Science* vol. 324, Springer, 14–26. doi: [10.1007/BFb0017128](https://doi.org/10.1007/BFb0017128).
- S. M. Sarwar, S. J. Hahn and J. A. Davis (1988), “Implementing Functional Languages on a Combinator-Based Reduction Machine”, *ACM SIGPLAN Notices* 23, 65–70. doi: [10.1145/44326.44333](https://doi.org/10.1145/44326.44333).
- P. P. Chu and J. A. Davis (1990), “Exploitation Fine-Grain Parallelism in a Combinator-Based Functional System”, in *Third Symposium on the Frontiers of Massively Parallel Computation*, Third Symposium on the Frontiers of Massively Parallel Computation 1990, IEEE, 489–493. doi: [10.1109/FMPC.1990.89500](https://doi.org/10.1109/FMPC.1990.89500).
- D. Spencer (1990), “A Survey of Categorical Computation: Fixed Points, Partiality, Combinators, ... Control?”, Oregon Graduate Institute, Department of Computer Science and Engineering, Technical Report No. CS/E 90-017. doi: [10.6083/2514nk87q](https://doi.org/10.6083/2514nk87q).
- E. Cherlin (1991), “Pure Functions in APL and J”, in *Proceedings of the International Conference on APL '91*, International Conference on APL 1991, North-Holland, 88–93. doi: [10.1145/114054.114065](https://doi.org/10.1145/114054.114065).
- J. Gateley and B. F. Duba (1991), “Call-by-Value Combinatory Logic and the Lambda-Value Calculus”, in *Mathematical Foundations of Programming Semantics*, S. Brookes, et al. (eds.), International Conference on Mathematical Foundations of Programming Semantics 1991, *Lecture Notes in Computer Science* vol. 598, Springer, 41–53. doi: [10.1007/3-540-55511-0_2](https://doi.org/10.1007/3-540-55511-0_2).
- P.-L. Curien (1993), *Categorical Combinators, Sequential Algorithms, and Functional Programming*, Springer Science & Business Media. doi: [10.1007/978-1-4612-0317-9](https://doi.org/10.1007/978-1-4612-0317-9).
- S. M. Sarwar and J. A. Davis (1994), “New Families of Combinators for Efficient List Manipulation”, *Journal of Systems and Software* 27, 137–146. doi: [10.1016/0164-1212\(94\)90027-2](https://doi.org/10.1016/0164-1212(94)90027-2).
- N. Raja and R. K. Shyamasundar (1995), “Combinatory Formulations of Concurrent Languages”, in *Algorithms, Concurrency and Knowledge*, K. Kanchanasut and J. J. Lévy (eds.), Asian Computing Science Conference 1995, *Lecture Notes in Computer Science* vol. 1023, Springer, 156–170. doi: [10.1007/3-540-60688-2_42](https://doi.org/10.1007/3-540-60688-2_42).
- Y. Lafont (1997), “Interaction Combinators”, *Information and Computation* 137, 69–101. doi: [10.1006/inco.1997.2643](https://doi.org/10.1006/inco.1997.2643).
- B. Meyer (1998), “The Component Combinator for Enterprise Applications”, *Journal of Object-Oriented Programming* 10, 5–9. archive.eiffel.com/general/editorial/1998/combinator.html.
- M. Goldberg (2005), “A Variadic Extension of Curry’s Fixed-Point Combinator”, in *Higher-Order Symbolic Computation* 18, 371–388. doi: [10.1007/s10990-005-4881-8](https://doi.org/10.1007/s10990-005-4881-8).

