The Logic of Curry and Church[*]

Jonathan P. Seldin
Department of Mathematics and Computer Science
University of Lethbridge
Lethbridge, Alberta, Canada
jonathan.seldin@uleth.ca
http://www.cs.uleth.ca/∼seldin

March 3, 2008

In the ordinary set-theoretic foundations of mathematics, functions are defined to be sets of ordered pairs in which no two distinct pairs have the same first element. Then if x is in the domain of a function f, $f(x)$ is defined to be the second element of the ordered pair whose first element is x. Thus, for example, the squaring function on the real numbers would be defined as

$$f = \{(x, x^2) : x \text{ is a real number}\},$$

and $f(3) = 3^2 = 9$.

However, there is another way to think of functions: as rules of correspondence. Thinking of the squaring function on the real numbers this way, we might represent it using the notation

$$x \mapsto x^2,$$

and the evaluation of the function at $x = 3$ would be written

$$(x \mapsto x^2)(3) = 3^2 = 9.$$

In fact, this idea occurred in the late 1920s to Alonzo Church, although instead of $x \mapsto x^2$ he wrote

$$\lambda x \,.\, x^2,$$

[*]This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

and for the evaluation of the function at $x = 3$ he wrote

$$(\lambda x \,.\, x^2)3 = 3^2 = 9.^1$$

This approach to functions is the basis of the work of both Alonzo Church (1903-1995) and Haskell Brooks Curry (1900-1982). Both of them set out to base logic and mathematics on functions instead of on set theory, and although neither of them achieved the kind of success he first sought, both of them wound up making major contributions to the foundations of logic and mathematics. Furthermore, the results of their work have come to be extremely important in theoretical computer science. The system introduced by Church is λ-*calculus*, the system of Curry is *combinatory logic*.[2]

For the convenience of readers, a complete description of lambda-calculus and combinatory logic is given in the Appendix of this article.

A history of both lambda-calculus and combinatory logic appears in this volume [Cardone and Hindley, 2006]. To avoid unnecessary duplication, I will follow a systematic rather than a historical order. Although combinatory logic began earlier than λ-calculus, it is easier to understand combinatory logic if one already knows λ-calculus. Therefore, I will begin with the latter.

I would like to thank Roger Hindley and Martin Bunder for their helpful comments and suggestions.

1 Lambda-calculus

1.1 Background of Alonzo Church

Alonzo[3] Church was born on June 14, 1903 in Washington D. C. He received his A.B. at Princeton University in 1924. In later life, he reported that he

[1]This was not his original notation; his original notation was $\{\lambda x[x^2]\}(3) = 3^2 = 9$; see [Church, 1932]. But the notation given in the text had become standard by 1941 [Church, 1941]. By the way, Church did not write "=" here, but wrote "conv" for *conversion*. In this paper, I will write $=_*$ for an unspecified conversion relation.

[2]Curry came to use the name "combinatory logic" to refer to both his system, which he called *synthetic combinatory logic*, and λ-calculus, but most people now use the name only for a variant of Curry's system.

[3]The material in this section comes from [Enderton, 1995] and from the transcript of an interview of Church by William Aspray on 17 May 1985, available on the web at `http://libweb.princeton.edu/libraries/firestone/rbsc/finding_aids/mathoral/pmc05.htm`.

was drawn to things of a fundamental nature, and while he was an undergraduate he published a minor paper about the Lorentz transformation, the foundation of the special theory of relativity [Church, 1924]. He also caught the eye of Oswald Veblen, who made important contributions to projective and differential geometry and topology, and who was interested in foundational issues in the sense of postulate systems for geometry. After Church graduated, the Department of Mathematics at Princeton gave him a fellowship to go to graduate school. While he was a graduate student, he published another paper related to foundational matters, [Church, 1925]. He completed his Ph.D. in 1927 with a dissertation on systems in which the axiom of choice might be false [Church, 1927].

After finishing his Ph.D., Church spent two years on a National Research Fellowship, one year at Harvard and another year at Göttingen and Amsterdam. Then, in 1929, he returned to Princeton to take up a faculty position in the Department of Mathematics. During this period, he published [Church, 1928], which was related to foundational matters, though not to those which later led him to introduce lambda-calculus.

1.2 Origins of Lambda-Calculus

Until he completed his Ph.D., Church's approach to foundational matters in the above mentioned papers been informal, but while he held his National Research Fellowship, in 1928–29, Church began to develop a system of formal logic that he hoped would be adequate for the foundations of mathematics. He wanted to build a type-free system in which all variables occurred only bound, and he hoped to avoid contradiction by restricting the law of excluded middle.[4] He included in this system notations for function application and abstraction, which had previously appeared in the work of Peano and Frege [Cardone and Hindley, 2006, §2], but apparently Church was not then familiar with these works [Cardone and Hindley, 2006, §4.1]. However,

[4]The idea was that without the law of excluded middle, the existence of the Russell class R would not imply a contradiction, since it would not be true that R is the sort of thing which can be an element of R, and without the law of excluded middle it would be impossible to prove $R \in R \vee R \notin R$. In Church's system, as in Curry's, this paradox would be represented not in terms of classes, but in terms of predicates, so that instead of the Russell class R, there would be a Russell predicate R, such that RR is equivalent to $\sim (RR)$. Then, if R is not automatically in the domain of R, one would need $RR \vee \sim (RR)$ to derive the paradox.

unlike Peano and Frege, Church gave formal rules for them and began to analyze their consequences in depth. This system was published in [Church, 1932]. His atomic constants were all logical operators, Π for a relative universality,[5] Σ for existence, & for conjunction, $\sim$ for negation, ι for the definite article, and A for a kind of abstraction operator. Well-formed formulas were formed from these atomic constants and variables by means of application and λ-abstraction, where in the latter, $\lambda x \,.\, M$ was well-formed whether or not x occurred free in M. In addition to his axioms and rules for the logical constants, he had three rules of procedure for abstraction, application, and substitution: (1) change of bound variables (α-conversion, using modern terminology); (2) β-contraction, where, in the redex $(\lambda x \,.\, M)N$, x must occur free in M; (3) the reverse of 2. There were also two rules for logical operators:[6] (4) for well formed M and N,

$$MN \vdash \Sigma M,$$

and 5) For well formed M, N, and P,

$$\Pi MN, MP \vdash NP.$$

The remaining postulates were all axioms. The paper concluded with proofs of a combination of the deduction and generalization theorems for Π:

1. Let x be a variable occurring free in M, and let $\vdash \Sigma(\lambda x \,.\, M)$, and let $M \vdash N$. Then if x occurs free in N, then $\vdash \Pi(\lambda x \,.\, M)(\lambda x \,.\, N)$. If x does not occur free in N, then $\vdash N$.

2. If $\vdash \Sigma M$ and x does not occur free in N and $Mx \vdash Nx$, then $\vdash \Pi MN$.

3. If $\vdash \Sigma M$ and x does not occur free in N and $Mx \vdash N$, then $\vdash N$.

Church did not have in this paper a general notion of deduction from premises, only deduction from one premise.

Very soon after writing [Church, 1932], Church discovered that his list of axioms was contradictory, so in [Church, 1933] he modified it to avoid that

[5]ΠMN is to be distinguished from $(\forall x)(Mx \supset Nx)$. The latter requires that MP and NP be defined for all possible values of P, whereas the former only requires that NP be defined and true whenever MP is true, so that NP may be undefined if MP is defined and not true.

[6]Stated here in modern notation.

contradiction, discussed the possibility of a consistency proof, derived a large number of results in the system, discussed the term that represents the Russell paradox, and ended with his proposal, now well-known, for representing the natural numbers, here given in modern notation: see Remarks 5 and 6 in Appendix A. In terms of these definitions, Church defined

$$\mathcal{N} \equiv \lambda x \, . \, (\forall P)(P\mathbf{1} \wedge (\forall u)(Pu \supset P(\boldsymbol{\sigma}u)) \supset Px).$$

Church noted that the first two Peano axioms and the induction axiom follow immediately from these definitions.

By this time, Church had two graduate students who later became two of the leaders of mathematical logic, Steven Cole Kleene (1909–1994) and John Barkley Rosser (1907–1989). Church set Kleene the problem of studying his system, especially the theory of natural numbers. Both worked on matters connected with Church's formal logic: Kleene, who started first, worked on the system itself, while Rosser worked on its connection to Curry's combinatory logic.

The first result of these studies was [Kleene, 1934], which showed that arguments using the intuitive rule of proof by cases could be carried out in that system. This made possible definition by cases, which Kleene needed for his dissertation. Kleene modified Church's system in relatively minor ways, the most important being that $\lambda x \, . \, M$ was to be well-formed only when x occured free in M. This modification was retained in all later publications having to do with untyped λ-calculus by Church, Kleene, and Rosser. In this paper Kleene also published for the first time a definition of combination by defining I to be $\lambda x \, . \, x$ and, following Rosser, J to be $\lambda uxyz \, . \, ux(uzy)$ and then taking combinations to be terms formed from I and J by applications. Kleene then reproduced a result from Rosser's as yet unpublished dissertation that for any term a combination could be found to which the term converts. He needed this result to show that definitions by cases and proofs by cases could be carried out in Church's system. This was the first published result relating λ-calculus and combinatory logic. (Rosser's dissertation was later published as [Rosser, 1935a; Rosser, 1935b]; see § 2.3 below.)

Next came Kleene's dissertation [Kleene, 1935], which was based on Church's definition of natural numbers. Kleene showed that Church's system was adequate for the elementary theory of natural numbers by proving the third and fourth Peano axioms. In the course of doing this, he showed that a number of (recursive) functions could be represented as λ-terms, including functions that Church had not thought could be represented that way. Rosser had also found some such representations in work that was never published, and by the fall of 1933, Church began to speculate that every function of positive integers that can be effectively calculated can be represented as a λ-term. By early the following year, Church was convinced of what is now known as "Church's Thesis." See [Rosser, 1984, p. 334].

But meanwhile, in late 1934, Kleene and Rosser succeeded in proving that Church's system of logic was inconsistent by showing that Richard's paradox could be derived in it. [Kleene and Rosser, 1935]. Their proof also applied to the system of combinatory logic then being developed by Curry (see §2.3 below).

Church, Kleene, and Rosser responded to this inconsistency by extracting from Church's system of logic what is now called the pure λ-calculus: the part of the system involving the formation of λ-terms and the rules for reduction and conversion. In Church's Thesis they already had an indication of the importance of pure λ-calculus, and in the years following the discovery of the inconsistency, they obtained a number of major results, results which fully justified treating the pure λ-calculus as a system on its own.

1.3 First Important Results using Pure $\boldsymbol{\lambda}$-Calculus

The first result needed to justify the pure λ-calculus was a consistency proof, and one was published as [Church, 1935]. This result was actually proved for an extension of the pure λ-calculus which included an axiom, **2** (here taken as the truth value for "true"), and a new operator δ with axioms to ensure that $\delta MN = \mathbf{2}$ if M and N are in normal form and they differ only by changes of bound variables and $\delta MN = \mathbf{1}$ (here taken as the truth value for "false") if M and N are in normal form and are not the same except for changes of bound variables. Church's intention was to incorporate logic by using **2** and **1** as the truth values, and he defined the logical connectives for negation and conjunction simply as functions on natural numbers and then defined the logical quantifiers in such a way that when a quantifier formula is true it converts to **2** but otherwise has no normal form. The consistency result depended on some results that Church obtained with Rosser and which were published soon after in [Church and Rosser, 1936]. This latter paper proved what is now known as the "Church-Rosser Theorem" or "confluence theorem," which is now considered basic in term rewriting. It says that if $M \rhd P$ and $M \rhd Q$, then there is a term N such that $P \rhd N$ and $Q \rhd N$. See

Theorem 5 in Appendix A below.

This was followed by a result of Kleene showing that the functions that can be represented as λ-terms are precisely those which are recursive in the Herbrand-Gödel sense [Kleene, 1936b]. This gave important support to Church's Thesis, which received additional support a couple of years later by work of Alan Turing [1936–1937; 1937a; 1937b].

But the most important result derived at this time from pure λ-calculus, the result for which Church is most famous, is his proof that there are unsolvable problems [Church, 1936c]. In this paper, Church proved that there is no effectively computable procedure (in the form of a recursive function) which will decide whether or not two given terms convert to each other, and he also proved that there is no effectively computable procedure that will determine whether or not a term has a normal form. A Gödel numbering converts these problems into problems of elementary number theory, and Church ended his paper by pointing out that his result implies that any formal system which includes elementary arithmetic and is ω-consistent is undecidable. In [Church, 1936b; Church, 1936a], Church extended this result to the undecidability of first-order logic. Rosser, in his [1936], showed that the assumption of ω-consistency is unnecessary, both for Gödel's incompleteness theorem and Church's undecidability result.

There was one more use of λ-calculus made in the 1930s by Church and Kleene, who showed in their [1936] how to represent some ordinal numbers as λ-terms. Church discussed this representation and its relation to constructivity in his [1938].

A summary of all results on λ-calculus up to this point appears in [Church, 1941].

After these results were obtained, Church, Kleene, and Rosser gave up on basing a system of logic on λ-calculus. Kleene moved on to recursive function theory, starting with [Kleene, 1936a]. When he wrote further on ordinal notations, he did not deal with representations in λ-calculus [Kleene, 1938; Kleene, 1944; Kleene, 1955].[7] Rosser's work on combinatory logic will be taken up in §2.3 below. And Church himself gave up on trying to construct a type-free logic based on λ-calculus.

[7]Kleene says in his [1981] that he stopped using λ-calculus as a basis for his work because of the negative reactions of audiences to his early papers which did use λ-calculus. The only later paper he wrote that was based in λ-calculus is [Kleene, 1962].

1.4 Church's Type Theory

Although he had given up on basing type-free logic on λ-calculus, he did use λ-calculus as a basis for a system of simple type theory [Church, 1940]. In this system, the types were defined as in Definition 4 of Appendix B, where the atomic types were o for propositions and ι for individuals.[8] Terms were allowed in the system only if they could be assigned types by the typing rules of TACh$_\lambda$ of Definition 5 of Appendix B. The atomic terms included terms for the boolean logical operators negation and disjunction and also a universal quantifier for every type and, in addition, a definite descriptions operator $\iota_{\alpha(o\alpha)}$ of type $(\alpha \rightarrow o) \rightarrow \alpha$ for every type α. The axioms and rules were sufficient to give higher-order classical logic with equality and descriptions, and there was an axiom which asserted the existence of two distinct individuals. In the paper, Church proved the Peano postulates using the typed version of his representation of the natural numbers, and he also proved that the definition of functions by primitive recursion could be carried out in the system.

A few years later, Church [1951] used this type theory in formulating a logic of sense and denotation. And still later, he applied this type theory in a number of papers including his [1973; 1974; 1989; 1993].

Otherwise, he published nothing further involving λ-calculus.

Leon Henkin [1950] proved Church's type theory complete. In his [1963], he gave a formulation in which the only primitive constants are the identity relations over each type. Peter B. Andrews extended it to include transfinite types in his [1965], and more recently he has included it in an introductory textbook, his [1986].

This kind of system has been extended in ways important for theorem provers and proof development systems. The first such extensions were due to Per Martin-Löf [1975; 1984] and Jean-Yves Girard [1971; 1972]. Martin-Löf's type theory was used by Robert Constable and his team as the basis for the Nuprl proof development system [Constable and others, 1986], while the systems of Girard were extended by Coquand and Huet to the calculus of constructions [Coquand and Huet, 1988], which formed the basis for the Coq proof assistant.[9]

For a general overview of typed systems see [Kamareddine *et al.*, 2004].

[8]But Church wrote '$(\beta\alpha)$' for $\alpha \rightarrow \beta$.

[9]See `http://coq.inria.fr/`.

2 Combinatory Logic

2.1 Background of Haskell Brooks Curry

Haskell[10] Brooks Curry was born on September 12, 1900 at Millis, Massachusetts. His parents ran the School for Expression, now known as Curry College. Curry entered Harvard University as an undergraduate in 1916, and originally intended to study medicine. However, when the United States entered World War I, he changed his major to mathematics. His original motivation was patriotic: he thought knowing mathematics would enable him to serve his country in an artillery unit. After his eighteenth birthday, he joined the Student Army Training Corps, but before he saw any action, the war ended.

Curry graduated from Harvard with the degree of A.B. in 1920 and began his graduate work in a program in electrical engineering at the Massachusetts Institute of Technology that involved working half-time at the General Electric Company. However, he became dissatisfied with the engineers because they did not share his interest in why results were true, so after two years he returned to Harvard University as a student of physics. For the first year he was in that program, 1922–23, he was a half-time research assistant to P. W. Bridgeman. He received his M.A. in physics in 1924. He then went back to mathematics, where he entered a Ph.D. program and was eventually assigned by George D. Birkhoff a dissertation topic on differential equations. However, he was becoming increasingly interested in logic, and this led him eventually to combinatory logic.

2.2 Origins of combinatory logic

Curry's first look at a work on logic came in May of 1922, when he first studied [Whitehead and Russell, 1910–1913]. He immediately noticed that of the two rules of inference in Chapter 1 of that work, the rule of substitution is substantially more complicated than the rule of modus ponens.[11] He

[10]Much of the information in this section and those which follow comes from my personal association with Curry. I was his research assistant at the Pennsylvania State University from 1964–1966, helped him train new assistants at the University of Amsterdam in 1966–67, finished my doctorate under his direction in 1968, and then became a co-author. I remained in touch with him for the rest of his life.

[11]The notes Curry made at the time can be seen on `http://www.sadl.uleth.ca` under the title "Works of Haskell Curry." Search by date for the item of 20 May 1922.

gradually began to feel a desire to break down the rule of substitution into simpler rules, and during 1926–1927 this brought him to the combinators I, B, C, and W[12] and the application operation. At that time it was standard to make underlying assumptions about substitution as part of what Curry called the "prelogic" of a system, not stated explicitly. Curry felt that it was important to write these assumptions out formally, so that they could be analyzed. At this point, he decided that he wanted to work in logic, and so he abandoned his work in differential equations. However, there was nobody at Harvard who could supervise a dissertation on this topic.

