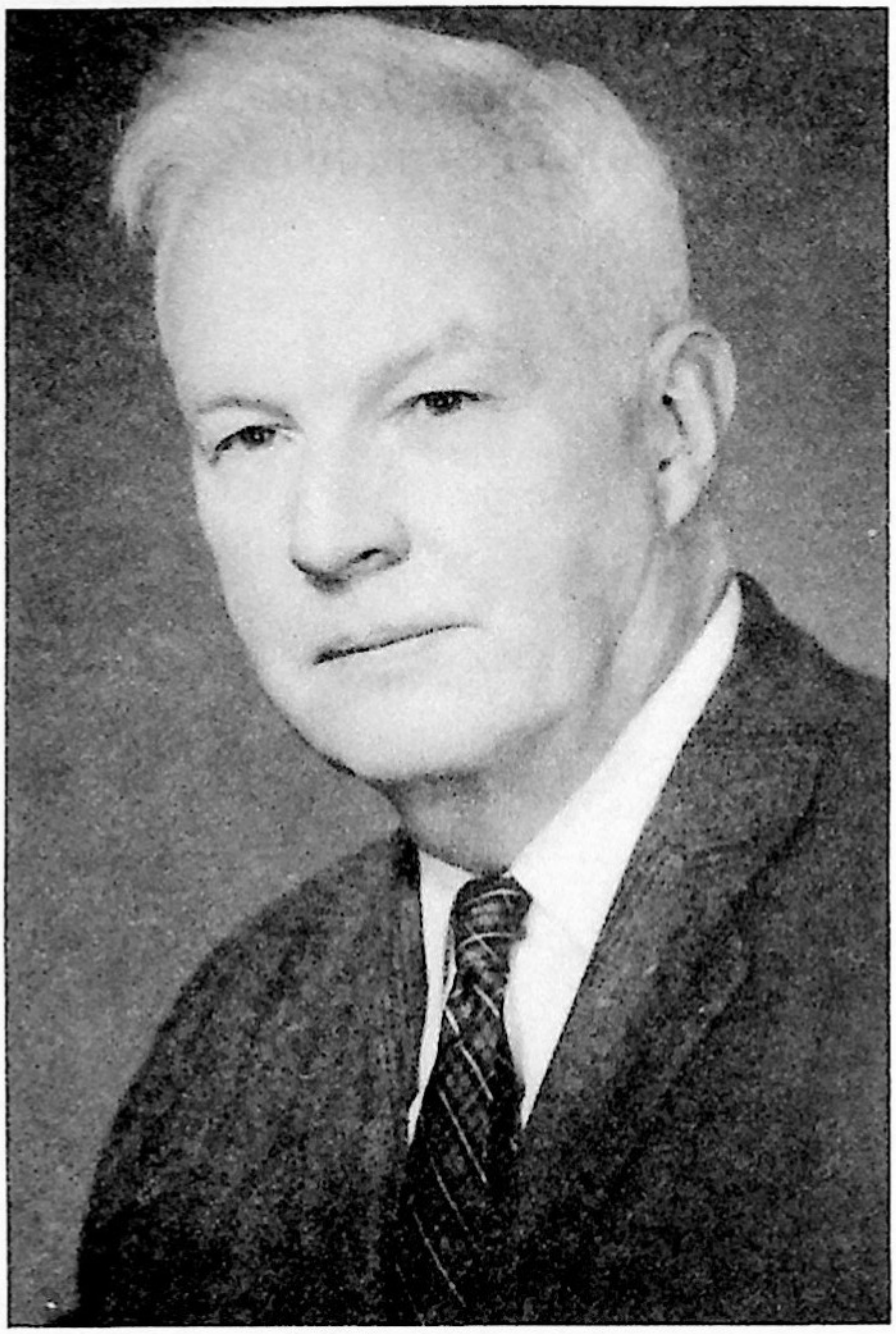
To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism



Dedicated to Haskell B. Curry on the occasion of his 80th Birthday

To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism

Edited by

J.P. SELDIN
Mathematics Department
Concordia University
Montreal, Canada

J.R. HINDLEY
Department of Pure Mathematics
University College of Swansea
Swansea, Wales

1980

ACADEMIC PRESS
A Subsidiary of Harcourt Brace Jovanovich, Publishers
London New York Toronto Sydney San Francisco

ACADEMIC PRESS INC. (LONDON) LTD.
24/28 Oval Road,
London NW1

United States Edition published by
ACADEMIC PRESS INC.
111 Fifth Avenue
New York, New York 10003

British Library Cataloguing in Publication Data
To H B Curry, essays on combinatory logic, lambda-calculus and formalism.
1. Logic, Symbolic and mathematical – Addresses, essays, lectures
I. Curry, Haskell Brooks II. Hindley James Roger
III. Seldin, Jonathan Paul
511'.3 QA9 80-40139
ISBN 0-12-349050-2

Printed in Great Britain by
Whitstable Litho Ltd., Whitstable, Kent

PREFACE

Haskell B. Curry is one of the founders of modern mathematical logic in the United States. After getting his degree at Göttingen under Hilbert, he returned to the United States at a time when almost the only other American mathematical logician was Alonzo Church. A few years later he became a founder of the Association for Symbolic Logic. In 1971, he and Church were jointly honored by a symposium at a meeting of the Association for Symbolic Logic, but so far no Festschrift has appeared in his honor. In recent years the interest in combinatory logic and related topics has grown enormously and at the same time the suggestion that there should be such a Festschrift has been made by several people, including in particular R. K. Meyer and R. M. Martin. The occasion of his 80th birthday on September 12, 1980 seems especially appropriate for one.

When Academic Press gave this project their encouragement and support, we circulated an invitation to contribute to former students, colleagues, and friends of Professor Curry. The enthusiastic response was highly gratifying. Many excellent papers on the subjects selected for the volume were offered, and we feel that those included represent a fitting tribute to him and his work. (Unfortunately, several other excellent papers had to be omitted for lack of space.)

In addition to the papers, the volume contains a list of those who wish to honor Professor Curry by sending their greetings. To these greetings, we the editors wish to add our acknowledgement of our own personal debt to Haskell Curry. He has been our main source of stimulation and ideas, both mathematical and philosophical. And working with him has been in itself one of the rewards of studying combinatory logic. He has always been approachable and willing to discuss a new idea or a difficult point. He has freely given encouragement and support, which he has also given to all younger logicians who came into contact with him, as many of those whose names are in the Tabula Gratulatoria can testify. (And on the non-technical side, his good sense and sound judgement have been clearly shown in his choice of his wife, Virginia. Her hospitality and cheerful personality are one of the great pleasures of combinatory logic.)

We record here our gratitude to those who put aside other work to referee the papers in an unusually short period of time. We also thank all those who typed the camera copy of the volume so quickly and accurately, and Martin Bunder for his generous help with some of the proof-reading. Finally, we are very grateful to the people at Academic Press for their encouragement and efforts on behalf of this project.

Roger Hindley
Jonathan P. Seldin

July 1980

A SHORT BIOGRAPHY OF HASKELL B. CURRY

Haskell Brooks Curry was born on September 12, 1900, at Millis, Massachusetts. His father, Samuel Silas Curry, was president of the School of Expression in Boston, and his mother, Anna Baright Curry, was its dean. (The institution is now known as Curry College.) He graduated from high school in 1916 and entered Harvard College with the intention of going into medicine. During his Freshman year, he took a mathematics course at the suggestion of his advisor and did very well. In the spring of 1917, the United States entered World War I, and Haskell responded by enlisting in the army, becoming a member of the Harvard Unit of the Student Army Training Corps on October 18, 1918. He felt that he would never see action if he continued with his premedical course, and so he changed his major to mathematics with the idea of going into the artillery. The war ended that November and Haskell left the army on December 9, 1918, but he kept on in mathematics, receiving his A. B. in 1920.

For the next two years he studied electrical engineering at MIT in a program that involved working half-time at the General Electric Company. Because he was usually interested in why an answer was correct when the engineers seemed only interested in the fact that it was correct, he decided that he would be better off pursuing a degree in pure science, and in 1922 he switched to physics. He returned to Harvard, where for the year 1922-23 he was a half-time research assistant to P. W. Bridgman. In 1924 he received his A. M. in physics (from Harvard). But by this time, his interests had shifted still further, and he now switched to mathematics.

He continued to study mathematics at Harvard until 1927, where he was a half-time instructor during the first semester of 1926-27 but otherwise studied full-time. He was also involved in the business affairs of his family: his father had died in 1921, and his mother had followed in 1924. From 1921 until 1927 (when it was incorporated), Haskell was in charge as a trustee of the business of his father's estate, and at the incorporation, when it became known as the Expression Company, he became its treasurer. It sold out in 1928.

Meanwhile, Haskell had become interested in logic. Originally, all of his logic was reading on the side, and at one point he was supposed to be working on a dissertation on a topic in differential equations assigned to him by George Birkhoff. Furthermore, faculty members at Harvard and elsewhere to whom he spoke about logic, and expecially Norbert Wiener, who was at MIT, advised him to stay away from the field. But during 1926-27 he had the idea of using combinators to analyze the rather complex rule of substitution in the first part of *Principia Mathematica*, as he has explained in Curry and Feys (1958a), pp. 1-4, 184. (See the bibliography, which appears after this Biographical Sketch.) This led him to think that he had the basis for a dissertation in logic. When he took this idea to several faculty members, he got a different response than he had had earlier about switching to logic. Norbert Wiener, for example, now encouraged him to go on, explaining that he (Wiener) considered logic a subject to be avoided "unless you have something to say", and since Haskell clearly had something to say, "strength to your right arm!"