- P. D. Mosses (2007), “VDM Semantics of Programming Languages: Combinators and Monads”, *Formal Methods and Hybrid Real-Time Systems*, C. B. Jones, Z. Liu and K. Woodcock (eds.), *Lecture Notes in Computer Science* vol. 4700, Springer, 483–503. doi: [10.1007/978-3-540-75221-9_23](https://doi.org/10.1007/978-3-540-75221-9_23).
- P. Weaver, et al. (2007), “Constructing Language Processors with Algebra Combinators”, in *Constructing Language Processors with Algebra Combinators*, Generative Programming and Component Engineering 2007, Association for Computing Machinery, 155–164. doi: [10.1145/1289971.1289997](https://doi.org/10.1145/1289971.1289997).
- N. A. Danielsson (2010), “Total Parser Combinators”, in *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, International Conference on Functional Programming, Association for Computing Machinery, 285–296. doi: [10.1145/1863543.1863585](https://doi.org/10.1145/1863543.1863585).
- M. Bolingbroke, S. Peyton Jones and D. Vytiniotis (2011), “Termination Combinators Forever”, in *Proceedings of the 4th ACM Symposium on Haskell*, Haskell 2011, Association for Computing Machinery, 23–34. doi: [10.1145/2034675.2034680](https://doi.org/10.1145/2034675.2034680).
- O. Danvy and I. Zerny (2011), “Three Syntactic Theories for Combinatory Graph Reduction”, in *Logic-Based Program Synthesis and Transformation*, M. Alpuente (ed.), International Symposium on Logic-Based Program Synthesis and Transformation 2010, *Lecture Notes in Computer Science* vol. 6564. Springer, 1–20. doi: [10.1007/978-3-642-20551-4_1](https://doi.org/10.1007/978-3-642-20551-4_1).
- D. Devriese and F. Piessens (2011), “Explicitly Recursive Grammar Combinators”, in *Practical Aspects of Declarative Languages*, J. Launchbury and R. Rocha (eds.), International Symposium on Practical Aspects of Declarative Languages 2011, *Lecture Notes in Computer Science* vol. 6539, Springer, 84–98. doi: [10.1007/978-3-642-18378-2_9](https://doi.org/10.1007/978-3-642-18378-2_9).
- T. Schrijvers, et al. (2012), “Search Combinators”, *Constraints* 18, 269–305. doi: [10.1007/s10601-012-9137-8](https://doi.org/10.1007/s10601-012-9137-8).
- P. Martins, J. P. Fernandes and J. Saraiva (2013), “A Combinator Language for Software Quality Reports”, *International Journal of Computer and Communication Engineering* 2, 377–382. doi: [10.7763/IJCCE.2013.V2.208](https://doi.org/10.7763/IJCCE.2013.V2.208).
- T. Ridge (2014), “Simple, Efficient, Sound and Complete Combinator Parsing for All Context-Free Grammars, Using an Oracle”, in *Software Language Engineering*, B. Combemale, et al. (eds.), International Conference on Software Language Engineering 2014, *Lecture Notes in Computer Science* vol. 8706, Springer, 261–281. doi: [10.1007/978-3-319-11245-9_15](https://doi.org/10.1007/978-3-319-11245-9_15).
- M. S. New, et al. (2017), “Fair Enumeration Combinators”, *Journal of Functional Programming* 27, e19. doi: [10.1017/S0956796817000107](https://doi.org/10.1017/S0956796817000107).
- M. Sičák and J. Kollár (2017), “Supercombinator Driven Grammar Reconstruction”, in *Proceedings of 2017 IEEE 14th International Scientific Conference on Informatics*, International Scientific Conference on Informatics, IEEE, 322–326. doi: [10.1109/INFORMATIC-S.2017.8327268](https://doi.org/10.1109/INFORMATIC-S.2017.8327268).

A. Farrugia (2018), “Combinatory Logic: From Philosophy and Mathematics to Computer Science”, Junior College Multi-disciplinary Conference: Research, Practice and Collaboration: Breaking Barriers: Annual Conference, 307–320. um.edu.mt/library/oar/handle/123456789/38118.

Metaprogramming with Combinators

L. Fegaras, T. Sheard and D. Stemple (1992), “Uniform Traversal Combinators: Definition, Use and Properties”, in *Automated Deduction—CADE-11*, D. Kapur (ed.), International Conference on Automated Deduction 1992, *Lecture Notes in Computer Science* vol. 607, Springer, 148–162. doi: [10.1007/3-540-55602-8_162](https://doi.org/10.1007/3-540-55602-8_162).

M. Fuchs (1997), “Evolving Combinators”, in *Automated Deduction—CADE-14*, W. McCune, International Conference on Automated Deduction 1997, *Lecture Notes in Computer Science* vol. 1249, Springer, 416–430. doi: [10.1007/3-540-63104-6_42](https://doi.org/10.1007/3-540-63104-6_42).