As a result, for the year 1927–1928, Curry took an instructorship at Princeton University.[13] During that year, he discovered in a library search that his work had been anticipated by Moses Schönfinkel [1924]. Schönfinkel had shown that all the combinators could be defined in terms of K and S, and Curry immediately saw the point of K and incorporated it in his system along with I, B, C, and W. However, for some years, Curry did not see the point of replacing I, B, C, and W by S. Curry also noticed that Schönfinkel had taken equality (conversion) informally, without giving any axioms or rules for it. Nevertheless, Curry realized that despite Schönfinkel's work, there was scope for a dissertation on this subject.

For this purpose, Curry sought the advice of Oswald Veblen. Veblen and Curry then sought information about Schönfinkel from the topologist Paul Alexandroff, who was visiting Princeton that year. Alexandroff informed them that Schönfinkel had returned to the USSR and was in a mental instution, so there was no chance that Curry could work with him. However, Schönfinkel's paper was based on a talk he had given to the Göttingen Mathematical Society in December of 1920, and there were people there who knew about this subject: Heinrich Behmann and Paul Bernays.

So Curry decided to go to Göttingen to get his doctorate. He wrote his first paper, [Curry, 1929], as part of an application for a grant to make that trip. In this paper, Curry followed Schönfinkel in taking K and S as basic combinators, application as the operation that formed terms, and $=_*$ (which Curry then wrote as '$=$' and thought of as equality) as the only other primitive, but unlike Schönfinkel he gave formal axioms and rules for his

[12]See Appendix C for the definitions. Curry's original notation for these combinators was I, B, C, and W. Curry first used the sans-serif notation for combinators and other constants in [Curry and Feys, 1958].

[13]He just missed Alonzo Church, who had completed his Ph.D. and left to study elsewhere for two years. See § 1.1 above.

system. He called attention to the fact that the reason for the interest in this system is its *combinatory completeness*, which says that if X is any term in which some of the variables $x_1, x_2, \ldots, x_n$ occur, then there is a term A, in which none of these variables occur, such that

$$Ax_1x_2 \ldots x_n =_* X.$$

Curry wanted this A to be uniquely determined by X and the variables $x_1, x_2, \ldots, x_n$, so one of his rules was a generalization of rule (ζ)[14]: if

$$Xx_1x_2 \ldots x_n =_* Yx_1x_2 \ldots x_n$$

for variables $x_1, x_2, \ldots, x_n$ which do not occur in X or Y, then

$$X =_* Y.$$

However, this rule was too complicated from the point of view of analyzing the logical rule of substitution into simpler rules, so Curry wanted to find a set of postulates which would imply this rule. In [Curry, 1929], he gave axioms that imply this rule in a special case, which he predicted would be important for the analysis of substitution.

Curry spent the year 1928–1929 in Göttingen completing his dissertation [Curry, 1930]. His official supervisor was David Hilbert, but he actually worked with Bernays. Church was in Göttingen for half of that year, and in his later years Curry remembered seeing a manuscript of Church's with many occurrences of the symbol 'λ', but at the time he had no idea that this manuscript had any connection with what he was doing. He did not realize there was a connection until he saw [Church, 1932] a few years later.

The system of [Curry, 1930] is based on the combinators B, C, W, and K, and had addional atomic terms Q (for logical identity, which here meant conversion), Π (for the universal quantifier), P (for implication), and Λ (for conjunction). The term forming operation was application, and formulas were of the form '$\vdash X$' for 'X is provable.' The axioms included

$$\vdash \Pi(\mathsf{W}(\mathsf{CQ})),$$

from which the reflexive law of Q can be proved (using the rules given below) and axioms from which the generalization of rule (ζ) follows. For example, the axiom giving the conversion rule for K is

$$\vdash \mathsf{Q}(\mathsf{K}XY)X.$$

[14]See Appendix A, Remark 1.

The rules included rules from which it can be proved that Q has all the properties of equality and weak conversion, and the following logical rules:

Rule Π. $\Pi X \vdash XY$,

Rule P. $\mathsf{P}XY, X \vdash Y$,

Rule Λ. $X, Y \vdash \Lambda XY$.

For this system, Curry proved combinatory completeness by giving an algorithm for what he later called *bracket abstraction*, although at this time he did not use that notation, and he described his algorithm as defining, for a combination X in which some of the variables $x_1, x_2, \ldots, x_n$ occur, a term A in which none of those variables occur such that

$$Ax_1x_2 \ldots x_n =_* X.$$

The first step is to remove the variables from $x_1, x_2, \ldots, x_n$ which do not occur in X using K and B. The second step is to repeat variables as often as they occur in X using W and B. Third, the variables must be arranged in the order in which they occur in X using C and B. And finally, parentheses that occur in X are inserted using B and I.

Curry also gave a consistency proof for the system he had defined. However, no logical axioms for Π, P, or Λ were given, and the development of the logical part of the system was left for the future.

2.3 Early Development of Combinatory Logic

After finishing his dissertation, Curry joined the faculty of the Pennsylvania State College in September 1929. At first, he was not happy there; he had been at Harvard and Princeton, and Penn State did not then support research,[15] he felt that Penn State was an institution of much lower standing. However, his choice of logic as a field had cut down his opportunities, and the Great Depression began that fall, so finding a job elsewhere was out of the question.

Since Penn State had heavy teaching loads, Curry did not have much time for research there for the next three years. There was also no support for graduate students at that time. In fact, Curry's first graduate student,

[15] It is very different now.

Edward Cogan, did not start working with him until 1950. So, unlike Church, Curry had to continue his development of combinatory logic on his own.

However, he did manage to do some research during his first years. During the very first year of teaching at Penn State, he published [Curry, 1931], in which he added postulates for the universal quantifier and proved some of its properties. But his further research had to wait until he was able to take a year off from teaching as the result of a National Research Fellowship, which he started in 1931. The fellowship was originally supposed to be for two years, from 1931 to 1933. However, after one year, his fellowship for the second year was cancelled because he had a job to go back to, and many others did not. Nevertheless, this one year, which he spent at the University of Chicago, was fruitful for him. He published [Curry, 1932; Curry, 1933; Curry, 1934a; Curry, 1934b; Curry, 1936] and the abstract [Curry, 1935], all of which were written or partly written during this year.

Of these, [Curry, 1932] contained a few technical modifications of [Curry, 1930], [Curry, 1933] introduced the notation of bracket abstraction, which allowed statements using bound variables to occur as abbreviations, and [Curry, 1934b] extended the logic by introducing axioms for equality and implication; it included a proof of the Deduction Theorem (equivalent to implication introduction in natural deduction) and the following generalization of it [Curry, 1934b, Theorem 14]:[16] *if*

$$A_1, A_2, \ldots, A_m \vdash X,$$

then there is a proof in which the variables $x_1, x_2, \ldots, x_n$ *do not occur of*

$$\vdash (\forall x_1)(\forall x_2)\ldots(\forall x_n)(A_1 \supset (A_2 \supset (\ldots(A_m \supset X)\ldots))).$$

The paper [Curry, 1934a] is an extended abstract of [Curry, 1936], which Curry had some trouble getting accepted. In these papers, Curry proposed a theory of predication or *functionality*, which would now be called a theory of categories or types.[17] These types are predicates, so what is now usually written as $X : \alpha$ was written by Curry as αX, so that the type, α is the predicate and the term, X is the subject. This is the source of the name of the

[16]Stated here using modern abbreviations.

[17]However, the theory of functionality was originally much less restrictive than type theory. It *mentioned* types, but it did not use them to define what it meant for a formula to be "well-formed," so that Curry originally anticipated allowing untyped terms in the system. This made it much less restrictive than Church's simple theory of types [Church, 1940].

Subject-Reduction Theorem. The main operator of this theory is F, which is characterized by the rule $\mathsf{F}\alpha\beta X, \alpha Y \vdash \beta(XY)$, so $\mathsf{F}\alpha\beta X$ is essentially equivalent to $X : \alpha \to \beta$.[18] This means that $\mathsf{F}\alpha\beta$ is the type of functions which map arguments of type α into values in type β. (This does not necessarily mean that α is the domain of such functions, only that this domain includes α.) Furthermore, subjects and predicates were not completely separated in the original theory. In [Curry and Feys, 1958, §10A3], Curry derives $\mathsf{K}X(\mathsf{WWW})$ for an arbitrary term X, and then applies a reduction step to deduce X, thus proving the full free system of [Curry and Feys, 1958, §10A3] inconsistent.

Curry had the original idea for functionality in December 1928.[19] Interestingly, his first notation for $\mathsf{F}\alpha\beta$ was $F_{\alpha\beta}$, but he found this awkward, and so tried the current notation, $[\alpha \to \beta]$, before rejecting that and settling on $F\alpha\beta$.[20] Among the categories Curry wanted in his system is H (originally called Pr), standing for *proposition.* Curry intended to use the theory of functionality to restrict logical postulates to some appropriately defined (by logical postulates) category of propositions as a means of avoiding any paradoxes.[21]

Meanwhile, Rosser published his dissertation [Rosser, 1935a; Rosser, 1935b], in which he presented a system of combinatory logic based on the combinators I and J. Rosser's rules for combinator conversion were as follows:

$$\begin{array}{rcl} \mathsf{I}X & \Leftrightarrow & X, \\ W(\mathsf{I}X) & \Leftrightarrow & WX, \\ W(\mathsf{J}XYZU) & \Leftrightarrow & W(XY(XUZ)), \end{array}$$

where W is an arbitrary term.

One of Rosser's aims was to eliminate any primitive rules whose conclusions are not uniquely determined by the premises. One such rule is Curry's rule for Π, which is $\Pi X \vdash XY$, since the choice of Y is arbitrary here.

[18]For Curry, this was roughly equivalent to $(\forall x)(\alpha x \supset \beta(Xx))$, but differed from that the way Church's ΠXY differed from $(\forall x)(Xx \supset Yx)$.

[19]See T281213A, the first item for December 13, 1928, under Works of Haskell Curry at `http://www.sadl.uleth.ca`.

[20]This became $\mathsf{F}\alpha\beta$ in [Curry and Feys, 1958]. See also [Seldin, 2003].

[21]Curry observed in these papers that there is a term, $[x](\neg(xx))$, which represents Russell's paradox (stated in terms of predicates in stead of sets), and he proposed to find postulates from which it could not be proved that the application of this term to itself is a proposition. However, it was important for Curry that this term exist in his system.

Rosser avoided this problem by using Church's Π, which is characterized by the rule

$$\Pi XY, XZ \ \vdash \ YZ.$$

To get the effect of a universal quantifier, he followed Church in using ΠE, where E is Church's E, which is defined by

$$E \equiv \lambda x \, . \, (\exists X)(Xx).$$

2.4 The Paradox of Kleene and Rosser and Curry's New Program for Combinatory Logic

Curry's papers up to this time continued the development of combinatory logic along the lines he had previously set out, and the abstract [Curry, 1935] had taken him up to the representation of set theory in combinatory logic. He seemed on the point of showing that combinatory logic could serve as a foundation for logic and mathematics.

However, Kleene and Rosser proved that Curry's system of [Curry, 1934b], like that of Church, was inconsistent. Their result appeared in print in [Kleene and Rosser, 1935], but Curry knew about it before that. He had been in correspondence with Rosser, who had informed him in December, 1933[22] that he thought he could prove Church's system inconsistent. Curry mentioned in a reply dated December 18, 1933 that he would like to see the proof of the contradiction Rosser mentioned as soon as possible. In a letter to Curry postmarked March 2, 1934, Rosser said that he and Kleene had finished the first draft of a proof that Church's system is inconsistent. Then in a joint letter dated November 15, 1934, Kleene and Rosser sent Curry a manuscript of [Kleene and Rosser, 1935]. The text of their letter began

> We are sorry to say that we have a proof of the inconsistency of your system of formal logic. We are also sorry that we did not obtain our results in definite form in time to let you know sooner.

Curry's reply to both of them, dated November 19, 1934, began

> Your letter of November 15th enclosing reprint and manuscript has just been received. You begin by saying that you are sorry

[22]Rosser did not date the letter, so I am relying on a postmark which is smudged. I think the date is about December 10, but it might be a few days later.

> that you have done this dreadful thing. I am afraid that I can not share your grief. In the first place I do not believe a word of it;– you are not sorry but you are full of that profound satisfaction which comes from doing something worth while. In the second place I have no occasion to feel sorry for myself; for I feel that whatever advances or deepens our knowledge of the subject to which I have devoted so much attention advances my interest as well. Consequently I am not inclined to feel sorry but just the opposite, and I hasten to congratulate you both.

A little later, he said

> . . . you seem to imply that you have refuted my position by showing that my system as you call it is inconsistent. On the contrary your results are in general agreement with my position and are in a sense a continuation of my work. You allude for example to my paper [reference to [Curry, 1934b]]. I proved in that paper (Theorem 14) that any logical system of a certain type, which I called an L system had a certain property, say T. As I understand it you have essentially proved that any system having the property T is "inconsistent". Very well, that simply means that any L system is inconsistent, which is a very much stronger result that (*sic*) which I had proved. Whether or not you needed my result in order to establish the stronger one I do not know; but in any case your result does not contradict mine. The conclusion is that a consistent system of logic, if any, must be weaker than an L system which is a positive result of some value.

Curry then went on to point out that the contradiction requires some of the postulates of his [1934b], and that the systems of his earlier papers, which did not include these assumptions, might well be consistent.

Thus, whereas Church, Kleene, and Rosser reacted to the inconsistency by abandoning the attempt to find a type-free formal logic that could serve as a basis for mathematics, Curry did not. He had already developed a strategy for dealing with contradictions, namely using a predicate to stand for "proposition" together with his theory of functionality to restrict the postulates for the logical operators. The contradiction of Kleene and Rosser only meant for him that he was required to face this problem earlier than he had anticipated in the development of the subject.

The first decision Curry made on learning of the paradox was that he needed to study it in detail "so as to lay bare its central nerve."[23] The direct result of this study was [Curry, 1941b], which shows how to derive Richard's paradox in a very general setting. Meanwhile, he also published [Curry, 1941c], in which he revised the fundamental postulates of combinatory logic taking into account some of the ideas of Rosser's thesis [Rosser, 1935a; Rosser, 1935b]. This was closely followed by [Curry, 1941a], in which he proved the consistency of a system stronger than that of [Curry, 1930]. Finally, he found a much simpler contradiction that could be derived under the assumptions of the Kleene-Rosser paradox in [Curry, 1942b]. This simpler contradiction, known as *Curry's paradox*, can be stated easily as follows: Suppose the system has an operator P for implication, so that $X \supset Y$ is an abbreviation for $\mathsf{P}XY$. Suppose that the following rules hold, either as primitive rules or as derived rules:

(Eq) $$\frac{X \quad X =_* Y}{Y,}$$

($\supset$E) $$\frac{X \supset Y \quad X}{Y.}$$

Suppose also that

(PW) $$(X \supset (X \supset Y)) \supset (X \supset Y)$$[24]

is provable for arbitrary terms X and Y. Then any term can be proved in the system. To see this, let Y be any term, and let $X \equiv \mathsf{Y}(\lambda y.X \supset (X \supset y))$, where Y is a fixed-point combinator,[25] so that $X =_* (X \supset (X \supset Y))$. Then we can prove Y as follows:

1.	$(X \supset (X \supset Y)) \supset (X \supset Y)$	By hypothesis
2.	$X \supset (X \supset Y)$	By 1 and Rule (Eq)
3.	X	By 2 and Rule (Eq)
4.	$X \supset Y$	By 2, 3, and Rule ($\supset$E)
5.	Y	By 3, 4, and Rule ($\supset$E)

[23][Curry and Feys, 1958, §8S1].

[24]This will be provable if, in addition to the above rules, the natural deduction rule ($\supset$I) for introducing implication (also known as the Dedution Theorem) is either a primitive rule or a derived rule.

[25]See Appendix A.

Before he discovered this paradox, Curry had assumed that the contradiction was caused by his postulates for the universal quantifier, but this paradox made it clear that the problem was the postulates for implication alone.[26]

Curry's next step was to set out a program to find type-free systems based on combinatory logic or λ-calculus that would be consistent. His idea was to define a restricted class of terms that could represent propositions.

For this purpose, he noted that there are different possible sets of logical operators that could be taken as primitive.

First, he could use P and Π, as he had done in his early papers. In order to meet Rosser's criterion that the conclusion of every primitive rule should be uniquely determined by the premises, he added a new atomic constant E with the property that $\mathsf{E}X$ is provable for every X. Then he characterized P and Π by the rules

Rule P. $\mathsf{P}XY, X \vdash Y$,

Rule Π. $\Pi X, \mathsf{E}Y \vdash XY$.

A second possible set consisted of Church's Π, which Curry renamed Ξ. Its characteristic rule is

Rule Ξ. $\Xi XY, XU \vdash YU$.

In terms of Ξ, both P and Π can be defined as follows:

$$\begin{aligned} \mathsf{P} &\equiv \lambda xy \,.\, \Xi(\mathsf{K}x)(\mathsf{K}y), \\ \Pi &\equiv \Xi\mathsf{E}. \end{aligned}$$

(Because Curry accepted the combinator K, he did not need a separate atomic term for implication the way Church did.) Conversely, Ξ can be defined in terms of P and Π by

$$\Xi \equiv \lambda xy \,.\, \Pi(\lambda u \,.\, \mathsf{P}(xu)(yu)),$$

or, in a more commonly recognized notation,

$$\Xi \equiv \lambda xy \,.\, (\forall u)(xu \supset yu).$$

[26]For further confirmation of this, see [Seldin, 2000].

A third set of primitives consisted of just F, which is characterized by the following rule:

Rule F. $\mathsf{F}\alpha\beta X, \alpha U \vdash \beta(XU)$.