Thus, Haskell decided to abandon his dissertation on differential equations for one in logic. He decided at this point that it would be useful to teach for a year, and, after getting a recommendation for the position from George Birkhoff, went to Princeton to assume an instructorship for the year 1927-28. At Princeton he consulted with Veblen, who, although a geometer, was always interested in foundational questions. He also began checking the library and, as a result, discovered the paper by Schönfinkel. As soon as he had read this paper, he rushed into Veblen's office to tell him that there was somebody else with the idea for combinators. "Good," said Veblen, "I am always glad when somebody has one of my ideas, for it shows that I am on the right track." Veblen called in Alexander (the topologist), who was then in Princeton. Alexander reported that Schönfinkel was in a mental hospital and was unlikely to continue his work on combinators. He also reported that the man best informed about combinators was Bernays at Göttingen. It was thus decided that Haskell should go to Germany, and as part of an application for financial support, Haskell wrote his first paper, Curry (1929). He was now almost ready to leave for Europe. But first, he got married; to Mary Virginia Wheatley, of Hurlock, Maryland, on July 3, 1928. (Virginia had been a student at the School of Expression, where they met.) After the wedding the Currys left for Germany, where they spent the year 1928-29 at Göttingen.

That year at Göttingen was enough for Haskell to complete his dissertation. His referee was Hilbert, although he actually did most of his work with Bernays, and he was examined on July 24, 1929. Some of his memories of the exam are recorded by Constance Reid in her biography "Hilbert" (Springer, Berlin, 1970), p. 190. The dissertation was published as Curry (1930).

In September 1929, the Currys moved to State College, Pennsylvania, where Haskell took up a position on the faculty of the Pennsylvania State College (since November 14, 1953, the Pennsylvania State University). Today those of us who know Haskell associate him with this institution, but when he first went there he did not plan to stay long. In the beginning he felt cut off from the academic community. Furthermore, in those days Penn State did not support research. (Later, thanks partly to Haskell's influence, Penn State changed its policy and now it is a major research institution.) But his arrival in State College coincided with the beginning of the great depression and the demand for mathematical logicians was not very high. So he remained and settled down at Penn State, staying there, with the exception of several leaves of absence, until his retirement in 1966.

In 1931-32 he went to the University of Chicago as a National Research Council Fellow, and he was able to spend 1938-39 in residence at the Institute for Advanced Study at Princeton.

Otherwise, Haskell spent the 1930's at Penn State teaching and carrying out his research. During this period he was on the reviewing staff of the *Zentralblatt für Mathematik und ihre Grenzgebiete* (1931-39), and in 1936 he was a founding member of the Association for Symbolic Logic (he was Vice President of the Association in 1936-37 and President in 1938-40, as well as being a member of the Council as an ex-President during 1941-46).

During this period, the Currys also began their family: Anne Wright Curry (now Mrs. Richard S. Piper) was born on July 27, 1930, and Robert Wheatley Curry followed on July 6, 1934.

By the end of the 1930's, Haskell was established as one of the most important mathematical logicians in the United States and the entire world. As such, he was asked to present his views on the nature of mathematics to the International Congress for the Unity of Science held at Cambridge, Massachusetts at the beginning of September, 1939. The result was Curry (1951a) (most of which was written in 1939) and a series of papers beginning with Curry (1939b) and continuing to the present.

The following year, 1940, Haskell became a member of the Board of Trustees of Curry College (formerly the School of Expression, the institution of which his father had been president). He remained a member until 1951. Later, on June 5, 1966, the college presented him with the honorary degree Doctor of Science in Oratory.

When the United States entered World War II, Haskell decided to put logic aside for the duration of the war. From 1940 until 1942 he had been a member of the National Committee on War Preparedness of the American Mathematical Society and the Mathematical Association of America. On May 25, 1942, he left Penn State and went to the Frankford Arsenal, where he worked as an applied mathematician until January 1944; then he went to the

Applied Physics Laboratory at Johns Hopkins University, where he remained until March, 1945. Next he went to the Ballistic Research Laboratories at the Aberdeen Proving Ground, where he stayed until September, 1946. During his last three months there, he was Chief of the Theory Section of the Computing Laboratory and for one month he was Acting Chief of the Computing Laboratory; it was during this period that he became involved with the ENIAC computer (see his papers for 1946). As a result of this experience he was a consultant in the field of computing methods to the United States Naval Ordnance Laboratory from June 1, 1948 until June 30, 1949.

In September, 1946, Haskell returned to Penn State. He wanted to pursue his work on electronic computers, and so he tried to interest the university in acquiring some computing equipment. But he was unsuccessful. He persisted until a colleague pointed out to him that if he did succeed, he would probably be made head of the program without any increase in salary. He then decided that this colleague was right and gave up the attempt. This effectively prevented him from pursuing computing theory.

He was, however, getting back to combinatory logic. In Amsterdam in the summer of 1948, during the Tenth International Congress of Philosophy, it was proposed to him that he write a little book of under 100 pages on the subject for the new North-Holland series in logic. He felt that there was too much unpublished research on the subject to write such a short book and so sent them Curry (1951a) instead. But this did suggest to him the project that led to Curry and Feys (1958a) and Curry, Hindley and Seldin (1972). In order to work closely with Feys, he went to Louvain on a Fulbright grant in 1950-51. After his return to Penn State in 1951, he and Feys continued their work, and the manuscript was completed in 1956.

Meanwhile, Haskell began for the first time (now that money for it became available) to have graduate students. Edward J. Cogan first approached him before he went to Louvain. He worked with Haskell after the latter's return in 1951 and finished his dissertation in 1955. Kenneth L. Loewen also studied with him during this period, but he left to take an academic position elsewhere in 1954 and did not finish his dissertation until 1962.

After the completion of Curry and Feys (1958a), Haskell turned his attention to Gentzen-style proof theory. He had done previous work on this (see Curry 1937b, 1939a, 1950a, and several other papers from the 1950's), and he felt that it formalized the kind of reasoning used in the development of combinatory logic; thus he now felt that it should be settled before he began work on Curry *et al.* (1972). Thus he began work on what eventually became Curry (1963a). This work was made easier when, in 1960, he became Evan Pugh Research Professor and was thus relieved of undergraduate teaching duties; the manuscript was completed in 1961. By this time there were two more graduate students, Bruce Lercher and Luis E. Sanchis, both of whom completed their dissertations in 1963.

From February to September, 1962, the Currys took a trip around the world, visiting a number of universities where Haskell gave talks.

The editors of this volume first met Haskell in 1964. Roger Hindley arrived at Penn State to do post-Doctoral work after completing his dissertation at Newcastle-upon-Tyne, and Jonathan P. Seldin arrived as a beginning graduate student. Haskell was just beginning work on what was to become Curry *et al.* (1972). Unfortunately, Feys had died in 1961, and Haskell was left to work alone. He soon realized that he needed collaborators, and so in 1965 he invited Hindley to join him on the project.

In 1966, Haskell retired from Penn State after being there for 37 years. He then went to Amsterdam, where for the next four years he was Professor of Logic, History of Logic, and Philosophy of Science and also Director of the Instituut voor Grondslagenonderzoek en Philosophie der Exacte Wetenschappen, both of the University of Amsterdam. Seldin went to Amsterdam on a Graduate Fellowship from the United States National Science Foundation, and completed his dissertation in 1968, after which he joined Haskell and Hindley as a co-author of the book then being written. Haskell had one more graduate student in Amsterdam, Martin W. Bunder, who finished his dissertation in 1969.

The manuscript of Curry *et al.* (1972) was completed in May, 1970 just before Haskell retired from the University of Amsterdam. He returned to State College, Pennsylvania (the town in which Penn State is located), where he continued his mathematical work, writing reviews (especially for *Mathematical Reviews*) and occasional papers. John A. Lever wrote a master's thesis with him there in 1977 after obtaining special permission from the university authorities to work under a retired professor. In 1971-72, Haskell accepted a visiting position at the University of Pittsburgh. Otherwise, he and Virginia have remained at State College.

Everybody who knows the Currys is aware of how friendly and helpful they always are. Haskell has always done more for colleagues and students than be a source of important ideas (although, of course, his ideas have been of tremendous importance). He has also always been willing to listen to anybody who wanted to talk to him, to discuss their ideas, and to give whatever encouragement he could. (Surely many of us have heard stories of his taking time to help a student having trouble in an elementary course taught by somebody else.) His office door has always been open. And this has undoubtedly been an important contribution to the enthusiasm of many of those of us working in combinatory logic. Also well known wherever the Currys have lived has been the hospitality they have both shown. There are always many parties and other, less formal gatherings, and we conjecture that Virginia's cooking has also played a role in the growth of interest in combinatory logic.

BIBLIOGRAPHY OF HASKELL B. CURRY

This bibliography includes all of Haskell B. Curry's publications on mathematics and philosophy. Abstracts are included only if the complete papers were never published or were not published for ten years or more. Translations and reprintings are listed under the original editions.

1929

An analysis of logical substitution, *Amer. J. Math.* 51, 363-384.

1930

Grundlagen der kombinatorischen Logik (Inauguraldissertation), *Amer. J. Math.* 52, 509-536, 789-834.

1931

The universal quantifier in combinatory logic, *Ann. of Math.* (2) 32, 154-180.

1932

Some additions to the theory of combinators, *Amer. J. Math.* 54, 551-558.

1933

Apparent variables from the standpoint of combinatory logic, *Ann. of Math.* (2) 34, 381-404.

1934

a. Some properties of equality and implication in combinatory logic, *Ann. of Math.* (2) 35, 849-860.
b. Functionality in combinatory logic, *Proc. Nat. Acad. Sci. U. S. A.* 20, 584-590.
c. Foundations of the theory of abstract sets from the standpoint

of combinatory logic (Abstract), *Bull. Amer. Math. Soc.* 40, 654.