M. Fuchs, D. Fuchs and M. Fuchs (1997), “Solving Problems of Combinatory Logic with Genetic Programming” in *Genetic Programming 1997: Proceedings of the Second Annual Conference*, J. R. Koza, et al. (eds.), Genetic Programming 1997, Morgan Kaufmann, 102–110. cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/gp1997/Fuchs_1997_spclGP.pdf.

A. Hamfelt and J. F. Nilsson (1998), “Inductive Synthesis of Logic Programs by Composition of Combinatory Program Schemes”, in *Logic-Based Program Synthesis and Transformation*, P. Flener (ed.), International Workshop on Logic Programming Synthesis and Transformation 1998, *Lecture Notes in Computer Science* vol. 1559, Springer, 143–158. doi: [10.1007/3-540-48958-4_8](https://doi.org/10.1007/3-540-48958-4_8).

P. S. di Fenizio (2000), “A Less Abstract Artificial Chemistry”, in *Artificial Life VII: Proceedings of the Seventh International Conference on Artificial Life*, M. A. Bedau, et al. (eds.), International Conference on Artificial Life 2000, MIT Press, 49–53.

J. Tromp (2002), “Kolmogorov Complexity in Combinatory Logic”. citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.7359&rep=rep1&type=pdf.

M. Stay (2005), “Very Simple Chaitin Machines for Concrete AIT”, *Fundamenta Informaticae* 68, 231–247. [arXiv:cs/0508056](https://arxiv.org/abs/cs/0508056).

M. Szudzik (2006), “An Elegant Pairing Function”, Special NKS 2006 Conference. szudzik.com/ElegantPairing.pdf.

F. Briggs and M. O’Neill (2008), “Functional Genetic Programming and Exhaustive Program Search with Combinator Expressions”, *International Journal of Knowledge-Based and Intelligent Engineering Systems* 12, 47–68. doi: [10.3233/KES-2008-12105](https://doi.org/10.3233/KES-2008-12105).

P. Liang, M. I. Jordan and D. Klein (2010), “Learning Programs: A Hierarchical Bayesian Approach”, in *ICML’10: Proceedings of the 27th International Conference on International Conference on Machine Learning*, International Conference on Machine Learning 2010, Omnipress, 639–646. doi: 10.5555/3104322.3104404.

P. Scott and J. Fleuriot (2012), “A Combinator Language for Theorem Discovery”, in *Intelligent Computer Mathematics*, J. Jeuring, et al. (eds.), International Conference on Intelligent Computer Mathematics 2012, *Lecture Notes in Computer Science* vol. 7362, Springer, 371–385. doi: 10.1007/978-3-642-31374-5_25.

M. Bellia and M. E. Occhiuto (2013), “DNA Tiles, Wang Tiles and Combinators”, *Fundamenta Informaticae* 133, 105–121. doi: 10.3233/FI-2014-1065.

L. R. Williams (2015), “Programs as Polypeptides”. [arXiv:1506.01573](https://arxiv.org/abs/1506.01573).

W. P. Worzel and D. MacLean (2015), “SKGP: The Way of the Combinator”, in *Genetic Programming Theory and Practice XII*, M. Kotanchek, R. Riolo and W. Worzel (eds.), Springer, 53–57. doi: 10.1007/978-3-319-16030-6_4.

G. Gerules and C. Janikow (2016), “A Survey of Modularity in Genetic Programming”, in *2016 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, 5034–5043. doi: 10.1109/CEC.2016.7748328.

D. Xiao, J.-Y. Liao and X. Yuan (2018), “Improving the Universality and Learnability of Neural Programmer-Interpreters with Combinator Abstraction”. [arXiv:1802.02696](https://arxiv.org/abs/1802.02696).

Specific Programming Tasks

P. Koopman and R. Plasmeijer (1999), “Efficient Combinator Parsers”, in *Implementation of Functional Languages*, C. Clack, T. Davie and K. Hammond (eds.), Symposium on Implementation and Application of Functional Languages 1998, *Lecture Notes in Computer Science* vol. 1595, Springer, 120–136. doi: 10.1007/3-540-48515-5_8.

M. Finger and W. Vasconcelos (2000), “Sharing Resource-Sensitive Knowledge Using Combinator Logics”, in *Advances in Artificial Intelligence*, M. C. Monard and J. S. Sichman (eds.), Ibero-American Conference on Artificial Intelligence 2000, *Lecture Notes in Computer Science* vol. 1952, Springer, 196–206. doi: 10.1007/3-540-44399-1_21.