Of course, F can be defined in terms of P and Π by

$$\mathsf{F} \equiv \lambda xyz \,.\, \Pi(\lambda u \,.\, \mathsf{P}(xu)(y(zu))).$$

or, in the more commonly recognized notation,

$$\mathsf{F} \equiv \lambda xyz \,.\, (\forall u)(xu \supset y(zu)).$$

Using essentially the same definition, F can be defined in terms of Ξ:

$$\mathsf{F} \equiv \lambda xyz \,.\, \Xi x(\lambda u \,.\, y(zu)).$$

But Ξ can also be defined in terms of F. In fact, it can be done in two ways.

$$\Xi \equiv \lambda xy \,.\, \mathsf{F}xy\mathsf{I},$$

and

$$\Xi \equiv \lambda xy \,.\, \mathsf{F}x\mathsf{I}y.$$

Curry then proposed to study three different classes of systems:

1. $\mathcal{F}_1$: systems of *functionality*, based on the logical operator F and Rule F,

2. $\mathcal{F}_2$: systems of *restricted generality* based on Ξ and Rule Ξ, and

3. $\mathcal{F}_3$: systems of *universal generality* based on P and Π and Rules P and Π.

For each kind of system, Curry proposed to find restricted forms of introduction rule(s) corresponding to the characteristic rules read as elimination rules, in the sense which is standard in natural deduction. Thus, for systems $\mathcal{F}_1$, he wanted to find a class of terms called *canonical*[27] for which he could prove the following theorem, where lower case Greek letters represent canonical terms.

[27]Curry actually called them *canonical obs* or *canobs*. This is because he was using the word 'ob' to refer to terms. See Definition 1 of Section §3 below.

Theorem 1 (Stratification Theorem) *If* Γ *is any set of assumptions, and* $\Gamma, \alpha x \vdash \beta X$*, where the variable* x *does not occur free in* Γ*,* α*, or* β*, then* $\Gamma \vdash \mathsf{F}\alpha\beta(\lambda x \,.\, X)$*.*

For systems $\mathcal{F}_2$, Curry wanted to define a class of canonical terms from which he could prove:

Theorem 2 (Deduction Theorem for Ξ) *If* Γ *is any set of assumptions and* $\Gamma, \alpha x \vdash \beta$*, where* x *does not occur free in* Γ *or* α*, then* $\Gamma \vdash \Xi\alpha(\lambda x \,.\, \beta)$*.*

Similarly, for systems $\mathcal{F}_3$, he wanted to define a class of canonical terms from which he could prove the following two theorems:

Theorem 3 (Deduction Theorem for P) *If* Γ *is any set of assumptions, and* $\Gamma, \alpha \vdash \beta$*, then* $\Gamma \vdash \mathsf{P}\alpha\beta$*.*

Theorem 4 (Universal Generalization) *If* Γ *is any set of assumptions, and* $\Gamma \vdash \alpha$*, where* x *does not occur free in* Γ*, then* $\Gamma \vdash \Pi(\lambda x \,.\, \alpha)$*.*

Of course, Curry wanted consistency proofs for all these systems. To obtain them, he proposed to use systems like those of Gentzen and prove results corresponding to the *Cut Elimination Theorem* [Gentzen, 1934].[28]

In his earlier versions of the above systems, the axioms of each system were presented as axiom schemes. But Curry was also interested in replacing these schemes by single axioms. This meant incorporating the definition of canonical terms into the logic using a new atomic constant H to indicate canonicalness. For each kind of system $\mathcal{F}_i$, Curry wanted to define a corresponding finitely axiomatized system $\mathcal{F}_i^*$.

Then the United States entered World War II. For a combination of patriotic and financial reasons, Curry took a leave of absence from Penn State starting in June, 1942, in order to do applied mathematics for the U. S. government. Before going, he wrote up his current research results and published them as [Curry, 1942a; Curry, 1942c].

Just before his leave of absence began, he finally came to understand the role of the combinator S after reading [Rosser, 1942]. Rosser had based his postulates on his favorite combinators, I and J. But most important for Curry's understanding of S, Rosser had defined abstraction by induction on

[28]See Section §3 below.

the structure of the term. When Curry translated Rosser's postulates into the more usual combinators, he finally saw that S could be used in a very elegant definition of bracket abstraction (see Appendix C). Because Curry was about to leave Penn State for his war effort, he was unable to write this up as a paper, so he sent his notes to Rosser.

Actually, the idea of defining abstraction by a direct induction was not really due to Rosser. The definition in [Rosser, 1942] had originally been given by Church for combinations in λ-calculus in [Church, 1935, §3] and in [Church, 1941, Chapter IV]. But it was [Rosser, 1942] that stimulated Curry to see how to use S.

2.5 Curry's War Work and Combinatory Logic

Most of the work Curry did during World War II was unrelated to combinatory logic. The greater part of it was on the fire control problem, which is the mathematics of aiming a projectile at a moving target, and he also did some work with I. J. Schoenberg on splines [Curry and Schoenberg, 1947; Curry and Schoenberg, 1966]. However, at the end of the war and immediately after it, in late 1945 and early 1946, Curry was involved with the ENIAC[29] computer project. This eventually led to the idea of using combinators to combine computer programs [Curry, 1954].

Curry's experience with the ENIAC project also had an after-effect on his return to Penn State in September 1946. He tried to pursuade the administration to get some computing equipment and start what is now called a computer science program. He had no success in this, but persisted until a colleague pointed out to him that if he did succeed, he would be made director of the program without any increase in salary. He then gave up trying to start such a computer science program at Penn State, and went back to logic.

[29]The ENIAC was one of the first electronic computers, built during World War II for the purpose of calculating the trajectories of projectiles. See [Goldstine, 1946], now available at `http://ftp.arl.mil/ mike/comphist/46eniac-report/index.html`.

2.6 Curry's Work on Combinatory Logic in the Early Postwar Years

The first thing Curry did about combinatory logic after returning to Penn State was to write to Rosser about the notes he had sent him on basing combinatory logic on K and S and defining abstraction by induction on the structure of the term. He wanted to know whether Rosser wanted to write a joint paper about this. But Rosser had moved on to other things, so Curry then wrote his own paper [Curry, 1949].

About the same time, Curry became aware of [Newman, 1942], and thought that he could use th ideas in it to give a new proof of the Church-Rosser Theorem in an abstract setting. In a sense he succeeded in [Curry, 1952c], since he did give a proof of an abstract version of the theorem, but after the paper appeared it was discovered that λ-calculus does not satisfy one of the hypotheses he assumed. (And Newman's proof failed for a similar reason.)

Meanwhile, Curry attended the Tenth International Congress of Philosophy in Amsterdam in the summer of 1948. During that congress, he was approached by one of the editors of the new North-Holland Publishing Company and asked to write a short monograph of at most 100 pages on combinatory logic for their series "Studies in Logic". Curry sent them a philosophical manuscript he had ready, [Curry, 1951]. But, for combinatory logic, he realised that so much unpublished research had accumulated that an adequate treatment would need to be much longer than the editors envisaged.

Nevertheless, the idea of writing a book on this subject appealed to him. He knew that he was not a good expository writer, and this would be a drawback in writing a book of this kind. So he approached Robert Feys, who was on the faculty of the University of Louvain in Belgium and had an interest in combinatory logic, to become a co-author. Feys accepted, and Curry applied for and received a Fulbright fellowship to spend the year 1950–1951 at Louvain. There, work began on [Curry and Feys, 1958].

From the beginning, the plan was for a two volume work, with the first volume dealing with properties of conversion, called *pure combinatory logic*, and the second volume dealing with systems of logic in the ordinary sense based on combinatory logic or λ-calculus, called *illative combinatory logic*. However, as the writing proceeded, it became clear that this plan needed to be modified: in particular, since Curry was continuing to do new research, some of his newer results on the theory of functionality came to be included in the first volume.

Thus, [Curry and Feys, 1958] started with introductory material on formal systems (Chapter 1) and metatheory[30] (Chapter 2), and then had a chapter (Chapter 3) on λ-calculus. Although the λ-calculus was basically due to Church, it was Curry who fixed some notations that became standard in subsequent works on λ-calculus, for example the names for alpha-reduction and beta-reduction. He used the lower-case Greek alphabet for names of binary relations, and (α) and (β) became his designations for Church's Rules I and II from [Church, 1932]. The chapter on λ-calculus was followed by a chapter (Chapter 4) on the Church-Rosser Theorem.[31] This was followed by three chapters on pure combinatory logic: Chapter 5 was on various commonly used composite combinators and the relations between them, Chapter 6 was on what is called the "synthetic" theory of combinators (formal definition of the system, definition of abstraction, and the relation to λ-calculus), and Chapter 7 was on what was called "logistic foundations," which means formulations in terms of provability. Chapter 8 was a general introduction to illative combinatory logic, and Chapters 9–10 were on the theory of functionality.

The new research included in [Curry and Feys, 1958][32] grew out of work on the theory of functionality, that is, systems of type $\mathcal{F}_1$. Curry had originally hoped that a theory of functionality in which all terms were canonical would be consistent, and beginning in April 1954, he spent a considerable amount of time trying to prove this. His approach was to show first that all inferences by Rule Eq could be postponed to the end of any deduction and then to show that any deduction in which the only rule was Rule F could be transformed into a standard form. Deductions in which the only rule is Rule F are called *F-deductions*. During the summer of 1954, he noticed if ξX could be deduced by an F-deduction from a set of assumptions, then X must be irreducible and perhaps in some kind of normal form. At that time, the only kind of normal form he had for combinators was weak normal form, which means a combinatory term with no redexes of the form $\mathsf{K}XY$ or $\mathsf{S}XYZ$. Thus, when he first wrote down the *Normal Form Theorem* in his notes, it took the form "*If ξX can be deduced from a [set of assumptions] B, then X_λ has a normal*

[30]There called "epitheory" for reasons which will be discussed in § 4 below.

[31]The proof presented in this chapter has since been superceded by a simpler proof; see [Hindley and Seldin, 1986, Appendix 1].

[32]A careful study of Curry's notes from the period and also of the manuscript that he had received from Feys shows that this work was all done by Curry with no input by Feys.

form."[33] It took Curry some time to prove the theorem, and the problem he had stating the theorem for combinatory terms motivated him to introduce a new kind of reduction for combinators,[34] but once he concluded that it was true, he realized that the system whose consistency he was trying to prove was, in fact, inconsistent, and he soon discovered a contradiction in it [Curry, 1955]. He then investigated some more restricted systems, and found some that he could prove consistent [Curry, 1956]. This material appeared in [Curry and Feys, 1958, Chapters 9–10]. Chapter 9 was devoted to the *basic theory of functionality*, in which the types are those definable from a set of atomic types by means of the operation of forming $\mathsf{F}\alpha\beta$ from α and β. Thus, types in the basic theory of functionality are the types defined in Definition 4 in Appendix B below. The axioms were given by two axiom schemes:

$$(\mathsf{FK}) \qquad \mathsf{F}\alpha(\mathsf{F}\beta\alpha)\mathsf{K},$$

$$(\mathsf{FS}) \qquad \mathsf{F}(\mathsf{F}\alpha(\mathsf{F}\beta\gamma))(\mathsf{F}(\mathsf{F}\alpha\beta)(\mathsf{F}\alpha\gamma))\mathsf{S}.$$

For Curry, the types were just a special kind of terms, but since the atomic types were all assumed to be like F in being constants that do not head redexes, there was no point in considering conversions between types. Thus, Curry restricted the conversion rule, Rule Eq, to conversions between subjects only, and he called this Rule Eq′:

$$(\mathrm{Eq}') \qquad \frac{\alpha X \quad X =_* Y}{\alpha Y}$$

For F-deductions, Curry proved the Subject-Construction Theorem (i.e. that deductions always follow the construction of the subjects),[35] the Subject-Reduction Theorem (see Theorem 6 in Appendix B below), and the Stratification Theorem (see Theorem 1 in Section 2.3 above). He also proved that in a deduction with Rule (Eq′), all inferences by Rule (Eq′) can be postponed

[33] The normal form theorem for typed λ-calculus had been stated earlier by Alan Turing in an unpublished manuscript (see [Gandy, 1980]), but, of course, Curry did not know of this.

[34] See Strong Reduction below.

[35] This theorem is based on the way he worked out examples in his attempt to prove the consistency of the "full free" theory of functionality. He later formally defined this as what is now known as a "typing algorithm."

until the end and combined into one. In addition, he defined a version of basic functionality for λ-calculus, in which the axiom schemes (FK) and (FS) are replaced by a rule which he called an axiom scheme:

$$(\mathsf{F}\lambda)\qquad \frac{\begin{array}{c}[\alpha_1 x_1, \alpha_2 x_2, \ldots, \alpha_m x_m]\\ \beta X\end{array}}{\mathsf{F}\alpha_1(\ldots(\mathsf{F}\alpha_m\beta)\ldots)(\lambda x_1 \ldots x_m \,.\, X)}$$

Condition: X is a combination of the variables $x_1, x_2, \ldots,$ x_m only, and these variables do not occur free in any undischarged assumption.

Finally, he noted that if the subjects are removed from all steps of a deduction and if each occurrence of F is changed to an occurrence of P, the result is a valid deduction in the implication fragment of intuitionistic propositional calculus.[36] He concluded the chapter with natural deduction and Gentzen-style L-formulations, and he used the latter to give the first published proof of the Normal Form Theorem; see Theorem 7 of Appendix B below).

In [Curry and Feys, 1958, Chapter 10], Curry presented the results of both [Curry, 1955] and [Curry, 1956]. As mentioned before, these included a proof of consistency for a restricted system.

The problem that Curry originally had stating the Normal Form Theorem let him to introduce another innovation in [Curry and Feys, 1958]: *strong reduction* in Chapter 6. In pure combinatory logic, the natural reduction, weak reduction, lacks a property that both $\lambda\beta$- and $\lambda\beta\eta$-reduction satisfy, namely the property

$$(\xi)\qquad X \rhd Y \rightarrow [x]X \rhd [x]Y.$$

Curry studied the relation $\succ\!\!-$ defined by adding (ξ) to the definition of weak reduction and inserting into the definition of bracket abstraction (see Appendix C) the clause

(c) if x does not occur free in U, then $[x](Ux) \equiv U$.

[36]This observation, in extended form, is now well known as the formulas-as-types or propositions-as-types or Curry-Howard isomorphism. See [Howard, 1980].

He called this relation *strong reduction*. He showed that it corresponded to $\lambda\beta\eta$- reduction, provided the combinator I was assumed to be an atom instead of being defined as SKK.[37]

Six years after Curry and Feys began their work, the writing was finished, and [Curry and Feys, 1958] appeared two years later.

Meanwhile, Curry had his first graduate student, Edward J. Cogan. Cogan wrote a dissertation on set theory within illative combinatory logic, [Cogan, 1955]. The idea was to use the ideas about set theory of [Gödel, 1940; von Neumann, 1925; von Neumann, 1928] within the framework developed by Curry.[38] Cogan chose to represent the set theory of [Gödel, 1940] in a system of type $\mathcal{F}_2^*$ based on Ξ, with F defined in terms of Ξ. Unfortunately, his system turned out to be inconsistent [Titgemeyer, 1961]. The problem came from an axiom that Curry had suggested to Cogan.

2.7 Curry's Later Work on Combinatory Logic

After the publication of [Curry and Feys, 1958], Curry himself, in preparation for writing the second volume of [Curry and Feys, 1958] turned his attention to systems of type $\mathcal{F}_2$, systems of restricted generality. In his [1960], Curry proved a version of the Deduction Theorem for Ξ, namely: *if* $\xi_1, \xi_2, \ldots, \xi_m, \eta$ *are all canonical, and if*

$$\Gamma, \xi_1 x_1, \xi_2 x_1 x_2, \ldots \xi_m x_1 x_2 \ldots x_m \;\vdash\; \eta,$$

where the variables $x_1, x_2 \ldots, x_m$ *do not occur free in* Γ*, then*

$$\Gamma \;\vdash\; \Xi\xi_1(\lambda x_1 \,.\, \Xi(\xi_2 x_1)(\ldots(\lambda x_{m-1} \,.\, \Xi(\xi_m x_1 x_2 \ldots x_{m-1})(\lambda x_m \,.\, \eta))\ldots)).$$

Canonical terms were defined in a such a way that the system was not really stronger than first order logic. Curry was unable to obtain this result by iterating Theorem 2 above because he had failed to assume an axiom generating rule. This problem arises in ordinary predicate calculus, where one needs in addition to modus ponens either a rule of universal generalization or else a specification that the universal closure of any axiom is an axiom. In

[37]The definition of a relation corresponding to $\lambda\beta$-reduction was not attempted. This involved some technical complications and was not done until [Mezghiche, 1984; Mezghiche, 1989; Mezghiche, 1997].

[38]In a sense this set-theory already contained a combinator concept in some of its axioms, [Curry and Feys, 1958, p. 10], [Cardone and Hindley, 2006].)

his original work, Curry left out this specification (which I called above the axiom generating rule), although in the paper he assumed it in the multiple form.

In his [1961], Curry gave a proof of a sort of cut-elimination theorem for a system related to a system of type $\mathcal{F}_2$ the way a Gentzen L-system is related to a natural deduction system.

Meanwhile, Curry had three more Ph.D. students. Kenneth Loewen [1962] wrote on the Church-Rosser Theorem and the Standardization Theorem[39] for strong reduction, and Bruce Lercher [1963] wrote on the relationship between strong reduction and the representation of recursive functions and functionals of higher type. Meanwhile, Luis E. Sanchis [1963] formulated Church's type theory within combinatory logic using the theory of functionality for the types, and proved the consistency of that system.