1936

a. First properties of functionality in combinatory logic, *Tôhoku Math. J.* 41, Part II, 371-401.
b. A mathematical treatment of the rules of the syllogism, *Mind*, ns 45, 209-216.
c. A note on the associative law in logical algebras, *Bull. Amer. Math. Soc.* 42, 523-524.

1937

a. On the use of dots as brackets in logical expressions, *J. Symbolic Logic* 2, 26-28.
b. Some properties of formal deducibility (Abstract), *Bull. Amer. Math. Soc.* 43, 615.

1939

a. A note on the reduction of Gentzen's calculus LJ, *Bull. Amer. Math. Soc.* 45, 288-293.
b. Remarks on the definition and nature of mathematics, *J. Unified Science* 9, 164-169. Reprinted, *Dialectica* 8 (1954), 288-233. Reprinted *in* "Philosophy of Mathematics: Selected Readings" (Eds. Paul Benacerraf and Hilary Putnam), pp. 152-156. Prentica-Hall, Englewood Cliffs, New Jersey, 1964. Italian translation, Observazioni sulla definizione e sulla natura della matematica. *In* "La Filosofia della Matematica" (Ed. Carlo Cellucci), pp. 153-159. Editori Laterza, Bari, 1967.

1941

a. A formalization of recursive arithmetic, *Amer. J. Math.* 63, 263-282.
b. Some aspects of the problem of mathematical rigor, *Bull. Amer. Math. Soc.* 47, 221-241.
c. A revision of the fundamental rules of combinatory logic, *J. Symbolic Logic* 6, 41-53.
d. The consistency and completeness of the theory of combinators, *J. Symbolic Logic* 6, 54-61.
e. The paradox of Kleene and Rosser, *Transactions Amer. Math. Soc.* 50, 454-516.

1942

a. Mathematical teaching and the national defense, *School Review* 50, 337-346.
b. The combinatory foundations of mathematical logic, *J. Symbolic*

Logic 7, 49-64.
c. The inconsistency of certain formal logics, *J. Symbolic Logic* 7, 115-117.
d. Some advances in the combinatory theory of quantification, *Proc. Nat. Acad. Sci. U. S. A.* 28, 564-569.

1943

The Heaviside operational calculus, *Amer. Math. Monthly* 50, 365-379.

1944

The method of steepest descent for nonlinear minimization problems, *Q. Applied Math.* 2, 258-261.

1946

a. (with Max Lotkin) A study in fourth order interpolation on the ENIAC, *Aberdeen Proving Ground, Ballistic Res. Lab. Rep.* 613.
b. (with Willa Wyatt) A study of inverse interpolation on the ENIAC, *Aberdeen Proving Ground, Ballistic Res. Lab. Rep.* 615.
c. The application of inverse interpolation to the calculation of firing tables for ground gunfire, *Aberdeen Proving Ground, Ballistic Res. Lab. Rep.* (apparently never issued).

1947

(with I. J. Schoenberg) On spline distributions and their limits: the Pólya distribution functions (Abstract), *Bull. Amer. Math. Soc.* 53, 1114.

1948

The logical structure of grammar (mimeographed at University of Chicago from notes of a lecture delivered to the "comparitive literature group" at Penn State; reissued with minor corrections by Penn State in 1959).

1949

a. Languages and formal systems. *In* "Proceedings of the Tenth International Congress of Philosophy (Amsterdam, August 11-18, 1948)", pp. 770-772. North-Holland, Amsterdam.
b. Certain basic theorems on linear differential equations, *Amer. Math. Monthly* 56, 398-402.
c. On the composition of programs for automatic computing, *U. S. Naval Ordnance Lab. Memo* 9805.
d. A simplification of the theory of combinators, *Synthese* 7, 391-399.

1950

a. "A Theory of Formal Deducibility", Notre Dame Mathematical Lectures No. 6, Notre Dame, Indiana. Second Edition 1957.
b. Language, metalanguage, and formal system, *Phil. Rev.* 59, 346-353.
c. A program composition technique as applied to inverse interpolation, *U. S. Naval Ordnance Lab. Memo* 10,337.
d. The inferential theory of negation. *In* "Proceedings of the International Congress of Mathematicians, Cambridge, Mass., U. S. A., Aug. 30 - Sept. 6, 1950", Vol. 1, p. 722.

1951

a. "Outlines of a Formalist Philosophy of Mathematics", North-Holland, Amsterdam.
b. Abstract differential operators and interpolation formulas, *Portugalia Math.* 10, 135-162.
c. L-semantics as a formal system. *In* "Congrès International de Philosophie des Sciences, Paris 1949, II Logique, Actualités Scientifiques et Industrielles 1134", pp. 19-29. Hermann & Cie., Paris.
d. Note on iterations with convergence of higher degree, *Q. Applied Math.* 9, 204-205.
e. Note on a theorem on abstract differential equations, *Portugalia Math.* 10, 23-24.
f. La théorie des combinateurs, *Rend. Mat. e Appl.* (5) 10, 347-359.
g. La logique combinatoire et les antinomies, *Rend. Mat. e Appl.* (5) 10, 360-370.
h. Philosophische Bemerkungen zu einigen Problemen der mathematischen Logik, *Archiv Phil.* 4, 147-156.

1952

a. "Leçons de Logique Algébrique", Gauthier-Villars, Paris and Nauwelaerts, Louvain.
b. The system LD, *J. Symbolic Logic* 17, 35-42.
c. On the definition of negation by a fixed proposition in the inferential calculus, *J. Symbolic Logic* 17, 98-104.
d. The permutability of rules in the classical inferential calculus, *J. Symbolic Logic* 17, 245-248.
e. The elimination theorem when modality is present, *J. Symbolic Logic* 17, 249-265.
f. A new proof of the Church-Rosser Theorem, *Koninkl. Nederl. Akad. Wet. Proc. Ser. A* 55 (= *Indag. Math.* 14), 16-23.
g. On the definition of substitution, replacement, and allied notions in an abstract formal system, *Rev. Phil. Louvain* 50, 251-269.

1953

a. Mathematics, syntactics, and logic, *Mind*, ns 62, 172-173.
b. Les systèmes formels et les langues. *In* "Les Méthodes Formelles en Axiomatique, Paris Décembre 1950", pp. 1-10. Colloques Internationaux du Centre National de la Recherche Scientifique no. 36, Paris.
c. Theory and experience, *Dialectica (Neuchatel)* 7, 176-178.

1954

a. The logic of program composition. *In* "Applications Scientifiques de la Logique Mathématique, Actes du 2e Colloque International de Logique Mathématique, Paris--25-30 Août 1952, Institut Henri Poincaré", pp. 97-102. Collection de Logique Mathématique, Série A, Gauthier-Villars, Paris and Nauwelaerts, Louvain.
b. Generalizations of the deduction theorem. *In* "Proceedings of the International Congress of Mathematicians, Amsterdam, September, 1954", Vol. 2, pp. 399-400.

1955

The inconsistency of the full theory of functionality (Abstract), *J. Symbolic Logic* 20, 91.

1956

The consistency of the theory of functionality (Abstract), *J. Symbolic Logic* 21, 110.

1957

Combinatory Logic. Notes by J. Barkley Rosser. *In* "Summaries of Talks Presented at the Summer Institute of Symbolic Logic at Cornell University, July 1957", pp. 90-99. Mimeographed.

1958

a. (with Robert Feys) "Combinatory Logic, Vol. I", North-Holland, Amsterdam. Reprinted 1968, 1974. Spanish translation, "Lógica Combinatoria" (Trans. Manuel Sacristan), Editorial Tecnos, Madrid, 1967.
b. Calculuses and formal systems, *Dialectica* 12, 249-273. (Festschrift in honor of P. Bernays, also issued separately as "Logica: Studia Paul Bernays Dedicata", Editions du Griffon, La Neauveville, Switzerland).
c. The tectonic property for calculuses. *In* "International Congress of Mathematicians, Edinburgh, 1958; Abstracts of Short Communications and Scientific Program", p. 6 (Listed in Pro-

ceedings of the Congress, p. xvii).
d. On definitions in formal systems, *Logique et Analyse* 1, 105-114.

1959

The interpretation of formalized implication, *Theoria* 25, 1-26.

1960

a. The deduction theorem in the combinatory theory of restricted generality, *Logique et Analyse* 3, 15-39.
b. The inferential approach to logical calculus, *Logique et Analyse* 3, 119-136 and 4 (1961) 5-22.

1961

a. Some logical aspects of grammatical structure. *In* "Structure of Language and its Mathematical Aspects", pp. 56-68. American Math. Society, Proc. Symposia in Applied Math., Vol. 12.
b. Basic verifiability in the combinatory theory of restricted generality. *In* "Essays on the Foundations of Mathematics", pp. 165-189. Hebrew University, Jerusalem.

1962

(with C. J. Ducasse) Early history of the Association for Symbolic Logic, *J. Symbolic Logic* 27, 255-258. Addendum, *Ibid.* 28 (1963), 279.

1963

a. "Foundations of Mathematical Logic", McGraw-Hill, New York. Reprinted by Dover, 1977. Russian translation, "Osnovaniya Matematicheskoj Logika" (Trans. V. V. Donchenko), Izdat. "MIR", Moscow, 1969.
b. Logic as a mathematical science, *Annals Japan Assoc. Phil. Sci.* 2, 131-143.

1964

a. The elimination of variables by regular combinators. *In* "The Critical Approach to Science and Philosophy: Essays in Honor of Karl Popper" (Ed. Mario Bunge), pp. 127-143. The Free Press, Glencoe, Illinois.
b. Combinatory recursive objects of all finite types, *Bull. Amer. Math. Soc.* 70, 814-817.

1965

a. Two examples of algorithms, *Archiv math. Logik* 7, 29-44.

b. The relation of logic to science. *In* "Information and Prediction in Science" (Eds. S. Dockx and P. Bernays), pp. 79-96. Academic Press, New York.

c. Remarks on inferential deduction. *In* "Contributions to Logic and Methodology in Honor of J. M. Bochenski" (Ed. Anna-Teresa Tymieniecka), PP. 45-72. North-Holland, Amsterdam.