G. J. Pace (2000), “The Semantics of Verilog Using Transition System Combinators” in *Formal Methods in Computer-Aided Design*, W. A. Hunt and S. D. Johnson (eds.), International Conference on Formal Methods in Computer-Aided Design 2000, *Lecture Notes in Computer Science* vol. 1954, Springer, 442–459. doi: 10.1007/3-540-40922-X_25.

D. Leijen (2001), “Parsec, a Fast Combinator Parser”. users.cecs.anu.edu.au/~Clem.Baker-Finch/parsec.pdf.

A. J. Kennedy (2004), “Pickler Combinators”, *Journal of Functional Programming* 14, 727–739. doi: 10.1017/S0956796804005209.

I. D. Peake and S. Seefried (2004), “A Combinator Parser for Earley’s Algorithm”.
citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.504.5998&rep=rep1&type=pdf.

J. N. Foster, et al. (2007), “Combinators for Bidirectional Tree Transformations: A Linguistic Approach to the View-Update Problem”, *ACM Transactions on Programming Languages and Systems* 29, 17–81. doi: 10.1145/1232420.1232424.

R. Lämmel (2007), “Scrap Your Boilerplate with XPath-Like Combinators”, *ACM SIGNPLAN Notices* 42, 137–142. doi: 10.1145/1190215.1190240.

S. Mazanek and M. Minas (2007), “Graph Parser Combinators”, in *Implementation and Application of Functional Languages*, O. Chitil, Z. Horváth and V. Zsók (eds.), Symposium on Implementation and Application of Functional Languages 2007, *Lecture Notes in Computer Science* vol. 5083, Springer, 1–18. doi: 10.1007/978-3-540-85373-2_1.

S. D. Swierstra (2009), “Combinator Parsing: A Short Tutorial”, in *Language Engineering and Rigorous Software Development*, A. Bove, et al. (eds.), International LerNet ALFA Summer School on Language Engineering and Rigorous Software Development 2008, *Lecture Notes in Computer Science* vol. 5520, Springer, 252–300. doi: 10.1007/978-3-642-03153-3_6.

T. Neward (2012), “The Working Programmer—Building Combinators”, *MSDN Magazine* 27. docs.microsoft.com/en-us/archive/msdn-magazine/2012/january/the-working-programmer-building-combinators.

H. Bergier (2020), “How Combinatory Logic Can Limit Computing Complexity”, *EPJ Web Conference* 244. doi: 10.1051/epjconf/202024401009.

Extensions & Applications

Extensions of Combinators

J. T. Kearns (1969), “Combinatory Logic with Discriminators”, *The Journal of Symbolic Logic* 34, 561–575. doi: 10.2307/2270850.

N. D. Belnap Jr. (1970), “Strict Polyadic Combinatory Logics”, “Strict Polyadic Lambda Calculus”, “Semantics of Strict Polyadicity”, University of Pittsburgh, Faculty of Arts and Sciences, unpublished manuscripts.

N. D. Goodman (1972), “A Simplification of Combinatory Logic”, *The Journal of Symbolic Logic* 37, 225–246. doi: 10.2307/2272970.