On April 16, 1964, Robert Feys died. This left Curry alone to work on the second volume of [Curry and Feys, 1958]. He realized that he needed collaborators and also realized that he needed to make a further study of the foundations of logic, and in particular, of Gentzen-style[40] proof theory before writing the second volume. The result of this study was [Curry, 1963]. Thus, he did not get back to work on the second volume until 1964. That September, J. Roger Hindley arrived at Penn State to do postdoctoral work with Curry. Hindley had just completed his dissertation [1964] on the Church-Rosser Theorem, and in 1965 Curry invited him to join the project. Also in September, 1964, I began my graduate studies with Curry. When Curry retired from Penn State in 1966 and took up a position at the University of Amsterdam, I accompanied him. After I finished my dissertation [1968], Curry invited me to join the project as well, and this is how the second volume became [Curry *et al.*, 1972], and some of the material of [Curry *et al.*, 1972, Chapters 14–16] came from [Seldin, 1968, Chapters 3–5]. When Curry arrived in Amsterdam, another graduate student, Martin W. Bunder began studying with him. Curry ran a seminar at which Martin Bunder and I presented our results, and although Martin did not become a co-author of [Curry *et al.*, 1972], his work did contribute to some of the later chapters.

The first chapter of [Curry *et al.*, 1972], Chapter 11,[41] is devoted to updates to the material on pure combinatory logic in the first volume. Curry's

[39]The Standardization Theorem says roughly speaking that any reduction can be carried out by reducing redexes from left to right.

[40]See § 3 below.

[41]The chapter numbering continues from [Curry and Feys, 1958].

original intention was to begin with a chapter on philosophical questions, but that proved too difficult to write, and so only two subsections of the first section of the chapter are devoted to that. In addition to the updates to the material on pure combinatory logic, the chapter contains a proposal due to Curry for a kind of system, which he called a *C-system*, which would treat combinatory logic and λ-calculus simultaneously. The idea was that since the various illative systems can be defined in both λ-calculus and combinatory logic, there should be a kind of system which could be instantiated to either. This idea was applied throughout the book, althoughin several later places it was still necessary to treat combinatory logic and λ-calculus separately.

Chapter 12 is an introduction to illative systems. Originally, illative combinatory logic meant combinatory logic with operators for the logical connectives and quantifiers, but by the mid-1960s, Curry had decided that any applied combinatory logic or λ-calculus should be considered an illative system.

Chapter 13 is entirely on the representation of natural numbers and recursive functions in combinatory logic. To allow for the various known ways of coding numerals as combinatory or lambda terms, Curry introduced two new atoms $\mathbf{0}$ and $\boldsymbol{\sigma}$, and postuated what he called an *iterator* Z with the property that

$$\mathsf{Z}\,(\underbrace{\boldsymbol{\sigma}(\boldsymbol{\sigma}(\ldots(\boldsymbol{\sigma}}_{n}\,\mathbf{0})\ldots)) \;\rhd \lambda xy\,.\; \underbrace{x(x(\ldots(x}_{n}\,y)\ldots)).$$

This system could be instantiated to any of the numerical codings by substituting appropriate terms for $\mathbf{0}$, $\boldsymbol{\sigma}$, and and constructing an appropriate pure combinator to serve as Z. For the coding used by Church, $\mathbf{0}$ and $\boldsymbol{\sigma}$ would be replaced by Church's definitions of those terms and Z would be replaced by I. The chapter included a proof that all recursive functions can be represented. It also discussed the use of terms with types, or functional characters, and ended with a discussion of Gödel's functionals of finite type. Included in the chapter was an undecidability result of Curry's for $\lambda\mathsf{K}$-calculus and equivalent combinatory systems, a result that was stronger than Church's result of his [1936c]. Curry [1969b] proved that the only *conversion-invariant*[42] recursive sets of terms were the empty set and the set of all terms. Since the set of terms which convert to a given term and the set of terms with

[42]A set of terms is conversion invariant if, whenever X is in the set and $X =_* Y$, then Y is in the set. Curry called these sets *equation-invariant*.

a normal form were conversion-invariant, this implied Church's two results, but Curry's proof required the combinator K.[43]

Chapter 14 is devoted to the theory of functionality. Curry had, by this time, split Rule (Eq) into separate rules for subject and predicate:

$$\text{(Eqs)} \qquad \frac{\alpha X \quad X =_* Y}{\alpha Y}$$

$$\text{(Eqp)} \qquad \frac{\alpha X \quad \alpha =_* \beta}{\beta X}$$

Here, (Eqs) was the rule (Eq′) of [Curry and Feys, 1958, Chapter 9], which is the only one appropriate for basic functionality, but for stronger systems of functionality (Eqp) would also be appropriate. Curry called a system using these rules a *separated system*, and it is with separated systems that functionality finally became what is now recognized as a type theory, in which Curry's type assignment statements αX could be rewritten $X : \alpha$. Chapter 14 also included a number of technical improvements to the systems of [Curry and Feys, 1958, Chapters 9–10]. One of the improvements is the proof that every term with a type has a *principal type scheme* or *principal functional character*. This result was proved by two different methods in [Curry, 1969a] and [Hindley, 1969].

Chapter 15 is devoted to systems of restricted generality, that is systems of type $\mathcal{F}_2$, based on Ξ. In it, a form of the Deduction Theorem for Ξ like Theorem 2 above that could be iterated was proved, and with this and a cut elimination theorem for a Gentzen-style L-system, the consistency was proved of a system whose rules for canonicalness are essentially of the strength of first-order logic. This settled the theory of $\mathcal{F}_2$ systems. However, the $\mathcal{F}_2^*$ systems were definitely not settled. Recall that the $\mathcal{F}_2^*$ systems are those in which canonicalness is defined using the theory of functionality (which can be embedded in these systems) in conjunction with a term H representing the predicate of being a proposition. A system of this kind, called $\mathcal{F}_{21}^*$, was proposed in [Seldin, 1968, Chapter 4], using an idea of Martin Bunder to restrict the deduction theorem by placing a restriction only on the antecedent:

[43]A very similar undecidability theorem had been proved by D. Scott in lecture notes in 1963, which Curry did not see until his own proof was substantially written. See [Curry, 1969b, p.10] or [Curry *et al.*, 1972, p.251, footnote 7].

i.e., to take for the deduction theorem

$$\Gamma, Xx \vdash Yx \ \Rightarrow\ \Gamma, \mathsf{L}X \vdash \Xi XY,$$

where x does not occur free in Γ, X, or Y, and L is FEH, the category of one-place predicates.[44] In [Curry *et al.*, 1972, §15C1] a different system also called $\mathcal{F}_{21}^{*}$ is defined. Both of these systems were defined to avoid some contradictions found by Martin Bunder in his [1969]. However, as indicated below, neither turned out to be consistent.

Chapter 16 of [Curry *et al.*, 1972] was on systems of universal generality, that is systems of type $\mathcal{F}_3$, based on Π and P. The basic conclusion of this chapter is that systems of universal generality are essentially equivalent to systems of restricted generality as long as E is present in the latter. This contradicts Curry's original idea that $\mathcal{F}_3$ systems might be stronger than $\mathcal{F}_2$ systems. Curry also originally assumed that both $\mathcal{F}_2$ and $\mathcal{F}_3$ systems were stronger than $\mathcal{F}_1$ systems, but if $\mathcal{F}_1$ systems are taken without the separated equality rules and Ξ is defined in terms of F in either of the ways indicated above, the result is equivalent to the corresponding $\mathcal{F}_2$ system. Thus, there are equivalent versions of all three kinds of systems.

Chapter 17 was on type theory based on combinatory logic and the theory of functionality. The chapter extended some of the results of [Sanchis, 1963; Sanchis, 1964]. One of the extensions was to a type theory with transfinite types, as in [Andrews, 1965].

The writing of [Curry *et al.*, 1972] was finished in May of 1970, and after that Curry retired from the University of Amsterdam and returned to State College, Pennsylvania. Except for the year 1971–1972, he spent the rest of his life there. But before he left Amsterdam, his last Ph.D. student, Martin Bunder, finished his dissertation [Bunder, 1969]. Bunder's thesis showed how to interpret both ZF and NBG set theories in a system of restricted generality. His system was similar to the two systems $\mathcal{F}_{21}^{*}$ of Seldin and Curry, but he defined L to be FAH, where A is a category of individuals and it is not the case that $\vdash \mathsf{A}X$ for every term X.

In his [1969, Chapter 2], Bunder gave a number of contradictions that could be derived in various illative systems as a motivation for the limitations of his own system (for which there was no consistency proof). Bunder later proved Seldin's system $\mathcal{F}_{21}^{*}$ inconsistent in his [1974]. This proof of

[44] All previous proposals for this kind of system had restricted the deduction theorem by means of restrictions on both the antecedent and the consequent.

inconsistency does not apply to the system $\mathcal{F}_{21}^*$ of [Curry *et al.*, 1972], but in his [Bunder, 1976] Bunder proved this system inconsistent as well. Finally, in their [1978], Bunder and Meyer proved inconsistent any system of illative combinatory logic which includes the following:

1. Rule (Eq),
2. Modus ponens: $\mathsf{P}XY, X \vdash Y$,
3. A deduction theorem as follows: $\Gamma, X \vdash Y \Rightarrow \Gamma, \mathsf{H}X, \mathsf{H}Y \vdash \mathsf{P}XY$.
4. $\mathsf{H}X, \mathsf{H}Y \vdash \mathsf{H}(\mathsf{P}XY)$,
5. $X \vdash \mathsf{H}X$, and
6. $\vdash \underbrace{\mathsf{H}(\mathsf{H}(\ldots(\mathsf{H}}_{n} X)\ldots))$ for a fixed n and every X.

The case $n = 2$ of item 6, namely $\vdash \mathsf{H}(\mathsf{H}X)$ for every X, was of interest because it is equivalent to $\vdash \mathsf{LH}$, which is necessary to derive any reasonable set of axioms from the deduction theorem. Finding a set of axioms equivalent to the deduction theorem had been one of Curry's major aims ever since he formulated his new program in the 1930s, and this result proved that impossible.

But even before Bunder published his contradictions, Curry became suspicious of $\vdash \mathsf{H}(\mathsf{H}X)$ for every X. His reason was that if $\vdash \mathsf{H}(\mathsf{H}X)$ holds for every term X, then it must hold for $X \equiv \mathsf{YH}$, where Y is a fixed point combinator, and from this follows $\vdash \mathsf{YH}$, which he thought was counterintuitive. For this reason, Curry gave up on trying to find a system with axioms equivalent to the deduction theorem, and, in his [1973], defined a system he called $\mathcal{F}_{22}$, which was the system $\mathcal{F}_{21}^*$ of [Curry *et al.*, 1972, §15C] without the postulate $\vdash \mathsf{H}(\mathsf{H}X)$, and proved it consistent in the weak sense that every theorem is canonical. Seldin obtained stronger consistency results for this system in his [1975b; 1977] by proving cut elimination, thus showing that there are canonical terms which are not provable. This system is defined as follows:

Definition 1 The *system* $\mathcal{F}_{22}$ is based on an applied $\lambda\beta\eta$-calculus or equivalent combinatory logic with atomic constants (that do not head redexes) Ξ, E,

H, and perhaps additional canonical atoms (i.e., atomic predicate functions) of degree (number of arguments) n. The term F is defined by

$$\mathsf{F} \equiv \lambda xyz \,.\, \Xi x(\lambda u \,.\, y(zu)).$$

The axioms are $\mathsf{E}X$ for all terms X and also FEHE. The rules are as follows:

(Eq) $$\frac{X \quad X =_* Y}{Y}$$

(ΞE) $$\frac{\Xi XY \quad XU}{YU}$$

(ΞI) $$\frac{\begin{matrix}[Xx] \\ Yx\end{matrix} \quad \mathsf{FEH}X}{\Xi XY}$$

Condition: x does not occur free in X, Y, or in any undischarged assumption.

(H) $$\frac{X}{\mathsf{H}X}$$

($\mathsf{H}\theta$) $$\frac{\mathsf{E}X_1 \quad \mathsf{E}X_2 \quad \ldots \quad \mathsf{E}X_n}{\mathsf{H}(\theta X_1 X_2 \ldots X_n)}$$

Condition: θ is a canonical atom of degree n.

($\mathsf{H}\Xi$) $$\frac{\mathsf{FEH}X \quad \mathsf{FEH}Y}{\mathsf{H}(\Xi XY)}$$

This was the last illative system about which Curry published any results. However, he did publish several more papers on pure combinatory logic. In his [1975b], Curry proved for a generalized reduction of the kind used in [Curry *et al.*, 1972, §13A] that if a term has a normal form, then it reduces to that normal form.[45] In his [1975a], he showed how to represent Markov algorithms in combinatory logic. His [1979] is on the representation of terms in normal form in the $\lambda\beta$-calculus. And his [1980], Curry's last published

[45]The generalized reduction was supposed to be that for what Curry called a C-system, which was supposed to be a general kind of system which could be either a system of λ-calculus or else a system of combinatory logic.

paper, is a brief survey of combinatory logic followed by some responses to some philosophical criticisms of it.

Martin Bunder has done more than anybody else to carry on Curry's tradition of illative combinatory logic. His most recent results are joint work with Wil Dekkers and Henk Barendregt [Barendregt *et al.*, 1993; Dekkers *et al.*, 1998a; Dekkers *et al.*, 1998b], and include the consistency proof of a stronger system than any system previously proved consistent.

3 Curry's Work on Proof Theory

When Curry first read [Gentzen, 1934], he immediately realized its importance. When he formulated his new program after learning of the paradox of [Kleene and Rosser, 1935], he realized that Gentzen's techniques might be useful in obtaining the consistency proofs that his new program called for.

As a result, he started writing a paper, of which [Curry, 1937] is the abstract, on the subject. However, this paper was never finished,[46] and all he published on this subject before World War II was [Curry, 1939a].

After the war, Curry was invited to give a series of lectures at Notre Dame University April 12–15, 1948, and he chose as his subject an exposition of Gentzen's work. Instead of revising his earlier paper, he started over, and the result was ultimately published as [Curry, 1950].

In this work, Curry started from the point of view of what we might call a *Curry semantics*. Since Curry has a reputation for not having paid attention to semantical considerations, some may find this notion surprising, but Curry did, in fact, have definite ideas about the meaning of the logical connectives and quantifiers, and if they have been misunderstood it is because they are not based as model theory is on set theory. Curry's idea behind his exposition was that he was formalizing a part of the metatheory of an elementary formal system, and he justified the properties of the logical connectives and quantifiers on that basis.

Curry defined a *formal system* as follows:

Definition 2 A *formal system*[47] is a set[48] E of statements, called *elementary*

[46]Only the first few sections were ever typed; the rest exists only as handwritten notes.

[47]This is a late form of Curry's definition, taken from [Curry, 1963]. However, the ideas behind this definition date from the beginning of his career. For an exposition of Curry's notion of formal system, see [Seldin, 1975a].

[48]Here the word "set" is taken to mean a conceptual class, and its use does not presup-

statements, together with a subset $T \subseteq E$ of *theorems*, where the two sets satisfy the following conditions:

1. The set T is *inductive*, that is, it is generated by a *definite*[49] set of *axioms* by a set of *rules* in such a way that there is an effective procedure which can decide, given a sequence of elementary statements, whether or not that sequence constitutes a valid proof.

2. The set E is formed from a definite set O of *formal objects* and a definite set of *elementary predicates*, each with a fixed number of arguments. A statement is in E if and only if it asserts that an elementary predicate of n arguments applies to an n-tuple of formal objects.

Formal systems are classified according to the way the set O of formal objects is formed.

1. If the formal objects are defined from primitive objects, called *atoms*, and *primitive operations* in such a way that each formal object has a unique construction, then the system is called an *ob-system*, and the formal objects are called *obs*.[50]

2. If the formal objects are words (strings of characters) on some alphabet,[51] then the system is called a *syntactical system*.[52]

An *elementary formal system* is a formal system in which all the rules have the form

$$\frac{E_1 \quad E_2 \quad \ldots \quad E_n}{E_0},$$

pose any formal theory of sets.

[49]Curry called a set *definite* if there is an effective procedure for deciding whether or not a given object of a suitable universe is a member of the set.

[50]This is a word that Curry coined for this purpose about 1950. In his earliest papers, he called them *entities*, but when a philosopher told him that the word "entity" implied a meaning that he did not intend, he switched to the word *term*. However, the word "term" has another use in connection with a system of logic with quantifiers, and Curry felt in need of another word. In this paper, I have used the word "term" rather than "ob" in discussing λ-calculus and combinatory logic because the former is more common.

[51]In general, such words do not have unique constructions. For example, the word 'abc' can be constructed by adding 'a' to the front of 'bc' or by adding 'c' to the end of 'ab'.

[52]Originally, Curry considered only ob systems as formal systems. However, starting in his [1963], Curry included syntactical systems as formal systems.

where $E_0, E_1, E_2, \ldots, E_n$ are schemes for elementary statements; i.e., are expressions that become elementary formal systems when formal objects are substituted for (meta-)variables.

Ordinary formal logical calculi, such as propositional calculus and first-order predicate calculus, are both syntactical systems and ob systems, since the formal objects are defined to be words on an alphabet, but they are also *well-formed formulas*, and well-formed formulas defined in the usual way each have a unique construction from the atomic formulas by means of the connectives and quantifiers.

For Curry, the elementary statements of logical calculi are defined by applying the predicate 'is provable' to well-formed formulas. Curry indicated this by writing the sign '$\vdash$' in front of each well-formed formula. Thus, whereas it is usual in logic to take a proof as a sequence of well-formed formulas $A_1, A_2, \ldots, A_n$, Curry took a proof as a sequence of elementary statements of the form $\vdash A_1, \vdash A_2, \ldots \vdash A_n$, and he wrote

$$P_1, P_2, \ldots, P_m \ \vdash \ Q$$

as an abbreviation for the statement: if $\vdash P_1, \vdash P_2, \ldots, \vdash P_m$ are added to the system as new axioms, then $\vdash Q$ is provable.[53]

Among logical calculi, formal systems of what is usually called a *Hilbert-style formulation*, in which the only rules are modus ponens and universal generalization or modus ponens alone, are elementary formal systems. Natural deduction systems, in which there are rules which discharge assumptions, are not elementary formal systems.