1966

(with I. J. Schoenberg) On Pólya frequency functions IV: the fundamental spline functions and their limits, *J. D'Analyse Math.* 17, 71-107.

1967

Logic, combinatory. *In* "The Encyclopedia of Philosophy" (Ed. Paul Edwards), Vol. 4, pp. 504-509. Macmillan and Free Press, New York.

1968

a. A deduction theorem for inferential predicate calculus. *In* "Contributions to Mathematical Logic" (Ed. K. Schütte), pp. 91-108. North-Holland, Amsterdam.

b. The purposes of logical formalization, *Logique et Analyse* 11, 357-366.

c. The equivalence of two definitions of elementary formal system, *Compositio Math.* 20, 13-20. (Festschrift in honor of A. Heyting, edited by D. van Dalen, S. C. Kleene, and A. S. Troelstra, Wolters-Noordhoff, Groningen).

d. Combinatory Logic. *In* "Contemporary Philosophy, A Survey" (Ed. Raymond Klibansky), pp. 295-307. La Nuova Italia Editrice, Florence.

e. Recent advances in combinatory logic, *Bull. Soc. Math. Belg.* 20, 288-298.

f. The achievement of logical formalization. *In* "Akten des XIV Internationalen Kongresses für Philosophie, Wien, 2.-9. September 1968", Vol. 3, pp. 198-201. Herder, Vienna.

g. Rapporten over het 14de Internationaal Congres v. Filosofie, gehouden van 2-9 september 1968 te Wenen, *Rapporten v. d. Koninkl. Nederl. Akad. Wet.* 20, 153-158.

1969

a. The undecidability of λK-conversion. *In* "Foundations of Mathematics: Symposium Papers Commemorating the Sixtieth Birthday of Kurt Gödel" (Eds. J. J. Bulloff, T. C. Holyoke, and S. W. Hahn), pp. 10-14. Springer, Berlin.

b. Modified basic functionality in combinatory logic, *Dialectica* 23, 83-92.

1972

(with J. R. Hindley and J. P. Seldin) "Combinatory Logic, Vol. II", North-Holland, Amsterdam.

1973

The consistency of a system of combinatory restricted generality, *J. Symbolic Logic* 38, 489-492.

1975

a. A study of generalized standardization in combinatory logic. *In* "ISILC 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" (Eds. J. Diller and G. H. Müller), pp. 44-55. Springer, Berlin (Springer Lecture Notes in Math., Vol. 500).

b. Representation of Markov algorithms by combinators. *In* "The Logical Enterprise" (Festschrift in honor of F. B. Fitch, Eds. A. R. Anderson, R. B. Marcus, and R. M. Martin), pp. 109-119. Yale University Press, New Haven, Connecticut.

1979

On a polynomial representation of λβ-normal forms. *In* "Konstruktionen versus Positionen: Beiträge zur Diskussion um die Konstruktive Wissenschaftstheorie" (Festschrift in honor of P. Lorenzen, Ed. Kuno Lorenz), pp. 94-98. Walter de Gruyter, Berlin.

1980

Some philosophical aspects of combinatory logic. *In* "The Kleene Symposium: A Collection of Papers on Recursion Theory and Intuitionism" (Eds. John Barwise, H. Jerome Keisler, and Kenneth Kunen), pp. 85-101. North-Holland, Amsterdam.

TABULA GRATULATORIA

Greetings and best wishes to Haskell Curry on the occasion of his 80th birthday, from:

Peter Aczel
John M. Anderson
Peter B. Andrews
Paul Axt
Raymond and Christine Ayoub
Henk Barendregt
James B. Bartoo
Nuel D. Belnap, Jr.
Choukri Ben-Yelles
Jack Biesterfeldt
Corrado Böhm
William W. Boone
N. G. de Bruijn
Martin Bunder
André Chauvin
Edward J. Cogan
Mario Coppo
John Corcoran
Newton C. A. da Costa
William Craig
Karel Culik
Diederik van Daalen
Dirk van Dalen
Mariangiola Dezani-Ciancaglini
Justus Diller
H. C. Doets
Solomon Feferman
Frederic B. Fitch
Orrin and Aline H. Frink
Robin O. Gandy
Harvey Gerber
Paul Gilmore
Jean-Yves Girard
Hans and Jeanette Gloor-Görtz
Thomas Grilliot
Ronald Harrop
John Heintz
Leon Henkin
Arend Heyting
Shigeru Hinata
Roger Hindley
Henry Hiż
William A. Howard
J. J. de Iongh
Kiyoshi Iscki
David Isles
Thomas Jech
Henry W. Johnstone, Jr.
John T. Kearns
Stephen C. Kleene

Jan Willem Klop
Karst Koymans
Erik C. W. Krabbe
Georg Kreisel
Saul A. Kripke
Alexander S. Kuzichev
Joachim Lambek
Bruce Lercher
John A. Lever
Jean-Jacques Lévy
M. H. Löb
Giuseppe Longo
Paul Lorenzen
Mary McCammon
Richard Mansfield
Richard M. Martin
David Meredith
Robert Meyer
Gerd Mitschke
J. J. A. Mooij
Gert H. Müller
Louis Nolin
Katuzi Ono
Charles Parsons
Gordon Plotkin
William L. van der Poel
Garrel Pottinger
Marian Boykan Pour-El
Robert Price
Willard V. Quine
Helena Rasiowa
Hugo and Pilar Ribeiro
Gene F. Rose
J. Barkley Rosser
Luis E. Sanchis
Kurt Schütte
Dana S. Scott
Jonathan P. Seldin
Alan L. Selman
L. V. Shabunin
Kajetan Šeper
Stephen G. Simpson
John Staples
Richard Statman
Manfred E. Szabo
Moto-o Takahashi
Gaisi Takeuti
Alfred Tarski
Richard J. Thompson
B. A. Trakhtenbrot
Anne S. Troelstra
Daniel Vanderveken
Betty Venneri
Marisa Venturini-Zilli
Albert Visser
Christopher Wadsworth
Peter Wesly
Mariko Yasugi

CONTENTS

CURRY'S PROGRAM AND PHILOSOPHY

CURRY'S PROGRAM

Jonathan P. Seldin

Mathematics Department
Concordia University
Loyola Campus
Montreal, Québec H4B 1R6
Canada

Dedicated to H. B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

The expression "Curry's Program" first arose as the result of a search for a good title for a seminar talk.[1] Curry has never used it or anything like it. Nevertheless, the expression makes sense: there is a single theme which has been the focus of most of Curry's mathematical work.[2] It is the purpose of this paper to explain that theme.

Of course it is possible to say simply that this theme is combinatory logic. But this is not really sufficient. There has been considerable misunderstanding of his basic aim and approach. Part of the reason for this seems to be that much of what he has written to explain this basic aim and approach has appeared in relatively inaccessible places, at least when compared with his best known philosophical writings. It therefore appears that his ideas may be clarified by considering what he says in detail in some of his less well known works. This is especially true of some of his earliest papers, for although his ideas have changed considerably since then, it may well help to clarify his later ideas to examine the earlier ideas from which they arose. It will be convenient to divide the history of these ideas into the following three periods: 1) 1926-1934, 2) 1935-1942, and 3) after 1945.

2. THE FIRST PERIOD: 1926-1934

As Curry has often explained, he first became interested in combinators as a result of reading Whitehead and Russell (1910).[3] In an attempt to divide the rule of substitution (for propositional calculus) into simpler rules, each of which was to be as simple as the rule of modus ponens, Curry discovered some combinators for himself before he was introduced to Schönfinkel (1924). Curry then set out to give the formal axioms and rules which Schönfinkel had not given in order to use combinators to provide the desired foundation for logic. Exactly what he meant by this at the time is perhaps best made clear by considering what he wrote in the first part of his first paper (1929):[4]

> Mathematical Logic has been defined as an application of the formal methods of mathematics to the domain of logic[5]. Logic, on the other hand, is the analysis and criticism of thought.[6] In accordance with these definitions, the essential purpose of mathematical logic is the construction of an abstract (or strictly formalized) theory, such that when its fundamental notions are properly interpreted, there ensues an analysis of those universal principles in accordance with which valid thinking goes on. The term analysis here means that a certain rather complicated body of knowledge is exhibited as derived from a much simpler body assumed at the beginning. Evidently the simpler this initial knowledge, and the more explicitly and carefully it is set forth, the more profound and satisfactory is the analysis concerned.
>
> In the present paper I propose to take some preliminary steps toward a theory of logic in which the assumed initial knowledge is simpler than in any existing theory with which I am acquainted. Before this is done, however, it is necessary to consider somewhat in detail what is meant by the phrase "abstract theory," and what is the significance of such a theory for the analysis of thought. The object of this discussion is to see just how the assumed knowledge enters into the theory; for this purpose we shall need to be explicit, even at the risk of repeating what has already been better said by others.
>
> Certain ideas concerning the nature of an abstract theory can be disposed of at once. In the first place the naive notion that such a theory consists of a set

of primitive ideas and propositions together with their consequences by the laws of pure logic, must be dismissed on the ground of its circularity. Again it is said that an abstract theory is one from which all meaning has been abstracted. This requires that the sense of the term meaning be explained. If we take the term meaning, as applied to objects, to signify the totality of properties (of those objects) which are directly apprehensible to our intuition, then every object presented to the mind has meaning, and a meaningless theory is a contradiction in terms. Even a symbol cannot be meaningless in this sense; for either it denotes some object, or else it is itself the object, and so has meaning. If we use the word meaning in some other sense, then it loses its significance as related to the assumed initial knowledge of our theory. Consequently the idea of a meaningless theory must be subjected to further scrutiny.