- A. S. Kuzičev (1973), “Consistent Extensions of Pure Combinatory Logic” (in Russian; abstract in English), *Vestnik Moskovskogo Universiteta . Serija I . Matematika, Mehanika* 28, 76–81.
- F. B. Fitch (1980), “A Consistent Combinatory Logic with an Inverse to Equality”, *Journal of Symbolic Logic* 45, 529–543. doi: 10.2307/2273420.
- R. K. Meyer, K. Bimbó and J. M. Dunn (1998), “Dual Combinators Bite the Dust” (abstract), *Bulletin of Symbolic Logic* 4, 463–464. doi:10.2307/420959.
- T. Jech (1999), “Some Results on Combinators in the System TRC”, *The Journal of Symbolic Logic* 64, 1811–1819. doi: 10.2307/2586813.
- K. Bimbó (2003), “The Church–Rosser Property in Dual Combinatory Logic”, *The Journal of Symbolic Logic* 68, 132–152. doi: 10.2178/jsl/1045861508.
- K. Bimbó (2004), “Semantics for Dual and Symmetric Combinatory Calculi”, *Journal of Philosophical Logic* 33, 125–153. jstor.org/stable/30226803.
- Y. Akama (2006), “SN Combinators and Partial Combinatory Algebras”, in *Rewriting Techniques and Applications*, International Conference on Rewriting Techniques and Applications 1998, *Lecture Notes in Computer Science* vol. 1379, Springer, 302–316. doi: 10.1007/BFb0052378.
- A. Di Pierro, C. Hankin and H. Wiklicky (2006), “On Reversible Combinatory Logic”, *Electronic Notes in Theoretical Computer Science* 135, 25–35. doi: 10.1016/j.entcs.2005.09.018.
- J. H. Andrews (2007), “An Untyped Higher Order Logic with Y Combinator”, *The Journal of Symbolic Logic* 72, 1385–1404. doi: 10.2178/jsl/1203350794.
- P. Hancock (2008), “AMEN: The Last Word in Combinators (A Naperian Meditation)”. citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.585.6287.
- B. Jay and T. Given-Wilson (2011), “A Combinatory Account of Internal Structure”, *Journal of Symbolic Logic* 76, 807–826. doi: 10.2178/jsl/1309952521.

Combinatory Grammars & Linguistics

- P.-L. Curien (1986), *Categorical Combinators, Sequential Algorithms and Functional Programming*, Research Notes in Theoretical Computer Science, Pitman Publishing. doi: 10.1007/978-1-4612-0317-9.
- R. D. Lins (1987), “On the Efficiency of Categorical Combinators as a Rewriting System”, *Software: Practice and Experience* 17, 547–559. doi: 10.1002/spe.4380170807.
- M. Steedman (1987), “Combinatory Grammars and Parasitic Gaps”, *Natural Language & Linguistic Theory* 5, 403–439. doi: 10.1007/BF00134555.

- M. Steedman (1988), “Combinators and Grammars”, in *Categorical Grammars and Natural Language Structures*, E. Bach, R. T. Oehrle and D. Weeler (eds.), Springer, 417–442. doi: [10.1007/978-94-015-6878-4_15](https://doi.org/10.1007/978-94-015-6878-4_15).
- H. Yokouchi and T. Hikita (1988), “A Rewriting System for Categorical Combinators with Multiple Arguments”, *SIAM Journal on Computing* 19, 78–97. doi: [10.1137/0219005](https://doi.org/10.1137/0219005).
- J. F. Nilsson (1989), “A Case-Study in Knowledge Representation and Reasoning with Higher-Order Combinators”, in *Scandinavian Conference on Artificial Intelligence 89: Proceedings of the SCAI '89*, H. Jaakkola and S. Linnainmaa (eds.), Scandinavian Conference on Artificial Intelligence 1989, IOS Press, 37–48.
- P. Simons (1989), “Combinators and Categorical Grammar”, *Notre Dame Journal of Formal Logic* 30, 241–261. doi: [10.1305/ndjfl/1093635081](https://doi.org/10.1305/ndjfl/1093635081).
- H. Yokouchi (1989), “Church-Rosser Theorem for a Rewriting System on Categorical Combinators”, *Theoretical Computer Science* 65, 271–290. doi: [10.1016/0304-3975\(89\)90104-7](https://doi.org/10.1016/0304-3975(89)90104-7).
- J. Villadsen (1991), “Combinatory Categorical Grammar for Intensional Fragment of Natural Language”, in *Scandinavian Conference on Artificial Intelligence—91*, B. H. Mayoh (ed.), Scandinavian Conference on Artificial Intelligence 1991, IOS Press, 328–339.
- I. Biskri and J.-P. Desclés (1995), “Applicative and Combinatory Categorical Grammar (from Syntax to Functional Semantics)” in *Recent Advances in Natural Language Processing*, R. Mitkov and N. Nicolov (eds.), John Benjamins Publishing Company, 71–84. doi: [10.1075/cilt.136.08bis](https://doi.org/10.1075/cilt.136.08bis).
- J.-P. Desclés (2004), “Combinatory Logic, Language, and Cognitive Representations,” in *Alternative Logics. Do Sciences Need Them?*, P. Weingartner (ed.), Springer, 115–148. doi: [10.1007/978-3-662-05679-0_9](https://doi.org/10.1007/978-3-662-05679-0_9).
- I. Biskri (2005), “Applicative and Combinatory Categorical Grammar and Subordinate Constructions in French”, *International Journal on Artificial Intelligence Tools* 14, 125–136. doi: [10.1142/S0218213005002028](https://doi.org/10.1142/S0218213005002028).
- M. Steedman and J. Baldridge (2006), “Combinatory Categorical Grammar”, in *Encyclopedia of Language & Linguistics (Second Edition)*, K. Brown (ed.), Elsevier, 610–621. doi: [10.1016/B0-08-044854-2/02028-9](https://doi.org/10.1016/B0-08-044854-2/02028-9).
- F. Hoyt and J. Baldridge (2008), “A Logical Basis for the D Combinator and Normal Form in CCG”, in *Proceedings of ACL-08: HLT*, J. Allen, et al. (eds.), Annual Meeting of the Association for Computational Linguistics with the Human Language Technology Conference, Association for Computational Linguistics, 326–334. aclweb.org/anthology/P08-1038.pdf.

