Irodalomjegyzék

[1] Aczel, Peter: Frege structures and the notions of proposition, truth and set. In Barwise, J. (szerk.): *The Kleene Symposium*. Studies in Logic sorozat, 101. köt. 1980, North-Holland, 31–60. p.

[2] Barendregt, Hendrik P.: *The Lambda Calculus, Its Syntax and Semantics*. 1984, North-Holland.

[3] Bird, Richard – Wadler, Philip: *Introduction to Functional Programming*. 1988, Prentice Hall.

[4] Church, Alonzo: A set of postulates for the foundation of logic. 33., 34. évf. (1932, 1933), *Annals of Math.*, 346–366., 839–864. p.

[5] Church, Alonzo: A formulation of the simple theory of types. 5. évf. (1940), *Journal of Symbolic Logic*, 56–58. p.

[6] Church, Alonzo: *The Calculi of Lambda Conversion*. 1941, Princeton University Press.

[7] Curry, Haskell B.: Grundlagen der kombinatorischen logik. 52. évf. (1930), *Amer. J. Math.*, 509–536., 789–834. p.

[8] Curry, Haskell B.: Functionality in combinatory logic. 20. évf. (1934), *Proc. Nat. Acad. Science USA*, 584–590. p.

[9] Curry, Haskell B. – Feys, Robert: *Combinatory Logic*. 1. köt. 1958, North-Holland.

[10] Csörnyei Zoltán: *Típuselmélet*. Előadásjegyzet, Eötvös Loránd Tudomány-egyetem, 2005.

[11] Damas, Luis – Milner, Robin: Principal type schemes for functional programs. In *Proc. 9th ACM Symposium on Principles of Programing Languages*. 1982, 207–212. p.

[12] Diller, Antoni: *Compiling Functional Languages*. 1988, John Wiley & Sons.

[13] Felleisen, Matthias – Flatt, Matthew: *Programming Languages and Lambda Calculi*. Course Notes, Utah University, 2003.

[14] Frege, Gottlob: *Grundgesetze der Arithmetik, begriffsschriftlich abgeleitet*. I–II. köt. Jena, 1893, 1903, H. Pohle. Utánnyomás: Wissenschaftliche Buchgesellschaft, Darmstadt, 1962.

[15] Goldberg, Mayer: *An Introduction to the Lambda Calculus*. Course Notes, Ben-Gurion University, 2000.

[16] Hanák Péter – Hanák Dávid – Csala Viktor: *Magasabbrendű funkcionális programozás*. Előadásjegyzet, Budapesti Műszaki és Gazdaságtudományi Egyetem, 2003.

[17] Hankin, Chris: *Introduction to Lambda Calculus*. Course Notes, Imperial College London, 2003.

[18] Harrison, John: *Introduction to Functional Programming*. Course Notes, University of Cambridge, 1997.

[19] Hindley, J. Roger: The principal type scheme of an object in combinatory logic. 146. évf. (1969), *Transactions of the American Mathematical Society*, 29–60. p.

[20] Hindley, J. Roger – Seldin, Jonathan P.: *Introduction to Combinators and λ-Calculus*. 1986, Cambridge University Press.

[21] Horváth Zoltán: Funkcionális programozási nyelvek elemei. In Nyékyné Gaizler Judit (szerk.): *Programozási nyelvek*. 2003, Kiskapu, 589–636. p.

[22] Hudak, Paul: Conception, evolution, and application of functional programming languages. 21. évf. (1989) 3. sz., *ACM Computing Surveys*.

[23] Joy, Mike S. – Rayward-Smith, V. J. – Burton, F. W.: Efficient combinator code. 10. évf. (1985) 3-4. sz., *Comput. Lang.*, 211–224. p.

[24] Ker, Andrew D.: *Lambda Calculus*. Course Notes, Oxford University, 2003.

[25] Kleene, Stephen C.: λ-definability and recursiveness. 2. évf. (1936), *Duke Math. J.*, 340–353. p.

[26] Kleene, Stephen C. – Rosser, John B.: The inconsistency of certain formal logics. 36. évf. (1936), *Annals of Math.*, 630–636. p.

[27] Milner, Robin: A theory of type polimorphism in programming. 17. évf. (1978), *Journal of Computer and Systems Sciences*, 348–375. p.

[28] Mitchell, John C.: *Foundations for Programming Languages*. 1996, The MIT Press.

[29] Mycroft, Alan: Polymorphic type schemes and recursive definitions. In *International Symposium on Programming*. Lecture Notes in Computer Science sorozat, 167. köt. 1984, Springer-Verlag, 217–228. p.

[30] Novitzká, Valerie: Church's types in logical reasoning on programming. 6. évf. (2006) 2. sz., *Acta Electrotechnica et Informatica*, 27–31. p.

[31] Ong, C-H. Luke: *Lambda Calculus*. Course Notes, Oxford University, 1997.

[32] Paulson, Lawrence C.: *Foundations of Functional Programming*. Course Notes, University of Cambridge, 1996.

[33] Peyton Jones, Simon L.: *The Implementation of Functional Programming Languages*. 1987, Prentice Hall.

[34] Pierce, Benjamin C.: *Types and Programming Languages*. 2002, MIT Press.

[35] Pierce, Benjamin C. (szerk.): *Advanced Topics in Types and Programming Languages*. 2005, MIT Press.

[36] Plasmeijer, Rinus – Eekelen, Marko van: *Functional Programming and Parallel Graph Rewriting*. 1993, Addison-Wesley.

[37] Plotkin, Gordon D.: Call-by-name, call-by-value, and the λ-calculus. 1. évf. (1975), *Theoretical Computer Science*, 125–159. p.

[38] Robinson, J. Alan: A machine oriented logic based on the resolution principle. 12. évf. (1965) 1. sz., *Journal of the Association for Computing Machinery*, 23–41. p.

[39] Rosen, Barry K.: Tree-manipulating systems and Church–Rosser theorems. 20. évf. (1973) 1. sz., *Journal of the ACM*, 160–187. p.

[40] Roversi, Luca: *λ-calculus as a Programming Language*. Course Notes, Università di Torino, 1999.

[41] Ruzsa Imre: *Bevezetés a modern logikába*. 2000, Osiris.

[42] Schönfinkel, Moses: Über die bausteine der mathematischen logik. 92. évf. (1924), *Math. Annalen*, 305–316. p.

[43] Sestoft, Peter: Demonstrating lambda calculus reduction. In Mogensen, T. – Schmidt, D. – Sudburough, I. H. (szerk.): *The Essence of Computation: Complexity, Analysis, Transformation*. Lecture Notes in Computer Science sorozat, 2566. köt. 2002, Springer-Verlag, 420–435. p.

[44] Sørensen, Morten H. – Urzyczyn, Pawel: *Lectures on the Curry-Howard Isomorphism*. Studies in Logic and the Foundations of Mathematics sorozat, 149. köt. 2006, Elsevier.

[45] Turing, Alan M.: On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of London Mathematical Society*. 42. köt. 1936–37, 230–265. p.

[46] Turing, Alan M.: Computability and λ-definability. 2. évf. (1937), *J. Symbolic Logic*, 153–163. p.

[47] Whitehead, Alfred North – Russell, Bertrand: *Principia Mathematica*. Cambridge, Anglia, 1910, Cambridge University Press.