Let us use the word meaning, as applied to concepts, in the sense of the preceding paragraph. Then, relative to a given theory, we may distinguish two kinds of meanings, which we shall call natural and conventional meanings respectively. Natural meanings are those which are comprehensible a priori in terms of our previous knowledge; conventional meanings those based on relations to the theory itself. Natural meanings we may further subdivide into essential and accidental; essential meanings are those on which the deduction of the theory depends; accidental meanings those which are non-essential. The distinction between these three kinds of meaning is important in what follows.

The distinction between natural and conventional meanings has a counterpart in that between statements of fact and statements of convention. By a statement of fact I mean something of which truth or falsehood can be significantly predicted; by a statement of convention a declaration of intention, definition or the like. The former corresponds to an act of judgement, the latter to one of volition. Common sense and grammar have long recognized both of these types; yet logicians seem to belittle the latter in that they define the proposition so as to exclude it.[7] Both these kinds of statements, however, are equally intelligible to a rational mind; in this sense it is false to say that one of them is less significant than the other. As examples of statements of convention we have of course the definitions of technical terms; but not all statements of convention are verbal--for instance the rules of chess, which, by a sufficient amount of circumlocution, may be stated without defining any new terms whatever. The postulates of any branch of mathematics are of this character.

Let us now return to the abstract theory. I suggest that such a theory is characterized by the following: 1) the explicit indication of all essential meanings; 2) the absence, or at least omission from consideration, of accidental meanings; 3) the circumstance that the statements with which the theory begins are conventional, and are, furthermore, sufficiently detailed so that all the acts necessary to the deduction are specified.

To be yet more precise, an abstract theory begins with a set of primitive notions, which, taken collectively, we shall call the *primitive frame*, as follows:

I. Non-formal Primitive Ideas.[8]

A set of ideas to each of which a certain amount of essential meaning is attached, although they need not coincide with any ideas previously entertained.[9] For example:

1. Entities. In order for an object to be considered in the theory at all, it must have some property; this fact we may express by saying it is an entity of one sort or another. These properties must then be among the primitive ideas of the theory; and they must have essential meaning in that they are predicates. In the simplified theory only one such notion is necessary; but in the more complicated ones there are several; e.g. in the Principia Mathematica there are individual, proposition, function, etc., the latter two of various orders and types.

2. Modes of combination. I.e. processes by means of which entities may be combined to get new entities. These have essential meaning in that they are combinations. It must be specified by rules that the results of combination are entities. In the simple cases only one such notion is necessary, and that a dyadic one; in the more complicated cases the various processes of substitution are of this nature.

3. Assertions. An assertion is a kind of entity, which is of special importance because the object of deduction is to derive new assertions. The idea of assertion has essential meaning only in that it is a predicate applicable to certain entities. Ordinarily an assertion is interpreted as a statement to which belief attaches, but this meaning is accidental.

II. Formal Primitive Ideas.

Ideas which have no essential meanings (except that they are concepts). They must of course be entities and their relations to other parts of the primitive frame will give them conventional meanings.

III. Postulates.

All propositions of the theory are statements that certain particular entities are assertions; the postulates are the propositions, if any, which are assumed at the beginning. They are purely conventional.

IV. Rules.

Statements of the processes by means of which new entities[10] or new propositions may be constructed. Such statements are of course conventional; moreover they are universal statements (involving the notion of "every" or its equivalent[11]). They thus differ from propositions not only in that they involve intuitive ideas from which the propositions are free, but also in that they form the methods of transition, rather than the stopping places in the theory. A typical example is the "rule of inference" which may be stated thus: whenever p and p $\supset$ q are assertions, then q shall also be an assertion.

In addition to the above notions there are yet to be considered those associated with the use of symbolism. Whether these are to be regarded as a part of the theory or as something superposed upon it, is a question which I prefer to leave to the reader to adopt such views as seem best to him. However he may decide, certainly language is necessary in order that the theory may be communicated. The use of this language may involve intuitive operations other than those we have mentioned; it is desirable that these, too, be specified by rules; because otherwise it is not certain that intuitive knowledge, other than that expressly mentioned, does not creep into the theory. We shall call such rules *symbolic conventions.*

So much for the primitive frame. The abstract theory itself may now be defined as the doctrine built upon such a primitive frame by means of the following processes: 1) the derivation of new propositions, each of which is of the form that such and such an entity is an assertion, by means of the rules; 2) the addition of new ideas by definitions. The latter process may be regarded as a symbolic matter, governed by the symbolic conventions, or as the introduction of a new idea along with postulates and rules to the effect that it is identical with some already existing entity. ...

The importance of such a theory for the analysis of thought lies in the definiteness with which the intuitive knowledge entering into it is set forth. Indeed, so far as the abstract theory itself is concerned, the only knowledge assumed is the appreciation of the essential meanings and conventional statements appearing in the primitive frame. When the theory is interpreted the additional knowledge that must be brought to bear consists of the following: that the concepts which we substitute for the primitive ideas have the necessary essential meanings, and that the conventional statements in the primitive frame correspond to facts. In both cases the required information is precisely specified.

On the philosophic nature of such a theory, its relations to the symbolism used in its expression, and to the various concrete theories obtained by interpreting it, it suffices to say that such questions are largely metaphysical, and therefore irrelevant to the present discussion. It is by no means self-evident that the best interests of science are served by adopting any one theory to the exclusion of all others; any more than it is desirable that two persons following the same argument should have the same mental imagery.[12]

The next point ... is the cardinal importance of the rules in any abstract theory related to logic. For the amount of initial knowledge which enters into the first three categories of the primitive frame is slight. In the rules, however, such knowledge is involved in every step of the construction; for we have to pass judgment as to whether the contemplated act is or is not according to Hoyle. These judgments, moreover, are the only ones which are required. The rules, therefore, form the port of entry of intelligence; and since nothing can be done without them, they represent the atoms of thought, so to speak, into which the reasoning can be decomposed. It follows that in constructing such a theory it is not sufficient to reduce the postulates and primitive ideas to their lowest terms; it is even more important to so choose the rules that they involve, in their application, only the simplest actions of the human mind.

Now although the rule of inference, stated above, is simple enough, yet in all current mathematical logics there exist rules which are highly complex. The presence of these complex rules raises the question whether it is possible to formulate a theory which is--1) adequate for the whole of logic, 2) based on a finite number of primitive ideas, postulates, and rules, the last of the same order of complexity as the rule of

> inference. I believe that it is;[13] indeed steps in that direction have already been taken.[14] As a preliminary to treating this general problem, I shall discuss in the rest of this paper a special one connected with it; viz., the analysis of the process of substitution. The latter process is one of those complicated rules which occur in practically every logical theory today.
>
> The reader will observe that in the theory which results from the analysis the formulas are more complicated, and the deductions required to produce them more lengthy, than would be the case in the older theory. ... Consequently we must adopt a point of view suggested by Hilbert. With each theory there is associated a metatheory in which we reason intuitively about the theory. In this metatheory we can derive more and more complicated rules by showing, in general terms, how any particular consequence of the derived rules can actually be deduced from the primitive ones. The aim of mathematical logic is, in fact, not to reduce mathematics to a formalism, à la *Principia*, in which all steps explicitly appear; but rather to analyze logic with a view to obtaining a greater command over its use and a more profound understanding of its nature.

Curry then discusses the way in which the substitution process is a complicated one. He points out, for example, that the result of substituting a for x and then b for y in an expression leads to the same result as substituting b for y and then a for x, but the two substitution processes are clearly distinct. He then relates substitution to transformation (changing the order of the arguments) of functions. He then continues:

> An important step toward the analysis of this situation was made by M. Schönfinkel.[15] Starting, apparently, from the fact that every logical formula is a combination of constants--the variables being only apparent--he shows that neither the notions of propositional function (of various orders) nor that of substituion need be assumed as primitive; his formulation of logic is such that variables, real or apparent, do not appear explicitly. His primitive frame is essentially as follows:[16]
>
> I. Non-formal Primitive Ideas.
>
> 1. *Entity*--not mentioned by Schönfinkel, but to be

understood essentially as a single notion of the sort mentioned in the general description of an abstract theory above.

2. *Application*--a mode of combination, the only one in the theory. Two entities x and y combine to give a third entity called the application of x to y and denoted by (xy). The *interpretation* of this is as follows: if x is a function, then (xy) is the result of substituting y for the first variable in x; thus if f denote a function of one variable, (fx) denotes what is ordinarily written f(x), if f is a function of two variables ((fx)y) denotes what is ordinarily written f(x,y), etc. Nothing is said concerning the interpretation of (xy) when x is not a function; if the reader is disturbed over this lack, he may invent one arbitrarily, e.g. (xy) may then be equal to x.

3. *Assertion*--to be understood as in the general description of an abstract theory. Not denoted by any particular symbol; but when a symbol of the form (((=)x)y) (or x = y) where x and y may be quite complicated, stands out by itself like an equation in algebra, then the proposition that the corresponding entity is an assertion is to be understood.

II. Formal Primitive Ideas.

Three, denoted by (=), S and K. In the interpretation (=) is to correspond with identity; S and K are operations in the sense defined by the rules.

Symbolic Conventions.

1. If x and y are any entities whatever, then instead of (((=)x)y) we may write (x = y).

2. If $x_1, x_2, \ldots, x_n$ are any entities, then instead of

$$((\ldots(((x_1x_2)x_3)x_4)\ldots)x_n)$$

we may write $(x_1x_2x_3\ldots x_n)$.

3. The outside parentheses may be left off in the case of a symbol standing by itself or on either side of the sign =.

III. Postulates.

None.

IV. Rules.

0. If x and y are entities, then (xy) shall be an entity.

1. (=) shall have the properties of identity. These properties may be specified by a few simple rules; but in this treatment we shall not go into detail. We shall treat (=) as if it were precisely the intuitive relation of equality.