Now Curry had originally defined combinatory logic in his [1929] as an ob system in which the combinatory terms are the obs and the elementary predicate is a predicate of two arguments which he denoted by '=', so that the elementary statements all had the form $X = Y$,[54] where X and Y are combinatory terms. However, starting in his [1930], Curry defined combinatory logic to be what he later called an *assertional system*, a system in which there is only one predicate, namely $\vdash$. Thus, the elementary statements all asserted the provability of combinatory terms, and $X = Y$ became an abbreviation for $\mathsf{Q}XY$.

[53]If $m = 0$, so that there are no new axioms added, this just asserts that $\vdash Q$ is provable in the existing system.

[54]There, $X = Y$ had the meaning now denoted by $X =_* Y$.

Curry considered the logical calculi he introduced in his [1950] to be formalizations of a part of the metatheory[55] of an elementary formal system. He thus defined the following metatheoretic connectives:

& : $E_1 \& E_2$ means that there is a proof of E_1 and a proof of E_2,

or: E_1 or E_2 means that there is a proof of E_1 or a proof of E_2 and there is an effectively computable process for deciding which,

$\rightarrow$: $E_1 \rightarrow E_2$ means that there is an effective process for converting any proof of E_1 into a proof of E_2, and is regarded as vacuously true if it can be constructively proved that there is no proof of E_1,

$\rightleftarrows$: $E_1 \rightleftarrows E_2$ means $(E_1 \rightarrow E_2)$ & $(E_2 \rightarrow E_1)$,

$\Rightarrow$: $E_1 \Rightarrow E_2$ means that if E_1 is adjoined to the system as an axiom, then E_2 is a theorem, and

$\Leftrightarrow$: $E_1 \Leftrightarrow E_2$ means $(E_1 \Rightarrow E_2)$ & $(E_2 \Rightarrow E_1)$.

(Clearly, these connectives can connect all statements formed by starting with elementary statements and using these connectives. And it is in terms of their meaning that Curry justified the rules of the connectives $\wedge$, $\vee$, and $\supset$.)[56]

Thus, his [1950] began with a chapter on formal systems, and throughout the rest of the book the logical systems were designed to formalize the metatheory of an elementary formal system $\mathfrak{S}$. In the second chapter, he treated the propositional calculus without negation, so the connectives were $\wedge$, $\vee$, and $\supset$. The first logical systems introduced were the Gentzen-style L-systems, LA($\mathfrak{S}$) and LC($\mathfrak{S}$). The first, LA($\mathfrak{S}$), was a singular system, one with only one formula on the right:

$$\Gamma \Vdash A,$$

[55]Kleene, in a review of [Curry, 1941d] published in the *Journal of Symbolic Logic*, 6:10–102, 1941, objected to Curry's use of the prefix "meta-", claiming that this prefix should only be used in connection with a distinction of levels of language. In 1951, Curry responded to this criticism by deciding to use the prefix "meta-" only in the sense Kleene had suggested and to use the prefix "epi-" instead. See the preface of [Curry, 1951]. I think that most logicians now use the prefix "meta-" in the sense in which Curry had used it prior to 1951, and so I will continue to use that prefix instead of "epi-" in this article.

[56]This reading of the connectives is close to the Brouwer-Heyting-Kolmogorov interpretation of them. See [Troelstra and van Dalen, 1988, p. 31].

where Γ is a sequence of formulas and A is a formula. The second, LC($\mathfrak{S}$), allowed sequences with more than one formula on the right:

$$\Gamma \Vdash \Delta,$$

where Γ and Δ are sequences of formulas. Both kinds of systems had the usual axiom $A \Vdash A$, and both also had as axioms $\Vdash E$, where E was an axiom of $\mathfrak{S}$. Both had the usual structural rules of permutation, weakening, and thinning, although LA had them only on the left while LC had them on the right as well, and both had the usual rules for $\wedge$, $\vee$, and $\supset$ on both the left and right. Both also had the following rule:

$$\vdash * \qquad \frac{\Gamma \Vdash E_1, \Theta \quad \Gamma \Vdash E_2, \Theta \quad \ldots \quad \Gamma \Vdash E_n, \Theta}{\Gamma \Vdash E_0, \Theta} \qquad \begin{array}{l}\textit{Condition:}\\ E_1, E_2, \ldots, E_n \vdash E_0 \\ \text{in } \mathfrak{S}.\end{array}$$

(In LA($\mathfrak{S}$), Θ was the void sequence.[57]) Curry did not postulate the rule Cut as part of either system, but proved as a metatheorem that if the premises of Cut were derivable, then so was the conclusion. He called this result the Elimination Theorem.

Curry then turned to natural deduction formulations. Since he wanted to reserve the letter 'N' for negation, he used the second consonant of the word "natural" and called the systems TA($\mathfrak{S}$) and TC($\mathfrak{S}$). The only axioms in either system were the axioms of $\mathfrak{S}$. The rules of TA($\mathfrak{S}$) were the standard introduction and elimination rules for $\wedge$, $\vee$, and $\supset$. TC($\mathfrak{S}$) was obtained from TA($\mathfrak{S}$) by adjoining one rule:

$$\supset\mathrm{K} \qquad \frac{\begin{array}{c}[A \supset B]\\ A\end{array}}{A.}$$

Curry also considered Hilbert-style formulations of both systems, HA and HC. These were formulations in which the only rule was modus ponens, or $\supset E$, and the other rules of TA and TC were replaced by axiom schemes.

[57]If $n = 0$, this means that if $\vdash E_0$ is an axiom of $\mathfrak{S}$, then $\Vdash E_0$ is an axiom of the L-system.

In the next chapter, Curry considered adding first-order quantification to obtain LA*($\mathfrak{S}$) and LC*($\mathfrak{S}$). The L-rules on the left and right for the quantifiers were standard. The T-rules and H-axioms were also standard.

In the next chapter, Curry took up negation. He defined two kinds of negation:

1. *Absurdity.* An elementary statement is *absurd* if, when it is adjoined to the system as an axiom, all elementary statements become provable.

2. *Refutability.* An elementary statement is *refutable* if, when it is adjoined to the system as an axiom, any one of a given set of *counteraxioms* becomes provable.

These two kinds of negation led to different systems:

1. *Minimal Logic*, LM, which was LA plus refutability.

2. *Intuitionistic Logic*, which was LA plus absurdity.

3. *Strict Refutability*, LD, which was LA plus excluded middle.

4. *Classical Logic*, LK, which was LC plus absurdity, or alternatively LJ plus excluded middle.

These were the systems of propositional logic; there were corresponding systems with quantifiers.

For technical reasons, Curry was unable to use the definition

$$\neg A \equiv A \supset \bot,$$

where $\bot$[58] stands for a fixed false proposition, in the L-formulation, so in this formulation, he took $\neg$ as a primitive connective. The L-rules for negation on the left and right, $*\neg$ and $\neg *$, were the traditional ones. For refutability, he also postulated

$\bot *$ $\qquad$ $$\frac{\Gamma \Vdash \bot_i, \Theta}{\Gamma \Vdash \Theta}$$ *Condition:* $\bot_i$ is a counteraxiom.

[58]Curry wrote 'F' instead of '$\bot$'.

For absurdity, he postulated instead[59]

$$\bot\text{J} \qquad \frac{\Gamma \Vdash \Theta}{\Gamma \Vdash A, \Theta.}$$

To get the law of the excluded middle, he postulated

$$\neg\text{X} \qquad \frac{\Gamma, \neg A \Vdash A, \Theta}{\Gamma \Vdash A, \Theta.}$$

In these rules, Θ was void in a singular system. Note that this meant that in systems with negation, it was possible in either singular or multiple systems to have a void right-hand side.

Then he defined the following propositional systems:

LM = LA + $*\neg$ + $\neg*$ + $\bot*$,

LJ = LM + $\bot$J,[60]

LD = LM + $\neg$X, and

LK = LC + $*\neg$ + $\neg*$.

Note that LM, LJ, and LD were all singular. Curry was dealing with quantified versions of all these sytems, and he proved the Elimination Theorem for all of them.

For the natural deduction systems, he used the definition $\neg A \equiv A \supset \bot$, where $\bot$ is a new atomic formula added to the system.[61] The standard introduction and elimination rules for $\neg$, $\neg$I and $\neg$E, then follow from the definition of $\neg A$. The other T-rules used in connection with negation were

[59]There were no counteraxioms in any system in which $\bot$J was postulated, so rule $\bot*$ was never postulated in such systems.

[60]But see the preceding footnote.

[61]In some systems, $\bot$ can be identified with a formula that is already in the system.

the following:

⊥I	$\dfrac{\bot_i}{\bot}$	*Condition:* $\bot_i$ is a counteraxiom.
¬J	$\dfrac{\bot}{A.}$	
¬D	$\dfrac{\begin{matrix}[\neg A]\\ A\end{matrix}}{A.}$	
¬K	$\dfrac{\neg\neg A}{A.}$	

Note that the rule ¬K is equivalent to

$$\dfrac{\begin{matrix}[\neg A]\\ \bot\end{matrix}}{A,}$$

which is the form used by Prawitz in his [1965]. The T-systems for negation are then defined as follows:

TM = TA + ¬E + ¬I + ⊥I,

TJ = TM + ¬J,[62]

TD = TM + ¬D, and

TK = TM + ¬J + ¬D = TM + ¬K.

Curry also dealt with Hilbert-style systems.

The last chapter was devoted to modal logic, with operators for necessity and possibility. Curry was unable to prove the Elimination Theorem for the modal systems.

In the preface, Curry discussed the algebraic approach to logic, saying that although he thought it was important, he had to leave it out of the lectures on which [Curry, 1950] was based, for lack of space. In fact, the

[62]There are no counteraxioms for TJ or for any system in which ¬J is postulated, so ⊥I is vacuous for TJ and TK.

book did not include answers to some questions left open in the lectures which he had settled after he delivered the lectures, since he wanted the book to represent the lectures themselves. But he did write up this work for publication:

1. In his [1952f], Curry proved some results about LD. He also suggested a singular version of LC and LK, namely LC_1 and LK_1. The former was to be defined by adding to LA the rule

   $\supset$C $$\frac{\Gamma, A \supset B \Vdash A}{\Gamma \Vdash A.}$$

   For LK_1, the rules $\perp$J and $\neg$X were postulated.

2. In his [1952d], Curry showed how to use the definition $\neg A \equiv A \supset \perp$ with the L-systems. When this was done there were no void right-hand sides, and rules $\perp *$, $\perp$J, and $\neg$X were replaced by, respectively,

   $\perp *$ $$\frac{\Gamma \Vdash \perp_i, \Theta}{\Gamma \Vdash \perp, \Theta}$$ *Condition:* $\perp_i$ is a counteraxiom.

   $\perp$J $$\frac{\Gamma \Vdash \perp, \Theta}{\Gamma \Vdash A, \Theta.}$$

   $\perp$X $$\frac{\Gamma, A \supset \perp \Vdash A}{\Gamma \Vdash A.}$$

   where, as usual, Θ was void in a singular system.

3. In his [1952e], Curry showed that under certain conditions, the order in which rules are applied can be reversed in classical L-systems.

4. In his [1952a], Curry proved the Elimination Theorem for systems with a necessity operator.

In his review of the first two of the above papers, Bernays [1953] suggested adding $\supset$C to LM without adding $\perp$J. This would be a system of classical refutability, which Curry later called LE. Some properties of this system were studied based on its H-formulation in [Kanger, 1955], and an extensive study starting from the L-formulation was made in [Kripke, 1958]. Although Curry was busy writing [Curry and Feys, 1958] during most of this period and was therefore unable to pursue his work on general proof theory, he kept up with these results.

Once [Curry and Feys, 1958] was finished, Curry decided that since he expected to make extensive use of Gentzen techniques to obtain consistency proofs in the projected second volume, he should put his work on those techniques in better order than it was. This work led to [Curry, 1963]. In this work, after a short introductory chapter, there were chapters on formal systems (Chapter 2) and epitheory (his word for metatheory, Chapter 3), which were largely rewritten from [Curry and Feys, 1958]. Then, in Chapter 4, he included the material on algebraic logic that he had lacked room for in his [1950]; he had been able to give a course on this subject during his year at Louvain, and the course notes had been published as [Curry, 1952b], and [Curry, 1963, Chapter 4] was expanded from this.

The real work on logical calculi begins in Chapter 5. Unlike [Curry, 1950], Curry started here with Hilbert-style and natural deduction systems. Chapter 5 was on propositional logic without negation, and included LA and LC. But it also included LC$_1$, and, to this, Curry added a multiple version of LA, LA$_m$, which was like LC except that the rule for implication on the right must be singular. The chapter closed with a system for tableaux, inspired by Beth's proof tableaux.

Chapter 6 is on negation, and here Curry added the E-systems, HE, TE, and LE to the systems he had previously. TE was defined by adding the introduction and elimination rules for $\neg$ to TC, and this meant that TK could be obtained by adding $\neg$J to TE. LE was defined from LC the way LM was defined from LA, by adding the rules for negation on the left and right and also adding $\bot *$. For LD, Curry replaced rule $\neg$X by a new one:

$$\frac{\Gamma, \neg A \Vdash \Delta \quad \Gamma, A \Vdash \Delta}{\Gamma \Vdash \Delta.}$$

He also had both singular and multiple formulations for all systems. Another innovation was three formulations of negation: 1) the F-formulation, where $\bot$ was postulated and $\neg A$ was defined to be $A \supset \bot$, 2) the N-formulation, in which $\bot$ was not postulated and $\neg$ was primitive, and the FN-formulation, in which both $\bot$ and $\neg$ were primitive and the equivalence of $\neg A$ and $A \supset \bot$ was a consequence of the logical rules.

Chapter 7 was on quantification, and while not much was new here, the chapter was quite complete. The tableau system of Chapter 5 was extended to all quantified systems.

Chapter 8 was on modal systems.

In his [1965], Curry gave an algorithm for converting the linear form of natural deduction rules as used in [Jaśkowski, 1934; Fitch, 1952] to the form based on trees used by Gentzen and Curry himself, discussed some improvements to his semantic justification of the L-rules, and introduced a form of dialogue or game logic based on the ideas of [Lorenz, 1961; Lorenzen, 1961; Stegmüller, 1964][63] equivalent to LJ^*.

In his [1968a], he showed how to associate formulas with sequents in such a way that for an L-system satisfying certain conditions, if one sequent can be derived from some others by the rules of the system, then a sequent can be proved in which the right-hand side is the formula corresponding to the concluding sequent and the formulas on the left are those corresponding to the premise sequents.

This was Curry's last paper on proof theory proper. However, since his work on proof theory was based in many ways on his idea of formal system, it is worth mentioning that in his [1968b], he proved that his notion of formal system was essentially equivalent to that of Smullyan [1961].

4 Curry's Philosophy of Mathematics

As part of his earliest work on combinatory logic, Curry developed a philosophy of mathematics that he was to call *formalism*. Although he had already formulated this philosophy in a preliminary way by 1929[64] the first exposition of it came with his [1939b], and it is best known from his [1951], most of which was written in 1939. To this day, this is all many philosophers of mathematics know of Curry's philosophy, even though both of these works date from 1939, early in his career.[65]

For Curry, the basic idea of *formalism* was that mathematics does not have a subject matter that is independent of our knowledge and comes from outside the subject. This is different from the views of both platonists and intuitionists, who believe that mathematics does have a subject matter external to the subject: platonists believe that there are such things as numbers, points, lines, etc. which exist independently of our knowledge of them, and

[63]The first two of these works were later reprinted in [Lorenzen and Lorenz, 1978].

[64]See my paper [1980, page 11].

[65]See, for example, [Shapiro, 2000, pages 168–170].

intuitionists believe that the subject matter of mathematics consists of mental constructions which are based on a special kind of primordial intuition.[66] For Curry, mathematics is like language, in the sense that it is the result of human activity, and in the terms used by Karl Popper [1968][67] it is part of the "third world." Thus, for Curry, no mathematical objects existed before human beings started doing mathematics.

In 1939, Curry proposed to define mathematics as the *science of formal systems*.[68] Many logicians think in terms of formal systems whose formal objects represent propositions. But other kinds of formal systems are possible. For example, consider the system $\mathcal{N}$:[69] there is one atom, 0, and one primitive operation, denoted by adding | on the right, so obs are represented by adding strings of the symbol | to the right of 0. Thus, the obs all have the form

$$0\underbrace{||\ldots|}_{n},$$

which represents the natural number n. There is one elementary predicate, which takes two arguments, so an elementary statement is written

$$X = Y$$

There is one axiom, namely

$$0 = 0,$$

and there is one rule:

$$X = Y \quad \Rightarrow \quad X| = Y|.$$

It is easy to see that all the obs represent natural numbers, and all the theorems have the form

$$X = X.$$

All of arithmetic and analysis can be constructed as part of the metatheory of this formal system $\mathcal{N}$, and by using the techniques of analytic geometry,

[66]See [Curry, 1963, page 9].

[67]This paper was delivered under the title "Epistemology and Scientific Knowledge" at a session of the Third International Congress for Logic, Methodology and Philosophy of Science in Amsterdam on Friday, August 25, 1967 at a session with Curry in the chair and at which I was present. After Popper finished speaking, Curry told me privately that he thought Popper had made a big deal out of something that was trivially true.

[68]See Definition 2 above.

[69]This is the system $\mathcal{N}_1$ of [Seldin, 1975a, page 456], which is the same as the system of [Curry, 1963, page 256].

it is also possible to make geometry part of this metatheory. This method of reconstructing mathematics in terms of formal systems seems different from the way most logicians and philosophers of mathematics think of using formal systems.

The definition of mathematics as the science of formal systems led to the criticism that he could not account for the existence of mathematics before formal systems first appeared at the end of the nineteenth century.[70] By the early 1960s, Curry proposed defining mathematics as the *science of formal methods*.[71] That this definition is broader than the earlier one can be seen by the fact that in elementary arithmetic, the operation of recognizing what a pile of five apples has in common with a pile of five oranges is a matter of abstracting from the kind of fruit one has in the pile and noting that what they have in common is their cardinality. Doing this is, in a sense, ignoring the *content* of the piles and noting only their *form*, which is a kind of formal activity.

