

Lambda-calculus and Combinators in the 20th Century ^{*}

Felice Cardone [†] J. Roger Hindley [‡]

Contents

1	Introduction	723
2	Pre-history	725
3	1920s: Birth of Combinatory Logic	726
4	1930s: Birth of λ and Youth of CL	730
4.1	Early λ -calculus	730
4.2	CL in the 1930s	733
5	1940s and 1950s: Consolidation	737
5.1	Simple type theory	737
5.2	Abstract reduction theory	738
5.3	Reductions in CL and λ	739
5.4	Illative systems	739
6	Programming languages	741
6.1	John McCarthy and LISP	742
6.2	Peter Landin	742
6.3	Corrado Böhm: λ -calculus as a programming language	743
7	Syntactical developments	744
7.1	Contributions from the programming side	745
7.2	Theory of reductions	748
8	Types	751
8.1	The general development of type theories	751
8.1.1	Types as grammatical categories	752
8.1.2	Types as sets	753
8.1.3	Types as objects	755
8.1.4	Types as propositions	759
8.2	Early normalization proofs	766
8.3	Higher-order type theories	768
8.4	Intersection types and recursive types	772
8.5	Algorithms for simple types	774
9	Models for λ	777
9.1	Scott's D_∞ model	777
9.2	Computational and denotational properties	778
9.3	Other models	781
10	Domain theory	783
10.1	Classical domain theory	783
10.2	Effective domains and Synthetic Domain Theory	788
10.3	Game semantics	789
	Bibliography	792

^{*}Published as Chapter 14, pp. 723–817, of *Handbook of the History of Logic* Volume 5, *Logic from Russell to Church*, Eds. D. M. Gabbay and J. Woods, Elsevier Co. (North-Holland), Amsterdam 2009. A list of errata etc. is available at www.users.waitrose.com/~hindley/

[†]felice@di.unito.it

[‡]j.r.hindley@swansea.ac.uk