2. If x and y are any entities, then
$$Kxy = x$$
3. If x, y, z are entities, then
$$Sxyz = xz(yz)$$
4. If X and Y are combinations of S and K, and if there exists an integer n such that by application of the preceding rules we can formally reduce the expressions $Xx_1x_2\ldots x_n$ and $Yx_1x_2\ldots x_n$ to combinations of $x_1x_2\ldots x_n$ which have the same structure, then X = Y.

If the above primitive frame were a part of a general theory of logic, the term entity would include not only the various combinations of S and K, but all the notions of logic as well.

Several things should now be clear. First, Curry had, by 1929, already formed, at least in a preliminary way, the philosophy of mathematics which he was later to call formalism and for which he was to become known, and this philosophy was closely related to his aims and assumptions at the beginning of his work on combinatory logic. Second, combinatory logic was originally intended to serve as a foundation for all possible formal systems (Curry called them formal theories at this stage) for mathematics and logic, and so Curry saw his work in much the same way as Church must have seen his (1932), namely as a continuation of the analysis of Frege and of Whitehead and Russell, but with a more sophisticated goal.

Although the primitive frame of Curry (1929) did not include any "notions of logic", that of his dissertation (1930) did. In addition to the basic combinators (there taken to be B, C, W, and K) and Q (in place of (=)), it also contains atomic terms for the universal quantifier, implication, and conjunction. In addition to the axioms and rules of pure combinatory logic, it includes an axiom expressing the reflexive law of equality,[17] the (natural deduction) elimination rules for the universal quantifier and implication, the (natural deduction) introduction rule for conjunction,[18] and the following rules for Q:

$$X,\ QXY \vdash Y,$$
$$QXY \vdash Q(ZX)(ZY).$$[19]

These "illative" notions, as he later called them, did not play much of a role in Curry (1930), but their presence is a clear indication of his intentions.

For further insight into his intentions, we may turn to the introductory and philosophical remarks of Curry (1930). Curry begins (§I A) by discussing the complexity of the theory that must be presupposed for any (then) existing (formal) theory for mathematics and logic. This theory must include propositions and propositional functions, various modes of combination for generating new objects from objects already present in the system, and substitution processes. Some of the properties of these kinds of things which are needed in this presupposed theory are quite complex, especially with regard to substitution. Curry calls this theory the *prelogic* (Urlogik)[20] and says about it the following:[21]

> In spite of the fundamental character of this prelogic for any existing mathematical theory of logic, some important problems have their essential origin in it. I consider here the following two:
>
> 1) The simplification of foundations. The nature of this problem appears to me to be that one wants to construct all mathematics and logic on a minimum of primitive knowledge; or more exactly one wants to analyze them into their elements in order to express this primitive knowledge with the sharpest clarity and distinctness. For this reason I suppose everybody tries to reduce the number of primitive concepts and axioms; and certainly some logicians have gone so far in this direction that they do not shrink from sacrificing generality, completeness, and exactness.[22] However, the number of primitive concepts and axioms of this kind is a consideration of little importance for logic. For what difference does it really make that we remove two or three primitive concepts when infinitely many primitive concepts and rules, i.e. not formal, are already present in the prelogic? The simplification of foundations should therefore begin with the simplification of the prelogic.
>
> 2) The elimination of the paradoxes. In order to see the relationship of these paradoxes to the prelogic, we consider e.g. the Russell Paradox, which can be stated in the following way: let F be that

> property of properties ϕ for which
> $$F(\phi) = \text{not } \phi(\phi);$$
> then $$F(F) = \text{not } F(F).$$
> If one maintains that F(F) is a proposition, where a proposition is defined as something that is either true or false, then one has a contradiction. But we can certainly avoid the paradox by denying that F(F) belongs to the category of propositions or F itself to that of properties. Just here we run into a theorem of the prelogic. And certainly it is these paradoxes that force us to adopt a more complex prelogic than would otherwise be conceivable. It may be, therefore, that a deeper study of the prelogic will spread light in this dark territory of the paradoxes.
>
> These particulars have suggested to me the treatment of this prelogic mathematically. I.e. more exactly: the construction of an abstract formal theory, which will be based on a very simple prelogic and through which the questions that one is accustomed to place in the usual intuitive prelogic can be answered by symbolic reasoning. In this work I intend to formulate such a theory. Because the respective questions have a clearly combinatory character, I call the theory *combinatory logic.*[23]
>
> This combinatory logic will be capable of serving as a foundation for an abstract theory of all logic and mathematics, including functions (predicates, relations) or arbitrarily many variables. In fact I am convinced that such a theory can be constructed by the addition of finitely many formal primitive concepts and axioms to the primitive frame (Grundgerüst) given below. Because in a theory so arranged the primitive frame has overall only finitely many constituents, and because further the rules have only about the same degree of complexity as the well known rule of inference,[24] clear progress is attained concerning the first problem stated above.[25]

It should now be clear that Curry had by the time he wrote this worked out many of the ideas that he would later apply to illative combinatory logic. The idea, expressed in his discussion of the second of the two important problems, the elimination of the paradoxes, that the paradoxes are caused by assuming that entities are propositions when they are not propositions is very important in his later work, as we shall see below.

A bit later on, in §I B, Curry made some remarks which include his first published argument for his preference for dealing with the paradoxes by means of the theoretical postulates of the system rather than by excluding from the system the terms which represent them, a preference which has been a major theme of his work ever since:

> §1. *Meaningless concepts.* In ordinary logic certain objects (concepts, things) occur that are usually called meaningless. I now ask, just what does this mean? One may well maintain that a word or a symbol is not yet defined in respect to a language. E.g., the word "Kuh" is meaningless in English, although incidentally meaningful in German. But in the present cases it is not the words but the concepts which are said to be meaningless. E.g., Whitehead and Russell (and others as well) maintain that $\phi(\phi)$ is meaningless for every ϕ. To say that this claim concerns only the symbol $\phi(\phi)$ is to beg the question; first because there is yet something in thought that the symbol $\phi(\phi)$ can mean according to convention, and second because the basis,--namely the "Vicious Circle Principle" etc.--, for the claim has nothing to do with the symbols. The meaninglessness seems to have an objective content, but what exactly is not clear.
>
> Above all there is a sense in which anything thinkable has a meaning, namely as a concept. Here a concept is to be understood as anything that can be identified or differentiated from other things.[26] Then it is absolute nonsense to say that something does not exist as a concept; for before one can understand such a sentence one must have already understood the thing as a concept. Therefore even the "meaningless" objects are concepts and as such have a meaning.
>
> But among the concepts are some which are "contradictory in themselves." Such are the above mentioned F(F), the greatest cardinal number, the least undefined ordinal number, the round square, etc. These concepts are considered meaningless because of the contradictions. But the contradictions do not lie in the concepts themselves, but in the properties associated with them: e.g. the above mentioned F(F) leads to a contradiction only if one maintains that it is a proposition; the greatest cardinal number only if it really is a cardinal number, etc.
>
> Now let us consider meaningless objects in general. Do not the same thoughts apply to them also? Yes, the meaninglessness of these concepts consists only in the existence of properties which do not hold of them.

And certainly I may say more exactly that they do not belong to the usual categories. These categories are in fact presupposed as somewhat contensive[26a] (inhaltlich) primitive concepts, and nothing is considered which does not belong to them. Naturally, concepts that have properties incompatible with the essence of these categories must be excluded from the theory as "meaningless". But just in this exclusion stands a flaw. The objective of logic is the explanation of thought; if there are thoughts excluded from the explanation, then it is deficient. Furthermore, it is exactly these meaningless concepts which lead to paradoxes; if one excludes them, one can avoid the paradoxes but not explain them. That something is a concept is the only requisite for its being subject to treatment in logic.

The category, concept--or, as I shall call it from here on in order to avoid certain extraneous meanings, entity[27] (Etwas),--is therefore in general the fundamental category of logic. This category is a concept, and its mere consideration is in itself consistent. For this reason I have postulated it as the fundamental category of the theory. From this follows an important consequence: I need no longer worry, at least insofar as it concerns the introduction of new objects into the theory, about the domains of the functions. E.g., if I postulate Schönfinkel's function concept as the fundamental mode of combination--and Schönfinkel has already shown that this alone is necessary--, then the combination of any two concepts, such as f and x into fx, is again a concept. E.g. the combination of the King of France with the proposition "the moon is made out of green cheese" is a concept, because it can be identified and differentiated from other things. Among concepts as given here, some will be "meaningful" and others "meaningless"; the main aim of combinatory logic is to distinguish these two kinds.

. . .

§3. *Absolute universals.* In number 1 I asserted that the concept "entity" may be assumed as a primitive concept. I now assert further that there are properties that hold for every entity. Against the possibility of such an assertion stands the well known prohibition on non-predicative concept formation. This prohibition has in general a somewhat pragmatic character, and thus can only be refuted by the actual construction of a theory contradicting the prohibition. In the "Principia Mathematica" this principle is defended a priori in the form of the "Vicious Circle Principle." The argument given

> there is essentially this: one can assert nothing about all propositions because this creates a new proposition, and so there is no determined collection of propositions that forms the range of the asserted properties. One might just as well argue as follows: one can assert nothing about all oranges because every year new oranges are created, and so there is no determined collection of oranges etc., etc. In fact we understand general sentences (Urtiele) not in extension but in intension--there is something in the essence of the property concerned by which we can conclude that it holds for every entity. This conclusion presupposes no collection. Thus this objection against the possibility of absolute universal lacks a theoretical foundation. Consequently I shall make the simpler and more natural presupposition that there are such universals; in particular, that such properties as the reflexive law of equality hold for every entity.