To further clarify this idea, consider what Curry has to say in his [1963, page12] about natural numbers:

> A formalist would not speak of "the natural numbers" but of a set or system of natural numbers. Any system of objects, no matter what, which is generated from a certain initial object by a certain unary operation in such a way that each newly generated object is distinct from all those previously formed and that the process can be continued indefinitely, will do as a set of natural numbers. He may, and usually does, objectify this process by representing the numbers in terms of symbols; he chooses some symbol, let us say a vertical stroke '|', for the initial object, and regards the operation as the affixing of another '|' to the right of the given expression. But he realizes there are other interpretations; in particular, if one accepts the platonist or intuitionist metaphysics, their systems will do perfectly well.

Note that this "system of objects" need not be given by a formal system. A natural number is any object of a system of this kind.

In my opinion, this means that Curry's formalism, at least the formalism of his mature years, is a form of *structuralism*. To see this, note what Stewart Shapiro says in his [2000, page 258]:

[70]See, for example, [Shapiro, 2000, page 170], but this criticism was made much earlier.
[71]See [Curry, 1963, page 14].

> The structuralist vigorously rejects any sort of ontological independence among the natural numbers. The essence of a natural number is its *relations* to other natural numbers. The subject-matter of arithmetic is a single abstract structure, the pattern common to any infinite collection of objects that has a successor relation, a unique initial object, and satisfies the induction principle. The number 2 is no more and no less than the second position in the natural number structure; and 6 is the sixth position. Neither of them has any independence from the structure in which they are positions, and as positions in this structure, neither number is independent of the other. (Emphasis in original.)

What Curry would have said about this is that extracting the pattern to which Shapiro refers is a matter of abstracting the form from the structure and disregarding the content. And if a structuralist does not assume that these structures exists independently of our knowledge of them, Curry would have regarded that structuralist as his kind of formalist.

Appendix

A Pure λ-Calculus

In our discussion of functions at the beginning of this article, we mentioned two operations:

1. The *application* of a function to its argument, as in $f(3)$.
2. The *abstraction* operation that defines a function from an expression for its values, as in $\lambda x \,.\, x^2$.

These two operations are the basis of the λ-*calculus*, which is a formal system for functions interpreted as rules of calculation. The λ-calculus is the legacy of Alonzo Church.

The λ-calculus differs from ordinary mathematical practice in two essential ways:

1. Application is taken as a binary operation. At first glance, this may appear to leave out functions of more than one argument. But this is not so. There is a way to reduce functions of more than one argument to functions of one argument, a way called *currying* after Haskell B. Curry. Consider the function $f(x, y) = x - y$ of two arguments. If we replace x by some constant, say 4, we get $g(y) = f(4, y) = 4 - y$, a function of one argument. This function can be taken as the value of a function f^* related to f, so that $f^*(4) = g(y) = 4 - y$, and then $f^*(4)(y) = 4 - y$, and $f^*(4)(3) = 4 - 3 = 1$. This function f^* is called the *curried version of* f, or $\mathsf{curry}\, f$. Thus, $(\mathsf{curry}\, f)\, 4\, 3 = f(4, 3)$, and $(\mathsf{curry} f)\, x\, y = f(x, y)$.
2. In ordinary mathematical discourse, each function has a domain, and a function can only be applied to arguments from that domain. In pure λ-calculus, every term is considered to be in the domain of every other term, and every term is a function whose domain includes every other term. At first glance, this may seem counter-intuitive, and it does make models of λ-calculus very complicated. But it does correspond, in a sense, to what goes on in a computer, where the program instructions are stored in memory as is the data to which the program is applied,

and all items stored in memory are simply strings of 0s and 1s.[72] In principle, there is no reason why a computer could not be told to take any such string as the program and any other such string as the data; if the first string is not in fact an executable set of instructions, the computer will not do anything useful, but only the human user will know that. The pure λ-calculus is simpler for not trying to define a domain for each function, and so the applicability of any term to any other term is not considered a problem.[73]

This has to do with the formation of *terms*. But any formal system for functions as rules must deal with the computations involved in calculating the values of functions. In the discussion of the squaring function at the beginning of this article, we noted that to evaluate the function at an argument, we always start by substituting the argument for the variable in the expression for the function. Thus, we had that

$$(\lambda x \, . \, x^2)3 = 3^2.$$

In general, we would have

$$(\beta) \qquad (\lambda x \, . \, M)N = [N/x]M,$$

where $[N/x]M$ represents the result of substituting N for x in M. This is, in fact, the only means the λ-calculus has for carrying out calculations. It is perhaps surprising that it is enough to carry out all recursive operations.

Now the substitution mentioned in the last paragraph is not simple replacement, for we are dealing with *bound variables*. A bound variable is one for which no substitution is possible, and it serves only as a place-holder. Furthermore, along with computation rule (β), there is a rule which asserts that bound variables can always be changed:

(α) $(\lambda x \, . \, M) = (\lambda y \, . \, [y/x]M)$, if y does not occur free in M.

Finally, let us note an additional rule of computation that is often postulated. Note that if x does not occur free in M, and if N is any term, then

$$(\lambda x \, . \, Mx)N = MN.$$

[72]Or, more precisely, pulses of high voltage and low voltage which are usually interpreted as 0s and 1s.

[73]Domains of λ-terms can be specified by using *types*. This is discussed in Appendix B.

This means that $(\lambda x \,.\, Mx)$ has the same effect on an argument N as M. So some systems of λ-calculus include the rule

(η) $(\lambda x \,.\, Mx) = M$, if x does not occur free in M.

Without this last rule, the system is called the $\lambda\beta$-calculus; with it, the calculus is called the $\lambda\beta\eta$-calculus or the $\lambda\eta$-calculus.

So far, the rules of calculation have been stated as equations. But in many actual calculation these rules are all read one way, from left to right. One calculation step that proceeds this way is called a *contraction*, and a sequence of such steps is called a *reduction*.

With these ideas, the pure λ-calculus is defined as follows:

Definition 3 The formal system Λ of λ-*calculus* is defined as follows:

1. The *terms* are defined by:

   (a) *Variables*. There are assumed to be countably many, and it is assumed that they come in an enumerated sequence. Here, variables will be denoted by lower-case Italic letters from the end of the alphabet, x, y, z, x_1, etc.

   (b) *Constants*. There may not be any, or there may be several, depending on the variant of system.

   (c) *Applications*. If M and N are terms, then so is (MN). Parentheses are omitted by association to the left, so that $MN_1N_2 \ldots N_n$ is $(\ldots((MN_1)N_2)\ldots N_n)$.

   (d) *Abstractions*. If x is a variable and M is a term, then so is $(\lambda x \,.\, M)$. Repeated abstraction $(\lambda x_1 \,.\, (\lambda x_2 \,.\, \ldots (\lambda x_n \,.\, M)\ldots))$ will be written $(\lambda x_1 x_2 \ldots x_n \,.\, M)$.

2. An occurrence of a variable x in a term M is said to be *bound* if it occurs in a subterm of M of the form $(\lambda x \,.\, N)$. An occurrence of a variable which is not bound is said to be *free*. That a variable x has a free occurrence in a term M is written $x \in \mathrm{FV}(M)$.

3. The *substitution* of a term N for a variable x in a term M, denoted by $[N/x]M$, is defined by induction on the structure of M as follows:

   (a) $[N/x]x$ is N.

(b) If a is any variable different from x or is a constant, then $[N/x]a$ is a.

(c) $[N/x](M_1M_2)$ is $([N/x]M_1[N/x]M_2)$.

(d) $[N/x](\lambda x\ .\ M)$ is $(\lambda x\ .\ M)$.

(e) If y is different from x, and if either $y \notin \mathrm{FV}(N)$ or else $x \notin \mathrm{FV}(M)$, then $[N/x](\lambda y\ .\ M)$ is $(\lambda y\ .\ [N/x]M)$.

(f) If $y \in \mathrm{FV}(N)$ and $x \in \mathrm{FV}(M)$, then $[N/x](\lambda y\ .\ M)$ is defined to be $(\lambda z\ .\ [z/y][N/x]M)$, where z is the first variable in the given sequence of variables which is not in $\mathrm{FV}(MN)$.

(There are many other notations for $[N/x]M$, including $M[N/x]$ and $M[x := N]$.)

4. There are two kinds of *reduction*. That M reduces to N will be written $M \rhd N$.

   (a) The first kind, known as *β-reduction*, is defined by the following axioms and rules:

   (α) $(\lambda x\ .\ M) \rhd (\lambda y\ .\ [y/x]M)$, where $y \notin \mathrm{FV}(M)$.

   (β) $((\lambda x\ .\ M)N) \rhd [N/x]M$.

   (ρ) $M \rhd M$.

   (μ) If $M \rhd N$, then $PM \rhd PN$.

   (ν) If $M \rhd N$, then $MP \rhd NP$.

   (ξ) If $M \rhd N$, then $(\lambda x\ .\ M) \rhd (\lambda x\ .\ N)$.

   (τ) If $M \rhd N$ and $N \rhd P$, then $M \rhd P$.

   This is equivalent to defining β-reductions as involving two kinds of replacements:

   (α) If $y \notin \mathrm{FV}(M)$, a replacement of $(\lambda x\ .\ M)$ by $(\lambda y\ .\ [y/x]M)$.

   (β) $((\lambda x\ .\ M)N)$ by $[N/x]M$. Here $((\lambda x\ .\ M)N)$ is called a *β-redex*, and $[N/x]M$ is called its *contractum*.

   Each of these kinds of replacements is called a *contraction*.

   The notation $M \rhd_\beta N$ will mean that M β-reduces to N.

   (b) The second kind of reduction, *$\beta\eta$-reduction*, is defined by adding to the definition of β-reduction the axiom

(η) If $x \notin \mathrm{FV}(M)$, then $(\lambda x \,.\, Mx) \rhd M$.

This is equivalent to adding the replacement

(η) $(\lambda x \,.\, Mx)$ by M, provided that $x \notin \mathrm{FV}(M)$. Here $(\lambda x \,.\, Mx)$ is called an *η-redex* and M is its *contractum*.

This kind of replacement is called an *η-contraction*.

The notation $M \rhd_{\beta\eta} N$ means that M $\beta\eta$-reduces to N.

5. *Conversion*. The relation $M =_{*} N$ (M *converts* to N), is defined by changing $\rhd$ to $=_{*}$ in the definition of reduction and adding the rule

   (σ) If $M =_{*} N$ then $N =_{*} M$.

   Conversion can also be defined as a sequence of contractions and reverse contractions. There are two kinds of conversion: β-conversion, $=_{\beta}$, and $\beta\eta$-conversion, $=_{\beta\eta}$.

Remark 1 For $=_{\beta\eta}$, we have the following derived rule:

(ζ) If $Mx =_{\beta\eta} Nx$ and if $x \notin \mathrm{FV}(MN)$, then $M =_{\beta\eta} N$.

This rule is also called (Ext), (for *extensionality*).

Remark 2 Terms which cannot be reduced (i.e., terms in which there are no occurrences of redexes), are said to be in *normal form*.

Remark 3 In general, the symbol $\equiv$ will be used to indicate identical terms. In λ-calculus, however, it will mean terms that are (α)-equivalent (i.e., that differ only in their bound variables). This is because it is common in λ-calculus to identify terms that differ only in their bound variables, since they mean the same thing.[74]

Remark 4 There is a fixed-point operator Y, which, when applied to any term, calculates a fixed-point of that term. In fact, there is more than one. One example is

$$\mathsf{Y} \equiv (\lambda xz \,.\, x(zzx))(\lambda xz \,.\, x(zzx)).$$

It is easy to show that $\mathsf{Y}x \rhd x(\mathsf{Y}x)$.

[74]But sometimes ignoring the difference between distinct (α)-convertible terms can lead to technical difficulties.

Remark 5 Natural numbers and numerical functions can be coded in Λ, by a method originally due to Alonzo Church, as follows:

- $\mathbf{0}$ is defined to be $\lambda xy\,.\,x$
- $\mathbf{1}$ is defined to be $\lambda xy\,.\,xy$.
- $\mathbf{2}$ is defined to be $\lambda xy\,.\,x(xy)$.
- $\vdots$
- $\mathbf{n}$ is defined to be $\lambda xy\,.\,\underbrace{x(\ldots(x(x}_{n}\,y))\ldots)$.

Numerical functions can be represented as terms of Λ. A numerical function can be represented as a term in Λ if and only if it is recursive.

Remark 6 Church did not define $\mathbf{0}$ in his system but started with $\mathbf{1}$ because he wanted $\lambda x\,.\,M$ to be well-formed only if there is a free occurrence of x in M, and so in his system $\mathbf{0}$ is not well-formed. This form of λ-calculus is called λI-calculus. See § 1.2 above. The system Λ defined here is often called λK-calculus when it needs to be distinguished from the λI-calculus.

Since reduction represents the process of calculation, it is important to have confidence that the order in which calculation steps are performed does not make a difference. That it does not is a result of the *Church-Rosser Theorem*:

Theorem 5 (Church-Rosser Theorem) *If $M \rhd P$ and $M \rhd Q$, then there is a term N such that $P \rhd N$ and $Q \rhd N$.*

The theorem holds for both β-reduction and $\beta\eta$-reduction.

Corollary 1 *If $M =_* N$, then there is a term P such that $M \rhd P$ and $N \rhd P$.*

Here, if $=_*$ is $=_\beta$, then $\rhd$ is $\rhd_\beta$, whereas if $=_*$ is $=_{\beta\eta}$, then $\rhd$ is $\rhd_{\beta\eta}$.

For more details, see [Hindley and Seldin, 1986] and the references given there.

B Lambda-Calculus with Types

In pure λ-calculus, every term is in the domain of every other term. If it is necessary to distinguish domains of function, we use *types*. There are now many versions of type assignment to λ-terms, but the original and most basic one, which is the one we consider here, uses the simplest definition of types.

Definition 4 *Types* are defined from *atomic types* $\theta_1, \theta_2, \ldots, \theta_n, \ldots$ (possibly infinitely many) by induction:

1. Every atomic type is a type.
2. If α and β are types, then so is $(\alpha \to \beta)$.

Remark 7 Other notations have been used in the past for the type $(\alpha \to \beta)$, including $(\alpha\beta)$, $(\beta\alpha)$, and Curry's notation $\mathsf{F}\alpha\beta$. The notation $(\alpha \to \beta)$ is now standard. The outermost parentheses are often omitted. Also, parentheses are omitted by association to the right, so that

$$\alpha_1 \to \alpha_2 \to \ \ldots \ \to \alpha_n \to \beta$$

is the type

$$(\alpha_1 \to (\alpha_2 \to (\ldots (\alpha_n \to \beta) \ldots))).$$

The type $(\alpha \to \beta)$ is intended to represent the type of functions which take arguments of type α and have values in β.

There are two approaches to the assignment of types to λ-terms, one due to Church and the other due to Curry.

In the Church style, every variable is assigned a unique type, and this type is indicated in λ-abstracts, so that if x has type α, the abstraction of M with respect to x is denoted $\lambda x : \alpha \,.\, M$. (An older notation, used by Church, is $\lambda x_\alpha \,.\, M_\beta$, where β is the type of M.) In Church-style typing, $\lambda x : \alpha \,.\, x$ (or, in Church's original notation $\lambda x_\alpha \,.\, x_\alpha$) is an identity function whose domain and range are the type α, and each type has its own identity function.

That a type α is assigned to a term M is denoted $M : \alpha$. If there are any atomic constants, types are assigned to them by axioms, but there may not be any atomic constants. The type assignment system, TACh_λ, is defined as follows:

Definition 5 The *system* $TACh_\lambda$ is defined as follows:

1. *Types* are defined as in Definition 4.

2. *Terms* are like the terms of Λ, except that clause 1d of Definition 3 is replaced by the following: if x is a variable, α is a type, and M is a term, then $\lambda x : \alpha \,.\, M$ is a term.

3. Typing judgements are all of the form $M : \alpha$, where M is a term and α is a type.

4. If c is an atomic (term) constant with a type γ, then

$$c : \gamma$$

is an axiom. (That this is an axiom is often written

$$\overline{c : \gamma,}$$

to indicate a step of a deduction with no premises.)

5. Types are assigned to compound terms by the following two rules:

$(\rightarrow \mathrm{E})$

$$\frac{M : \alpha \rightarrow \beta \quad N : \alpha}{MN : \beta}$$

$(\rightarrow \mathrm{I_{Ch}})$

$$\frac{\begin{array}{c}[x : \alpha] \\ M : \beta\end{array}}{\lambda x : \alpha \,.\, M : \alpha \rightarrow \beta}$$

Condition: x does not occur free in any undischarged assumption.

In Curry-style typing, the types of the variables are not fixed, but can be any types for which the rules allow a term to be assigned a type. The terms are simply the terms of Λ. In Curry-style typing, $\lambda x \,.\, x$ is an identity function on any type, so $\lambda x \,.\, x : \alpha \rightarrow \alpha$ for every type α.

The formal definition is as follows

Definition 6 The *system* $TACu_\lambda$ is defined as in Definition 5 with the following changes:

1. In Clause 2 of that definition, the *terms* are the terms of Λ.
2. In Clause 5 of that definition, rule $(\to \mathrm{I_{ch}})$ is replaced by

$$(\to \mathrm{I_{Cu}}) \qquad \frac{\begin{array}{c}[x:\alpha]\\ M:\beta\end{array}}{\lambda x\,.\,M:\alpha\to\beta}$$

Condition: x does not occur free in any undischarged assumption.

Note that in both styles of typing, the deduction of a type assignment follows the construction of the term.

These rules are for deductions from assumptions. Assumptions are usually assumed in the form of *contexts*, which assign variables to types, assigning distinct variables to distinct types. Thus, a context would have the form

$$x_1:\alpha_1, x_2:\alpha_2, \ldots, x_n:\alpha_n.$$

Here, the variables $x_1, x_2, \ldots, x_n$ are all required to be distinct (but the types $\alpha_1, \alpha_2, \ldots, \alpha_n$ need not be distinct).