- J. Kang and J.-P. Desclés (2008), “Korean Parsing Based on the Applicative Combinatory Categorical Grammar”, in *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation*, R. E. O. Roxas (ed.), Pacific Asia Conference on Language, Information and Computation, De La Salle University, 215–224. aclweb.org/anthology/Y08-1021.pdf.
- C. Bozşahin (2012), “The Lexicon, Argumenthood and Combinators”, in *Combinatory Linguistics*, De Gruyter, 31–42. library.oapen.org/bitstream/id/1eb35764-bbdd-444c-b2ad-13981a63672f/1005450.pdf.
- C. Bozşahin (2012), “Syntacticizing the Combinators”, in *Combinatory Linguistics*, De Gruyter, 43–60. library.oapen.org/bitstream/id/1eb35764-bbdd-444c-b2ad-13981a63672f/1005450.pdf.
- C. Bozşahin (2013), *Combinatory Linguistics*, De Gruyter, doi: 10.1515/9783110296877.
- M. Steedman (2018), “The Lost Combinator”, *Computational Linguistics* 44, 613–620. doi: 10.1162/coli_a_00328.
- S. T. Piantadosi (2021), “The Computational Origin of Representation”, *Minds and Machines* 31, 1–58. doi: 10.1007/s11023-020-09540-9.

Confusing Issues

The term “combinatory analysis” has nothing to do with “combinators”; it’s an earlier name for “combinatorics”, used for example in:

P. A. MacMahon (1915), *Combinatory Analysis*, The University Press (Cambridge).

“Combinatory” is also not used in the sense of combinators in:

E. Post (1936), “Finite Combinatory Processes—Formulation 1”, *The Journal of Symbolic Logic* 1, 103–105. doi: 10.2307/2269031.

“Combinator” is sometimes used as a fairly general term for a function or operation that combines computational operations, as in:

R. Milner (1982), “Four Combinators for Concurrency”, in *Proceedings of the First ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, Symposium on Principles of Distributed Computing, Association for Computing Machinery, 104–110. doi: 10.1145/800220.806687.

L. Cardelli and R. Davies (1999), “Service Combinators for Web Computing”, in *Transactions on Software Engineering*, IEEE, 309–316. doi: 10.1109/32.798321.

“The Combinator” is a recent combinatorial tool for idea generation, that seems to have no relation to combinators:

J. Han, et al. (2018), *The Combinator—A Computer-Based Tool for Creative Idea Generation Based on a Simulation Approach*, Cambridge University Press. doi: 10.1017/dsj.2018.7.

Y Combinator is a startup accelerator founded by P. Graham et al. in 2005; its name is derived from the fixed-point combinator, but otherwise it is unrelated:

Y Combinator (accessed March 22, 2021). www.ycombinator.com.

Thanks

Thanks to Henk Barendregt, Ariela Böhm, Emanuele Böhm, Michele Böhm, Martin Bunder, Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, Roger Hindley, Oleg Kiselyov, Jan Willem Klop, Gerd Mitschke, Alberto Pettorossi and Adrian Rezus for suggestions and material for this bibliography, and to Paige Bremner, Amy Simpson and the University of Illinois library for extensive work in tracking down documents.

Cite as: S. Wolfram (2021), “A Bibliography of Combinators”. wolframcloud.com/obj/sw-writings/Combinators/bibliography.pdf.