As I mentioned above, the idea that the paradoxes are caused by assuming that entities are propositions when they are not propositions is very important for Curry's later work. He elaborated further on this idea in another of his early papers, (1936a), which, although not published until 1936 was actually written in 1932. This paper opens as follows:

> The field of combinatory logic as hitherto conceived falls naturally into two main divisions. The first of these concerns the processes of logical combination as such; the second deals with the fundamental categories of logic, and the grounds from which we conclude that various entities belong to them. ...
>
> These two divisions of the field differ not only in subject matter but also in aim. Two principal problems were stated in my thesis as motivating the entire investigation; viz. 1) the simplification of the foundations, and 2) the avoidance of contradictions. The first phase of combinatory logic has a direct relation only to the first of these problems; on the other hand the second phase, although it also is of significance in connection with the first problem, derives its primary interest from its bearing on the second. Indeed I suspect that these contradictions arise from the application of a formal calculus, valid for a certain logical category, to entities which lie outside that category; and that

> if we have a properly constructed formal system for proving that entities belong to that category, then a fallacy will show itself when we attempt to apply this method of proof in a case involving contradictions. If this surmise is substantiated, then the second phase of combinatory logic is more directly concerned with these paradoxes than any other subject whatsoever. ...
>
> The opening wedge in this inquiry is the concept of function. Without concerning ourselves, at first, with what the categories of logic shall be, we may investigate in abstracto the processes by which we infer the category to which a given combination of entities belongs, when those to which the original entities belong are known. This amounts to studying the machinery by which we deal with categories derived from known categories by means of the notion of function, for if the category of a given combination is determinate, it will be because the combination is constructed by substituting in certain functions entities which combine with those functions to given other entities of determinate character. Thus, to take a simple example, suppose X is a function from α to β (α and β being categories) and Y belongs to α, then (XY) belongs to β; or again if X is a function from β to γ and Y is a function from α to β, then BXY is a function from α to γ. Such properties as these are capable of abstract formulation; and such a formulation forms an appropriate foundation for the second phase of combinatory logic as a whole.

Curry then gives an informal explanation of the new atomic term F, for which FXYZ is intended to be interpreted as saying that Z is a function from X to Y. He also discusses the possibility of defining F in terms of implication and the universal quantifier. He then says that in the last section of the paper, on the Russell Paradox,

> some further new entities, including negation (N) and proposition (Pr) are introduced tentatively for the purpose of dealing with the Russell paradox, which may be stated as follows: Let X be the property of properties defined by
>
> $$X \equiv [f]N(ff)$$
>
> (i.e. let X be the property of functions f, such that Xf $\equiv$ not ff); then

$$\vdash XX = N(XX),$$

> and hence if XX is true it must be false, and conversely. Now everything which is formal in the above statement appears to me to be perfectly valid; the entity XX is defined--it is, in a notation free from variables, the entity W(BN)--, and the entity XX does have the property of being equal to its own negative. But it does not follow that XX is a proposition, (where a proposition is thought of as whatever is either true or false), and until some proof of this supposition is offered, a paradox does not exist. ...
>
> In regard to the discussions of the contradictions it may be advisable to emphasize that I have neither proved nor attempted to prove that a contradiction is impossible. I have simply examined certain particular, more or less plausible, arguments, and have tried to show that, if the assumptions of logic are properly chosen, those arguments are invalid.

At this point Curry introduces the following long footnote:

> That a proof of absolute consistency is after all a secondary matter, is an opinion toward which I am inclined by the following considerations. The primary importance of consistency proofs in the minds of Hilbert and his followers is a consequence of the fact that these writers postulate the a priori certainty of a certain portion of our logical machinery, and believe that if we can show, by means of this portion, that a more extended apparatus is consistent, that the latter is, in a sense, a priori justified. But the existence of such an a priori certainty is a hypothesis; without this hypothesis the cardinal importance of consistency proofs falls to the ground. This is not, of course, the place to discuss the merits of such a hypothesis; but since the current altercation over the foundations of mathematics resolves itself largely into a dispute as to what is a priori certain and what is not, it is pertinent to urge that mathematics can and should be so conceived, as to be independent of what views one may hold regarding the nature of what is a priori known. For we can think of any mathematical theorem as stating that a certain conclusion is a "logical consequence" of such and such premises; and the term "logical consequence" can be defined objectively by means of a formal theory such as the present one, presupposing nothing except the ability to identify

> the symbols and to understand and apply the rules. For such a theory, then, the ultimate criterion of acceptibility should be neither whether it can be shown, on the basis of a more restricted definition, to be consistent, nor whether its individual propositions conform to an "aussersprachliche Intuition" or a "Wesensanschauung", but whether it is as a whole so constituted that, by thinking in accordance with it, we can react advantageously to the environment in which we live. Thus the truths of logic become, in common with all other general truths, hypotheses which we accept so long as they are workable, but which we freely discard or amend as soon as they become unsatisfactory,--whether through the discovery of an internal contradiction or for some other reason. Such a view does not, it should be added, amount to a denial, on a priori grounds, that a priori truths are possible,--for that would be self contradictory--; nor does it imply that investigations motivated by this or that prioristic philosophy are devoid of value;--for the opposite is true. But such investigations are of a secondary nature; --useful additions to our knowledge, but not exclusively decisive in the fundamental question of what a logical consequence is. ...

The main text then continues:

> The notion of functionality is related to Zermelo's notion of definiteness, which has been lately subjected to criticism by Fraenkel, von Neumann, and others. However a property X can be defined as definite if, for example,
>
> $$\vdash \mathrm{FMPrX},$$
>
> is true, where M stands for Menge, and Pr for proposition. If the theory of abstract sets (Mengenlehre) be founded on combinatory logic as a basis then the objections to Zermelo's definiteness will probably not apply. ... the investigations of Fraenkel and von Neumann do, in fact, belong to the second phase of combinatory logic, which is, therefore, older than the first phase.

Also of interest here are some of the concluding remarks of Curry (1934b), which was written when it appeared that Curry (1936a) would not appear in print for some time.

> 1. Since the preceding theory does not involve, except in illustrations, any special assumptions as

> to what the categories of logic shall be, it is not tied to any particular logical theory, new or old, but can serve as foundation for a great variety of them. Thus it is compatible with the engere Funktionenkalkül ..., with the Principia Mathematica (when suitably modified), with Zermelo's Theory of abstract sets ..., or with Heyting's quasi-intuitionistic theories ... . In fact it would seem to be fundamental to any theory in which distinction of categories is made at all. I am in general agreement with those who believe that any satisfactory logical theory must make such a distinction. ...
>
> 3. It has been objected that there is no assurance that this theory does not lead to other contradictions. True. However, the theory is no different from any other logical theory in that respect; indeed it is questionable if we shall ever have such assurance, other than that derived from empirical considerations, for any logical theory sufficiently powerful to be of any use. There is, therefore, little to be gained by adopting a Fabian policy in regard to these contradictions. Our friends the physicists have ceased to search for a theory of which they could be sure beforehand that it would explain the universe. We shall do better if we likewise make bold hypotheses which can be modified later, as further research shows their inadequacy.[28]

In addition to making clear that Curry never proposed to abolish completely all type distinctions, these quotations show that for Curry, what he later came to call illative combinatory logic was to be considered as broader than theories of combinators as such. This lends support to a proposal by John Kearns (1980) to distinguish between combinatory logic (which he takes to be essentially about what he calls "fully explicit deductive systems") and theories of combinators. Curry brought combinators into the picture only because he thought it would be useful to use them as a tool to make systems for logic fully explicit. Furthermore, from the beginning he felt that any justification of their use would have to be essentially pragmatic.

With this set of basic objectives, Curry developed the system of combinatory logic that he had begun in his (1929) and (1930) in the papers (1931), (1932), (1933), (1934a), (1934b), (1934c), and (1936a). But this line of development was brought to a halt in 1934 by Kleene and Rosser, who found a contradiction in the systems of Church (1932) and Curry (1934a) and published this in Kleene and Rosser (1935).[29]

3. THE SECOND PERIOD: 1935–1942

After the discovery of the paradox of Kleene and Rosser, Curry set out to "study the paradox thoroughly so as to lay bare its central nerve"[30] before proceeding. At first he thought the problem was one or more of the axioms and rules for the universal quantifier, but the result of his study[31] convinced him that the problem was the incompatibility between what he calls *combinatory completeness* (the possibility of defining [x]X for any variable x and term X) and the unrestricted deduction theorem for implication.[32]

Most people react to this incompatibility by giving up combinatory completeness, or, in other words, by giving up the idea of using combinatory logic as a basis for logic in the ordinary sense. This seems to have been the reaction of Church and his followers, who abstracted the pure lambda-calculus from the system of Church (1932). As Stenlund (1972) says (p. 72), "Therefore, like the Kleene-Rosser paradox, it appears that Curry's paradox is an argument against the requirement of combinatorial completeness with unrestricted self-application." Or, as Dana Scott (1975) puts it even more strongly (p. 1), referring to the original systems of Church and Curry as being in a "Fregian paradise of 'type-free' functions", "Alas, the first workers hardly had time to savour the forbidden fruit before they were turned out of this paradise by the discovery of the paradoxes. A cruel fate, but in retrospect, it is not a very surprising one."

But this was not Curry's reaction. For to Curry giving up combinatory completeness would be giving up on the kind of analysis of the prelogic (as defined in the above quoted part of Curry (1930) § I A) that he wanted to carry out. Furthermore, it would impose restrictions on the formation rules that Curry believed should be part of the theoretical rules of the system (see the above quotes from Curry (1930) § I B). And finally, Curry had never been committed to the idea that the system he proposed in (1934a) was consistent. As the above quotes from (1936a) and (1934b) make clear, he was already prepared for the possibility of a contradiction and had a plan for dealing with it. Curry's purpose in studying the paradox of Kleene and Rosser was to enable him to find the appropriate axioms and rules so that the deduction theorem would be suitably restricted to terms representing propositions. The intuitions that he (and others) had concerning combinators and lambda-terms were not (yet) sufficiently well developed for them to be relied upon in choosing postulates. For this pragmatic reason, it had now become desirable to find consistency proofs whenever possible. Pure combinatory logic was already proved consistent in Curry (1930). The pure lambda-calculus was proved consistent by a different method in Church and Rosser (1936). Here was a secure basis on which to build. Any new systems proved consistent would extend that basis.