Some important results about this system are the following:

Theorem 6 (Subject-Reduction Theorem) *If* $M \rhd N$ *and* $\Gamma \vdash M:\alpha$ *for any context* Γ *and any type* α*, then* $\Gamma \vdash N:\alpha$*.*

Theorem 7 (Normal Form Theorem) *If* $\Gamma \vdash M:\alpha$ *for any context* Γ *and any type* α*, then* M *has a normal form.*

C Pure Combinatory Logic

The λ-calculus has two primitive operations for constructing terms: abstraction and application. In *combinatory logic*, there is only one such operation, application. Instead of abstraction, combinatory logic has some special constants, called *basic combinators*, and from these basic combinators abstraction can be defined. Since the only primitive term-forming operation is application, there are no bound variables to worry about, and variables will simply be atomic terms that we decide to call variables and for which we are prepared to make substitutions.[75] Furthermore, the substitution of a term Y for a variable x in another term X is simply the replacement of all occurrences of x by Y.

In standard combinatory logic, the basic combinators are K and S. These two constants have reduction rules defined as follows:

(K) $\mathsf{K}XY \rhd X$,

(S) $\mathsf{S}XYZ \rhd XZ(YZ)$.

(Weak) reduction, or $\rhd_w$ is defined as replacements using these two rules, and the two kinds of redexes are of the form $\mathsf{K}XY$ and $\mathsf{S}XYZ$. Its corresponding conversion is $=_w$. It is easy to see that if $X \rhd Y$, then for any term Z, $[Z/x]X \rhd [Z/x]Y$.

Other important functions can be defined in terms of these two, for example the identity function I is defined to be SKK, since

$$\mathsf{SKK}X \rhd \mathsf{K}X(\mathsf{K}X) \rhd X.$$

Other definitions are for the composition operator B, which is defined to be $\mathsf{S}(\mathsf{KS})\mathsf{K}$ and has the property that $\mathsf{B}XYZ \rhd X(YZ)$, the commutator C, which has the property that $\mathsf{C}XYZ \rhd XZY$ and is defined to be $\mathsf{S}(\mathsf{BBS})(\mathsf{KK})$, and the duplicator W, which satisfies $\mathsf{W}XY \rhd XYY$ and is defined to be $\mathsf{SS}(\mathsf{KI})$.

It turns out that the combinator S can be defined in terms of B, C, and W as $\mathsf{B}(\mathsf{B}(\mathsf{BW})\mathsf{C})(\mathsf{BB})$; see [Curry and Feys, 1958, §5B1].

In his dissertation [1935a; 1935b], Rosser discovered that all the other combinators can be defined in terms of I and a combinator J with the reduction rule

$$\mathsf{J}UXYZ \rhd UX(UZY).$$

Now, for a variable x, an abstraction term $[x]X$ can be defined by induction on the structure of X as follows:

[75]This is essentially the same procedure used in computer languages, where variables are simply identifiers for which we are prepared to make substitutions and constants are simply identifiers for which we are not prepared to make substitutions.

- If a is an atomic term, then $[x]a$ is
    - I if a is x,
    - $\mathsf{K}a$ if a is not x.
- If X is YZ, then assuming that $[x]Y$ and $[x]Z$ are already defined, $[x]XY$ is $\mathsf{S}([x]Y)([x]Z)$.

Then it follows that $[x]X$ is a term in which the variable x does not occur. Furthermore, it can easily be proved by induction on the structure of the term X that

$$([x]X)Y \rhd [Y/x]X.$$

However, this is not the only way to define abstraction, and this definition results in longer terms than some others. For example, the following definition is often preferred:

(a) If $x \notin \mathrm{FV}(U)$, then $[x]U$ is $\mathsf{K}U$,

(b) $[x]x$ is I, and

(f) If X is YZ, $x \in \mathrm{FV}(X)$, and $[x]Y$ and $[x]Z$ are already defined, then $[x]X$ is $\mathsf{S}([x]Y)([x]Z)$.[76]

Still other definitions are possible, and some are significantly more efficient than this one.

Remark 8 The notation $[x]X$ is Curry's original notation, and it is still in use by many computer scientists. The abstraction operator in combinatory logic is often called "bracket abstraction." Another notation, proposed by Barendregt [Barendregt, 1984] and used in [Hindley and Seldin, 1986] is $\lambda^* x \,.\, X$, but this notation seems currently less common.

We can now formally define combinatory logic:

Definition 7 The formal system H of *combinatory logic* is defined as follows:

1. *Terms* are defined as follows:

[76]On the names of these clauses, see [Curry and Feys, 1958, §6A3, page 190] and [Curry *et al.*, 1972, §11C1, page 43].

   (a) *Variables*. There is given an infinite sequence of variables, x, y, z, x_1, etc.
   (b) *Constants*. There are two special constants: K and S. There may be more constants.
   (c) *Applications*. If X and Y are terms, so is (XY).

2. All occurrences of variables in terms are free. That x occurs in X will be denoted $x \in \mathrm{FV}(X)$.

3. The *substitution* of a term Y for a variable x in another term X, $[Y/x]X$, is defined as in Definition 3, Clause 3, subclauses (a)-(c).

4. *Abstraction* is defined by the second definition given above just before Remark 8.

5. *(Weak) reduction*, $\triangleright_w$, is defined by the following axioms and rules:

   (K) $\mathsf{K}XY \triangleright X$.
   (S) $\mathsf{S}XYZ \triangleright XZ(YZ)$.
   (ρ) $X \triangleright X$.
   (μ) If $X \triangleright Y$, then $ZX \triangleright ZY$.
   (ν) If $X \triangleright Y$, then $XZ \triangleright YZ$.
   (τ) If $X \triangleright Y$ and $Y \triangleright Z$, then $X \triangleright Z$.

   This is equivalent to defining weak reductions as replacements of redexes by contracta, where there are two kinds of redexes: 1) K redexes ($\mathsf{K}XY$), whose contractum is X, and 2) S redexes ($\mathsf{S}XYZ$), whose contractum is $XZ(YZ)$.

6. *(Weak) conversion*, $=_w$, is defined by changing $\triangleright$ to $=_w$ in the definition of weak reduction and adding the rule

   (σ) If $X =_w Y$, then $Y =_w X$.

   Weak reduction can also be defined as a sequence of contractions and reverse contractions.

Remark 9 Weak reduction and conversion satisfy the Church-Rosser Theorem, i.e., Theorem 5 and Corollary 1.

Combinatory terms can be translated to λ-terms by the following translation:

Definition 8 The *translation* $-_L$ from terms in H to terms in Λ is defined as follows:

1. If a is any atomic term except K or S, then $a_L \equiv a$.
2. $\mathsf{K}_L \equiv \lambda xy \, . \, x$.
3. $\mathsf{S}_L \equiv \lambda xyz \, . \, xz(yz)$.
4. $(XY)_L \equiv (X_L Y_L)$.

With this definition, it is easy to prove the following:

Theorem 8 *If $X \rhd_w Y$ in H, then $X_L \rhd_\beta Y_L$ in Λ. If $X =_w Y$ in H, then $X_L =_\beta Y_L$.*

The converse translation depends on the abstraction algorithm used in H:

Definition 9 The *translation* $-_H$ from Λ to H is defined as follows:

1. If a is an atomic term, then $a_H \equiv a$.
2. $(MN)_H \equiv (M_H N_H)$.
3. $(\lambda x \, . \, M)_H \equiv ([x]M_H)$.

Remark 10 Note that for any abstraction algorithm, if $y \notin \mathrm{FV}(M)$, then $([x]M)_H \equiv ([y][y/x]M_H)$. Thus, rule (α) is mapped to an identity between terms by the mapping $-_H$. This is another reason for identifying α-convertible terms in λ-calculus.

Remark 11 There is no theorem for the translation $-_H$ corresponding to Theorem 8. The reason for this is that rule (ξ) does not hold for $\rhd_w$. This can be seen from the following example: $\mathsf{I}x \rhd_w x$, but

$$[x](\mathsf{I}x) \equiv \mathsf{S}(\mathsf{K}\mathsf{I})\mathsf{I} \not\rhd_w \mathsf{I} \equiv [x]x.$$

A form of $\rhd_w$ and $=_w$ can be defined for λ-terms: drop rule (ξ) or, which is equivalent, forbid replacements within the scope of a λ-abstract. For this reduction, we do have that if $M \rhd_w N$ (respectively $M =_w N$), then $M_H \rhd_w N_H$ (respectively $M_H =_w N_H$).

D Combinators with Types

To have a Church-style typing, we would need a combinator $\mathsf{K}_{\alpha\beta}$ of type $\alpha \to \beta \to \alpha$ for each pair of types α and β, and similarly for S, so that there would be a combinator I_α of type $\alpha \to \alpha$ for each type α. This would lead to an infinite number of atomic combinators, and so is not usually considered. Instead, type assignment for combinators is usually taken in the Curry style.

In assigning types to terms of combinatory logic, deductions that terms have types will follow the construction of the terms. For this reason, the definition of the system is as follows:

Definition 10 The *system TACu* is defined as follows:

1. The *types* are those defined in Definition 4 above.
2. The *terms* are the terms of the system H of Definition 7 above.
3. Typing judgements are of the form $X : \alpha$, where X is a term and α is a type.
4. If c is an atomic constant different from K and S, and if c is assigned type γ, then $c : \gamma$ is an axiom. There are two additional axiom schemes:

   (FK) For all types α and β, $\mathsf{K} : (\alpha \to \beta \to \alpha)$.

   (FS) For all types α, β, and γ, $\mathsf{S} : (\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to (\alpha \to \gamma)$.
5. The rule of the system is the rule ($\to$ E) of Definition 5, clause 5.

The Subject-Reduction Theorem, Theorem 6 above, and the Normal Form Theorem, Theorem 7 above, both hold for system TACu.

References

[Andrews, 1965] P. B. Andrews. *A Transfinite Type Theory with Type Variables.* North-Holland Publishing Company, Amsterdam, 1965.

[Andrews, 1986] P. B. Andrews. *An Introduction to Mathematical Logic and Type Theory: to Truth thorugh Proof.* Academic Press, 1986.

[Barendregt *et al.*, 1993] H. P. Barendregt, M. W. Bunder, and W. J. W. Dekkers. Systems of illative combiantory logic complete for first-order propositional and predicate calculus. *Journal of Symbolic Logic*, 58(3):769–788, September 1993.

[Barendregt, 1984] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland Publishing Company, Amsterdam, New York, and Oxford, 1984.

[Bernays, 1953] P. Bernays. Review of H. B. Curry, "The system LD," J. Symbolic Logic 17:35–42, 1952 and "On the definition of negation by a fixed proposition in the inferential calculus," Ibid., pp. 98–104. *Journal of Symbolic Logic*, 18:266–268, 1953.

[Bunder and Meyer, 1978] M. W. Bunder and R. K. Meyer. On the inconsistency of systems similar to $\mathcal{F}^*_{21}$. *Journal of Symbolic Logic*, 43(1):1–2, March 1978.

[Bunder, 1969] M. W. Bunder. *Set Theory Based on Combinatory Logic*. PhD thesis, University of Amsterdam, 1969.

[Bunder, 1974] M. W. Bunder. Some inconsistencies in illative combinatory logic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 20:199–201, 1974.

[Bunder, 1976] M. W. Bunder. The inconsistency of $\mathcal{F}^*_{21}$. *Journal of Symbolic Logic*, 41(2):467–468, June 1976.

[Cardone and Hindley, 2006] F. Cardone and J. R. Hindley. History of lambda-calculus and combinatory logic. To appear in the *Handbook of the History of Logic*, volume 5., 2006.

[Church and Kleene, 1936] A. Church and S. C. Kleene. Formal definitions in the theory of ordinal numbers. *Fundamenta Mathematicae*, 28:11–31, 1936.

[Church and Rosser, 1936] A. Church and J. B. Rosser. Some properties of conversion. *Transactions of the American Mathematical Society*, 39:472–482, 1936.

[Church, 1924] A. Church. Uniqueness of the Lorentz transformation. *American Mathematical Monthly*, 31:376–382, 1924.

[Church, 1925] A. Church. On irredundant sets of postulates. *Transactions of the American Mathematical Society*, 27:318–328, 1925.

[Church, 1927] A. Church. Alternatives to Zermelo's assumption. *Transactions of the American Mathematical Society*, 29:178–208, January 1927.

[Church, 1928] A. Church. On the law of the excluded middle. *Bulletin of the American Mathematical Society*, 34:75–78, January–February 1928.

[Church, 1932] A. Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33:346–366, 1932.

[Church, 1933] A. Church. A set of postulates for the foundation of logic (second paper). *Annals of Mathematics*, 34:839–864, 1933.

[Church, 1935] A. Church. A proof of freedom from contradiction. *Proceedings of the National Academy of Sciences of the United States of America*, 21:275–281, 1935.

[Church, 1936a] A. Church. Correction to *A note on the Entscheidungsproblem*. *Journal of Symbolic Logic*, 1(3):101–102, September 1936.

[Church, 1936b] A. Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(1):40–41, March 1936.

[Church, 1936c] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:345–363, 1936.

[Church, 1938] A. Church. The constructive second number class. *Bulletin of the American Mathematical Society*, 44:224–232, April 1938.

[Church, 1940] A. Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[Church, 1941] A. Church. *The Calculi of Lambda Conversion*. Princeton University Press, 1941.

[Church, 1951] A. Church. A formulation of the logic of sense and denotation. In P. Henley, editor, *Sturcture, Method and Meaning: Essays in Honor of Henry M. Sheffer*, pages 3–24. Liberal Arts Press, New York, 1951.

[Church, 1973] A. Church. Outline of a revised formulation of the logic of sense and denotation (Part I). *Noûs*, 7(1):24–33, March 1973.

[Church, 1974] A. Church. Outline of a revised formulation of the logic of sense and denotation (Part II). *Noûs*, 8(2):135–156, May 1974.

[Church, 1989] A. Church. Intensionality and the paradox of the name relation. In J. Almog, J. Perry, and H. Wettstein, editors, *Themes from Kaplan*, pages 151–165. Oxford University Press, New York, 1989.

[Church, 1993] A. Church. A revised formulation of the logic of sense and denotation. alternative (1). *Noûs*, 27(2):141–157, June 1993.

[Cogan, 1955] E. J. Cogan. A formalization of the theory of sets from the point of view of combinatory logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 1:198–240, 1955. PhD thesis, The Pennsylvania State University, 1955.

[Constable and others, 1986] R. Constable et al. *Implementing Mathematics with the Nuprl Proof Development System.* Prentice-Hall, Englewood Cliffs, NJ, 1986.

[Coquand and Huet, 1988] Thierry Coquand and Gérard Huet. The calculus of constructions. *Information and Computation*, 76:95–120, 1988.

[Curry and Feys, 1958] H. B. Curry and R. Feys. *Combinatory Logic*, volume 1. North-Holland Publishing Company, Amsterdam, 1958. Reprinted 1968 and 1974.

[Curry and Schoenberg, 1947] H. B. Curry and I. J. Schoenberg. On spline distributions and their limits: The Pólya distribution functions (Absract). *Bulletin of the American Mathematical Society*, 53:1114, 1947.

[Curry and Schoenberg, 1966] H. B. Curry and I. J. Schoenberg. On Pólya frequency functions IV: the fundamental spline functions and their limits. *Journal D'Analyse Mathématique*, 17:71–107, 1966.

[Curry *et al.*, 1972] H. B. Curry, J. R. Hindley, and J. P. Seldin. *Combinatory Logic*, volume 2. North-Holland Publishing Company, Amsterdam and London, 1972.

[Curry, 1929] H. B. Curry. An analysis of logical substitution. *American Journal of Mathematics*, 51:363–384, 1929.

[Curry, 1930] H. B. Curry. Grundlagen der kombinatorischen Logik. *American Journal of Mathematics*, 52:509–536, 789–834, 1930. Inauguraldissertation.

[Curry, 1931] H. B. Curry. The universal quantifier in combinatory logic. *Annals of Mathematics*, 32:154–180, 1931.

[Curry, 1932] H. B. Curry. Some additions to the theory of combinators. *American Journal of Mathematics*, 54:551–558, 1932.

[Curry, 1933] H. B. Curry. Apparent variables from the standpoint of combinatory logic. *Annals of Mathematics*, 34:381–404, 1933.

[Curry, 1934a] H. B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences of the United States of America*, 20:584–590, 1934.

[Curry, 1934b] H. B. Curry. Some properties of equality and implication in combinatory logic. *Annals of Mathematics*, 35:849–850, 1934.

[Curry, 1935] H. B. Curry. Foundations of the theory of abstract sets from the standpoint of combinatory logic (abstract). *Bulletin of the American Mathematical Society*, 40:654, 1935.

[Curry, 1936] H. B. Curry. First properties of functionality in combinatory logic. *Tôhoku Mathematical Journal*, 41:371–401, 1936.

[Curry, 1937] H. B. Curry. Some properties of formal deducibility (Abstract). *Bulletin of the American Mathematical Society*, 43:615, 1937.

[Curry, 1939a] H. B. Curry. A note on the reduction of Gentzen's calculus LJ. *Bulletin of the American Mathematical Society*, 45:288–293, 1939.

[Curry, 1939b] H. B. Curry. Remarks on the definition and nature of mathematics. *Journal of Unified Science*, 9:164–169, 1939. All copies of this issue were destroyed during World War II, but some copies were distributed at the International Congress for the Unity of Science in Cambridge, MA, in September, 1939. Reprinted with minor corrections in *Dialectica* 8:228–333, 1954. Reprinted again in Paul Benacerraf and Hilary Putnam (eds) *Philosophy of Mathematics: Selected Readings*, pages 152–156, Prentice Hall, Englewood-Cliffs, New Jersey, 1964.

[Curry, 1941a] H. B. Curry. The consistency and completeness of the theory of combinators. *Journal of Symbolic Logic*, 6:54–61, 1941.

[Curry, 1941b] H. B. Curry. The paradox of Kleene and Rosser. *Transactions of the American Mathematical Society*, 50:454–516, 1941.

[Curry, 1941c] H. B. Curry. A revision of the fundamental rules of combinatory logic. *Journal of Symbolic Logic*, 6:41–53, 1941.

[Curry, 1941d] H. B. Curry. Some aspects of the problem of mathematical rigor. *Bulletin of the American Mathematical Society*, 47:221–241, 1941.

[Curry, 1942a] H. B. Curry. The combinatory foundations of mathematical logic. *Journal of Symbolic Logic*, 7:49–64, 1942.

[Curry, 1942b] H. B. Curry. The inconsistency of certain formal logics. *Journal of Symbolic Logic*, 7:115–117, 1942.

[Curry, 1942c] H. B. Curry. Some advances in the combinatory theory of quantification. *Proceedings of the National Academy of Sciences of the United States of America*, 28:564–569, 1942.

[Curry, 1949] H. B. Curry. A simplification of the theory of combinators. *Synthese*, 7:391–399, 1949.

[Curry, 1950] H. B. Curry. *A Theory of Formal Deducibility.* Notre Dame Mathematical Lectures Number 6. University of Notre Dame, Notre Dame, Indiana, 1950. Second edition, 1957.

[Curry, 1951] H. B. Curry. *Outlines of a Formalist Philosophy of Mathematics.* North-Holland Publishing Company, Amsterdam, 1951.

[Curry, 1952a] H. B. Curry. The elimination theorem when modality is present. *Journal of Symbolic Logic*, 17(4):249–265, December 1952.

[Curry, 1952b] H. B. Curry. *Leçons de Logique Algébrique.* Gauthier-Villars and Nauwelaerts, Paris and Louvain, 1952.

[Curry, 1952c] H. B. Curry. A new proof of the Church-Rosser Theorem. *Indagationes Mathematicae*, 14:16–23, 1952.

[Curry, 1952d] H. B. Curry. On the definition of negation by a fixed proposition in inferential calculus. *Journal of Symbolic Logic*, 17(2):98–104, June 1952.

[Curry, 1952e] H. B. Curry. The permutability of rules in the classical inferential calculus. *Journal of Symbolic Logic*, 17(4):245–248, December 1952.

[Curry, 1952f] H. B. Curry. The system LD. *Journal of Symbolic Logic*, 17(1):35–42, March 1952.

[Curry, 1954] H. B. Curry. The logic of program composition. In *Applications Scientifiques de la Lagique Mathématique, Actes du 2^e Colloque International de Logique Mathématique, Paris—25–30 Août 1952, Institut Henri Poincaré*, pages 97–102. Gauthier-Villars, Paris, 1954.

[Curry, 1955] H. B. Curry. The inconsistency of the full theory of combinatory functionality (abstract). *Journal of Symbolic Logic*, 20:91, 1955.

[Curry, 1956] H. B. Curry. Consistency of the theory of functionality (abstract). *Journal of Symbolic Logic*, 21:110, 1956.

[Curry, 1960] H. B. Curry. The deduction theorem in the combinatory theory of restricted generality. *Logique et Analyse*, 3:15–39, 1960.

[Curry, 1961] H. B. Curry. Basic verifiability in the combinatory theory of restricted generality. In *Essays on the Foundations of Mathematics*, pages 165–189. Magnes Press, Hebrew University, Jerusalem, 1961.

[Curry, 1963] H. B. Curry. *Foundations of Mathematical Logic*. McGraw-Hill Book Company, Inc., New York, San Francisco, Toronto, and London, 1963. Reprinted by Dover, 1977 and 1984.

[Curry, 1965] H. B. Curry. Remarks on inferential deduction. In A.-T. Tymieniecka, editor, *Contributions to logic and Methodology in Honor of J. M. Bocheński*, pages 45–72. North-Holland, Amsterdam, 1965.

[Curry, 1968a] H. B. Curry. A deduction theorem for infrential predicate calculus. In K. Schütte, editor, *Contributions to Mathematical Logic*, pages 91–108. North-Holland, Amsterdam, 1968.

[Curry, 1968b] H. B. Curry. The equivalence of two definitions of elementary formal system. *Compositio Mathematica*, 20:13–20, 1968. Festschrift in honor of A. Heyting, edited by D. van Dalen, S. C. Kleene, and A. S. Troelstra, Wolters-Noordhoff, Groningen.

[Curry, 1969a] H. B. Curry. Modified basic functionality in combinatory logic. *Dialectica*, 23:83–92, 1969.

[Curry, 1969b] H. B. Curry. The undecidability of $\lambda\mathsf{K}$-conversion. In J. J. Bulloff, T. C. Holyoke, and S. W. Hahn, editors, *Foundations of Mathematics: Symposium Papers Commemorating the Sixtieth birthday of Kurt Gödel*, pages 10–14. Springer Verlag, Berlin, 1969.

[Curry, 1973] H. B. Curry. The consistency of a system of combinatory restricted generality. *Journal of Symbolic Logic*, 38:83–92, 1973.

[Curry, 1975a] H. B. Curry. Representation of Markov algorithms by combinators. In A. R. Anderson, R. B. Marcus, and R. A. Martin, editors, *The Logical Enterprise*, pages 109–119. Yale University Press, New Haven, Connecticut, 1975. Festschrift in honor of F. B. Fitch.

[Curry, 1975b] H. B. Curry. A study of generalized standardization in combinatory logic. In J. Diller and G. H. Müller, editors, *ISLIC Proof Theory Symposium. Dedicated to Kurt Schütte on the Occasion of his 65th Birthday. Proceedings of the International Summer Institute and Logic Colloquium, Kiel, 1974*, pages 44–55. Springer Verlag, Berlin, 1975. *Lecture Notes in Mathematics*, volume 500.

[Curry, 1979] H. B. Curry. On a polynomial representation of $\lambda\beta$-normal forms. In K. Lorenz, editor, *Konstruktionen versus Positionen: Beiträge zur Diskussion um die Konstruktive Wissenschaftstheorie*, pages 94–98. Walter de Gruyter, Berlin and New York, 1979.

[Curry, 1980] H. B. Curry. Some philosophical aspects of combinatory logic. In John Barwise, H. Jerome Keisler, and Kenneth Kunen, editors, *The Kleene Symposium: A Collection of Papers on Recursion Theory and Intuitionism*, pages 85–101. North-Holland, Amsterdam, 1980.

[Dekkers *et al.*, 1998a] W. J. M. Dekkers, M. W. Bunder, and H. P. Barendregt. Completeness of the propositions-as-types interpretation of intuitionistic logic into illative combinatory logic. *Journal of Symbolic Logic*, 63(3):869–890, September 1998.

[Dekkers *et al.*, 1998b] W. J. M. Dekkers, M. W. Bunder, and H. P. Barendregt. Completeness of two systems of illative combinatory logic for first-order propositional and predicate calculus. *Archive for Mathematical Logic*, 37(5–6):327–341, 1998.

[Enderton, 1995] H. B. Enderton. In memoriam: Alonzo Church. *Bulletin of Symbolic Logic*, 1(4):486–488, December 1995.

[Fitch, 1952] Fredric Brenton Fitch. *Symbolic Logic.* The Ronald Press Company, New York, 1952.

[Gandy, 1980] R. O. Gandy. An Early Proof of Normalization by A.M. Turing. In J. P. Seldin and J. R. Hindley. editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 453–455. Academic Press, 1980.

[Gentzen, 1934] Gerhard Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934. Translated in Sabo (ed.), *The Collected Papers of Gerhard Gentzen* as "Investigations into Logical Deduction," pages 68–131, North-Holland, Amsterdam and London, 1969.

[Girard, 1971] J.-Y. Girard. Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J. E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 63–92, Amsterdam, 1971. North-Holland.

[Girard, 1972] J.-Y. Girard. *Interprétation Fonctionnelle et Élimination des Coupures de L'Arithmétique d'Orde Supérieur.* PhD thesis, University of Paris VII, 1972.

[Gödel, 1940] Kurt Gödel. *The Consistency of the Continuum Hypothesis.* Princeton University Press, Princeton, New Jersey, USA, 1940.

[Goldstine, 1946] A. Goldstine. Report on the eniac (electronic numeric integrator and computer). Technical report, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, 1946.

[Henkin, 1950] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, June 1950.

[Henkin, 1963] L. Henkin. A theory of propositional types. *Fundamenta Mathematicae*, 52:323–344, 1963.

[Hindley and Seldin, 1986] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and λ-Calculus*. Cambridge University Press, 1986.

[Hindley, 1964] J. R. Hindley. *The Church-Rosser Property and a Result in Combinatory Logic*. PhD thesis, Newcastle upon Tyne, 1964.

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

[Howard, 1980] W. A. Howard. The formulae-as-types notion of construction. In J. Roger Hindley and Jonathan P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, New York, 1980. A version of this paper was privately circulated in 1969.

[Jaśkowski, 1934] Stanisław Jaśkowski. On the rules of supposition in formal logic. *Studia Logica*, 1:5–32, 1934.

[Kamareddine *et al.*, 2004] F. Kamareddine, T. Laan, and R. Nederpelt. *A Modern Perspective on Type Theory*. Kluwer, Dordrecht, Boston, London, 2004.

[Kanger, 1955] S. Kanger. A note on partial postulate sets for propositional logic. *Theoria (Sweden)*, 21:99–104, 1955.

[Kleene and Rosser, 1935] S. C. Kleene and J. B. Rosser. The inconsistency of certain formal logics. *Annals of Mathematics*, 36:630–636, 1935.

[Kleene, 1934] S. C. Kleene. Proof by cases in formal logic. *Annals of Mathematics*, 35(3):529–544, July 1934.

[Kleene, 1935] S. C. Kleene. A theory of positive integers in formal logic. *American Journal of Mathematics*, 57:153–173, 219–244, 1935.

[Kleene, 1936a] S. C. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112:727–742, 1936.

[Kleene, 1936b] S. C. Kleene. λ-definability and recursiveness. *Duke Mathematical Journal*, 2:340–353, 1936.

[Kleene, 1938] S. C. Kleene. On notation for ordinal numbers. *Journal of Symbolic Logic*, 3(4):150–155, December 1938.

[Kleene, 1944] S. C. Kleene. On the forms of predicates in the theory of constructive ordinals. *American Journal of Mathematics*, 66:41–58, 1944.

[Kleene, 1955] S. C. Kleene. On the forms of predicates in the theory of constructive ordinals (second paper). *American Journal of Mathematics*, 77:405–428, 1955.

[Kleene, 1962] S. C. Kleene. Lambda-definable functionals of finite type. *Fundamenta Mathematicae*, 50:281–303, 1962.

[Kleene, 1981] S. C. Kleene. Origins of recursive function theory. *Annals of the History of Computing*, 3:52–67, 1981.

[Kripke, 1958] S. A. Kripke. The system LE. Submitted to Westinghouse Science Talent Search, February, 1958; not published, 1958.

[Lercher, 1963] B. Lercher. *Strong reduction and recursion in combinatory logic*. PhD thesis, The Pennsylvania State University, 1963.

[Loewen, 1962] K. Loewen. *A study of strong reduction in combinatory logic*. PhD thesis, the Pennsylvania State University, 1962.

[Lorenz, 1961] K. Lorenz. *Arithmetik und Logik als Spiele*. PhD thesis, Universität Kiel, 1961.

[Lorenzen and Lorenz, 1978] P. Lorenzen and K. Lorenz. *Dialogische Logik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1978.

[Lorenzen, 1961] P. Lorenzen. Ein dialogisches Konstruktivitätskriterium. In *Infinitistic Methods: Proceedings of the Symposium on Foundations of Mathematics, Warsaw, 2–9 September 1959*, pages 193–200. Pergamon Press and Państwowe Wydawnictwo Naukowe, 1961.

[Martin-Löf, 1975] P. Martin-Löf. An intuitionistic theory of types: Predicative part. In H. E. Rose and J. C. Shepherdson, editors, *Logic Colloquium '73*, pages 73–118. North-Holland Publishing Company, Amsterdam, 1975.

[Martin-Löf, 1984] P. Martin-Löf. Intuitionistic type theory. Bibliopolis, Naples, 1984. Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980.

[Mezghiche, 1984] Mohamed Mezghiche. Une nouvelle Cβ-réduction dans la logique combinatoire. *Theoretical Computer Science*, 31:151–165, 1984.

[Mezghiche, 1989] Mohamed Mezghiche. On pseudo-cβ normal form in combinatory logic. *Theoretical Computer Science*, 66:323–331, 1989.

[Mezghiche, 1997] Mohamed Mezghiche. $c\beta$-machine with $\lambda\beta$-reduction. *Theoretical Computer Science*, 189:221–228, 1997.

[Newman, 1942] M. H. A. Newman. On theories with a combinatorial definition of "equivalence". *Annals of Mathematics*, 43(2):223–243, 1942.

[Popper, 1968] K. R. Popper. Epistemology without a knowing subject. In *Logic, Methodology and Philosophy of Science III: Proceedings of the Third International Congress for Logic, Methodology and Philosophy of Science, Amsterdam 1967*, pages 333–373, Amsterdam, 1968. North-Holland.

[Prawitz, 1965] D. Prawitz. *Natural Deduction.* Almqvist & Wiksell, Stockholm, Göteborg, and Uppsala, 1965.

[Rosser, 1935a] J. B. Rosser. A mathematical logic without variables, Part I. *Annals of Mathematics*, 36:127–150, 1935.

[Rosser, 1935b] J. B. Rosser. A mathematical logic without variables, Part II. *Duke Mathematical Journal*, 1:328–355, 1935.

[Rosser, 1936] J. B. Rosser. Extensions of some theorems of Gödel and Church. *Journal of Symbolic Logic*, 1(3):87–91, September 1936.

[Rosser, 1942] J. B. Rosser. New sets of postulates for combinatory logics. *Journal of Symbolic Logic*, 7:18–27, 1942.

[Rosser, 1984] J. B. Rosser. Highlights of the history of the lambda-calculus. *Annals of the History of Computing*, 6:337–339, 1984.

[Sanchis, 1963] L. E. Sanchis. *Normal combinations and the theory of types.* PhD thesis, The Pennsylvania State University, 1963.

[Sanchis, 1964] L. E. Sanchis. Types in combinatory logic. *Notre Dame Journal of Formal Logic*, 5:161–180, 1964.

[Schönfinkel, 1924] M. Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, 92:305–316, 1924. English translation, "On the building blocks of mathematical logic," in Jean van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*, pages 355–366, Harvard University Press, Cambridge, MA and London, 1967.

[Seldin, 1968] J. P. Seldin. *Studies in Illative Combinatory Logic.* PhD thesis, University of Amsterdam, 1968.

[Seldin, 1975a] J. P. Seldin. Arithmetic as a study of formal systems. *Notre Dame Journal of Formal Logic*, 16:449–464, 1975.

[Seldin, 1975b] J. P. Seldin. Equality in $\mathcal{F}_{22}$. In H. E. Rose and J. C. Shepherdson, editors, *Logic Colloquium '73*, pages 433–444. North-Holland, 1975.

[Seldin, 1977] J. P. Seldin. The Q-consistency of $\mathcal{F}_{22}$. *Notre Dame Journal of Formal Logic*, 18:117–127, 1977.

[Seldin, 1980] J. P. Seldin. Curry's program. In J. R. Hindley and J. P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 3–33. Academic Press, London, 1980.

[Seldin, 2000] J. P. Seldin. On the role of implication in formal logic. *Journal of Symbolic Logic*, 65(3):1076–1114, 2000.

[Seldin, 2003] J. P. Seldin. Curry's anticipation of the types used in programming languages. In *Proceedings of the Canadian Society for the History and*

Philosophy of Mathematics, Volume 15, Twenty-Eighth Annual Meeting, University of Toronto, Toronto, Ontario, 24–26 May 2002, pages 148–163. Canadian Society for the History and Philosophy of Mathematics, 2003.

[Shapiro, 2000] Stewart Shapiro. *Thinking About Mathematics.* Oxford University Press, Oxford, England, 2000.

[Smullyan, 1961] R. Smullyan. *Theory of Formal Systems.* Princeton University Press, Princeton, New Jersey, 1961.

[Stegmüller, 1964] W. Stegmüller. Remarks on the completeness of logical systems relative to the validity-concepts of P. Lorenzen and K. Lorenz. *Notre Dame Journal of Formal Logic*, 5:81–112, 1964.

[Titgemeyer, 1961] R. Titgemeyer. Über einen Widerspruch in Cogans Darstellung der Mengenlehre. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 7:161–163, 1961.

[Troelstra and van Dalen, 1988] A. S. Troelstra and D. van Dalen. *Constructivism in mathematics: An introduction*, volume I. North-Holland, Amsterdam, 1988.

[Turing, 1936–1937] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1936–1937. Correction, *Ibid.* 43:544–546, 1937.

[Turing, 1937a] A. M. Turing. Computability and λ-definability. *Journal of Symbolic Logic*, 2(4):153–163, December 1937.

[Turing, 1937b] A. M. Turing. The $\mathfrak{p}$-function in λ-K-conversion. *Journal of Symbolic Logic*, 2(4):164, December 1937.

[von Neumann, 1925] J. von Neumann. Eine Axiomatizierung der Mengenlehre. *Journal für die Reine und Angewandte Mathematik*, 154:219–240, 1925. Correction, *ibid.* 155:128. English translation: An axiomatization of set theory, in J. van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic 1879–1931*, Harvard University Press, Cambridge, Mass., USA, 1967, pp. 393–413.

[von Neumann, 1928] J. von Neumann. Die Axiomatizierung der Mengenlehre. *Mathematische Zeitschrift*, 27:669–752, 1928. Based on the author's doctoral dissertation in Hungarian, University of Budapest, September 1925 (author's footnote, p. 669). Reprinted in A. H. Taub, editor, *The Collected Works of J. von Neumann*, volume 1, Pergamon Press, Oxford, England, 1961, pp. 339–422.

[Whitehead and Russell, 1910–1913] A. N. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, Cambridge, England, 1910–1913. Second edition, 1925–1927.

Index