After Curry completed his study of the paradox, which resulted in the publication of Curry (1941e), he worked out a plan for using proof theoretical techniques due to Gentzen to obtain consistency proofs for various kinds of systems of illative combinatory logic. These plans were set out in Curry (1942b) and (1942d). But before he could carry out this plan, his work was interrupted again, this time by World War II.[33]

4. THIRD PERIOD: AFTER 1945

After the interruption caused by the war, Curry never quite got back to the point at which he had left his research. He had done applied mathematics during the war[34], and his first major project in logic thereafter was the series of lectures at Notre Dame in April, 1948 which lead to Curry (1950a). Then, while he was attending the Tenth International Congress of Philosophy in Amsterdam in August, 1948, he received the suggestion from a representative of the North-Holland Publishing Company that he write a book on combinatory logic. He thus immediately set to work on the project that led to Curry and Feys (1958a) and Curry, Hindley and Seldin (1972). This project required considerable attention to exposition, and so he began his collaboration with Robert Feys. It also required considerable work on pure combinatory logic. So the only work he did between 1945 and 1958 on his prewar plan was his work on the theory of functionality (theories based on F),[35] which took the theory more in the direction of type theory than was anticipated in 1941.[36]

After the publication of Curry and Feys (1958a), Curry did publish two papers on other aspects of the prewar plan,[37] but most of his time was taken up with writing Curry (1963a). Thus, by the time he began to work on Curry *et al.* (1972) in earnest, in 1964, most of this prewar plan remained incomplete. As a result, I was able to make this the subject of my dissertation Seldin (1968), most of the results of which were incorporated (in improved form) in Curry *et al.* (1972).

But even after the publication of Curry *et al.* (1972), Curry's original aim of showing that combinatory logic could be used as a foundation for logic had not been completely achieved. The systems proved consistent in Curry *et al.* (1972) §§15B, 16B, and 16C1 are little more than first order logic with some machinery of combinatory logic added. Furthermore, it is impossible to state their postulates with a finite number of axioms and rules; axiom schemes are needed. An example of such a scheme is $P\alpha(P\beta\alpha)$,[38] where α and β belong to a class of terms intended to represent propositions and P is implication. Given Curry's goal of constructing a system based on a finite number of simple postulates, it is desirable to be able to refer to propositions in the system; i.e., to have a term H in the system which can be used to convert schemes like this into axioms. (This H is what was called Pr in Curry (1934b) and Curry (1936a); the idea is that HX is to mean that X is a proposition.) Systems of this sort were proposed in Curry *et al.* (1972) §§15C and 16D, but they were later shown to be inconsistent by Martin Bunder.[39]

Curry (1973) showed that if one of the axioms of the system of §15C of Curry *et al.* (1972) is dropped, the resulting system is consistent in a very weak sense (that every term which occurs in a proof belongs to a class of terms intended to represent propositions). I have proved a stronger consistency result (normalization) in Seldin (1975), (1977a), and (1977b). A formulation of this system is given in Seldin (1976), pp. 82-84. Dana Scott (1975) proposes a model in which this system can be interpreted; in fact, a stronger system can be interpreted in it as well.[40] This model is what is called a Frege structure in Aczel (1980).

However, even in the stronger system which can be interpreted in this model, it is not really possible to quantify over propositions; for we cannot carry out the inference from $Hx \vdash X$ to $\vdash (\forall x \in H)X$. Aczel (1980) shows that the addition of a rule permitting this inference makes the system inconsistent, and concludes from this that it is wrong to quantify over propositions.

However, to give up the possibility of quantifying over propositions would be to give up some of Curry's basic objectives, especially those in the quotation from Curry (1930) I B given above. Besides, in ordinary second order and higher order logic and in various systems of type theory it is possible to quantify over propositions. Hence, it is still important for Curry's Program to seek systems based on combinatory logic in which quantification over propositions is possible.

Recently, Martin Bunder has proposed such a system. It is based on some ideas from type theory, but it is not so restricted. A formulation is given in Seldin (1976), pp. 85-86.[41] What we need to note here is that there is, in effect, a universal quantifier for each type, where types are defined from the basic types A (individuals) and H (propositions) by the operation which forms $F\alpha\beta$ from α and β.[42] A comparison with the systems previously studied suggests that the basic type A be replaced by the universal type E which is such that $\vdash$ EX holds for every term X. (This is equivalent to postulating $\vdash$ AX for all terms X.) However, if this is done then the system is inconsistent.[43] Since the postulates of Bunder's system seem to be the minimum required for quantification over propositions (and propositional functions) in a system based on combinatory logic that does not have all the complexities of type theory, this result suggest strongly that there is a fundamental incompatibility between quantifying over E and quantifying over H.

A possible reason for this incompatibility results from considering the way combinators and λ-terms are to be interpreted. Recall that in the quotation from Curry (1929) above, it is stated that if X is a function then (XY) represents the result of substituting Y for the first variable of X, but if X is not a function then nothing is said about the interpretation of (XY), which may be left uninterpreted. Now consider the following interpretation, under which (XY) is always interpreted provided that the atomic terms are interpreted: the domain is the set of signals that can be fed into a computer. This set will include the signals for a program and for any data that may be used in running it. Then the job control language will provide an operation representing application. If jobs are presented to the computer by means of decks of IBM cards, then the standard deck for a job will provide a place for the cards containing the statements of the program and another place for the cards containing the data for the program. The job control cards will, of course, contain instructions indicating the language in which the program is written and the input-output devices to be used. If these cards are considered part of the program, then the remaining job control cards, with the places they provide for the program deck and the data deck, will provide a framework for the application operation. With the right sort of instructions to the computer, we can put a deck representing such an application in either the program or the data slot of another deck, and so nest applications. Furthermore, if we forget about the interpretation of the output, it makes no difference what signals are put on the cards for the program deck and the data deck, for the computer will read the entire deck and react (perhaps with an error message) no matter what signals are fed in. For this interpretation, (XY) has a meaning for every X and Y in the domain. Furthermore, if X is a properly written program and Y an appropriate set of data, then (XY) will cause the computer to produce the output for the program X applied to the data Y. The important fact about this interpretation for our purposes here is that (XY) will, in general, be distinct from this value. Thus, to say that (XY) exists only means that it makes sense as a computing instruction, not that it has a value.

Now in any system in which we have $\vdash$ QXX for all terms X (i.e. for everything in E) and Rule Eq (which says that if X = Y we have X $\vdash$ Y; this rule is standard in all of these systems), we have $\vdash$ QXY whenever X = Y. But here = refers not to identity (syntactical identity between terms), but rather to giving the same values in any computation. Thus, Q should not refer to logical identity. Another way to put this is that if Z is the value of (XY), then (XY) = Z although (XY) may not be (syntactically) identical to Z, and hence it will follow that $\vdash$ Q(XY)Z. However, there are properties of (XY) which do not hold of Z and vice-versa. For example, if a value is always a term in normal form, then (XY) may not be in normal form but Z clearly will.

This causes no problems in systems in which Q must be taken as a primitive (if it is to be included at all), and this is the case in those systems which can be interpreted in a Frege structure.[44] But as soon as we can quantify over propositions as in Bunder's system, we can define Q by the usual second order definition of logical identity. If this is done, it will follow that $\vdash$ QXX will hold for all terms X for which $\vdash$ AX holds. If A is taken to be E, then $\vdash$ QXX holds for all terms X, and so we have the problem mentioned at the beginning of this paragraph.

What this suggests is that the postulates for A should be chosen so that if $\vdash$ AX holds then X has a value. Then $\vdash$ QXX will hold only for those X which have values, so the only terms identified with their values will be those which have values. I suspect that this is unlikely to cause problems.

An examination of the definition of logical identity shows that the problem is not really quantifying over E, but rather quantifying over FEH. Thus, it may still be possible to add E as a "large type", with the understanding that $F\alpha\beta$ is a (small) type only when α and β are small types.

Since the set of terms which represent values can be defined in several ways (e.g. terms which have a normal form, terms which have a head normal form, etc.), there will be several ways to define the postulates for A. One way is simply to take AX as an axiom for each X with a value. But then the set of axioms will not be decidable. An alternative is to use postulates such as those of Feferman (1975a) and (1975b) for a term to be defined or those of Chauvin (1979) for a term to be an object.[45] Since these postulates result from semantic considerations, it is not unreasonable to hope that models can be found for Bunder's system with these (or similar) postulates for A.

FOOTNOTES

1. The talk was given in December 1976 in Oxford. In April 1977 I delivered essentially the same talk at Rome, Pisa, and Turin, and a summary of it was published as Seldin (1976). In April 1978 I gave an updated version of the same talk at Zagreb.
2. The major exception is the work Curry did for the U. S. government during World War II, when he put logic aside. See the biographical sketch elsewhere in this volume.
3. See the discussion in Curry and Feys (1958a) §0A.
4. References to works by Curry or by Curry and others are given in the bibliography of his works elsewhere in this volume. References to all other works cited in this paper are given at the end of the paper. It is important to stress that Curry's ideas have changed since 1929.
5. (Curry's footnote) "Hilbert, D., and Ackerman, W., *Grundzüge der theoretischen Logik*, 1928, p. 1."
6. (Curry's footnote) "Johnson, W. E., *Logic*, Part I, Cambridge (1921), p. xiii."
7. (Curry's footnote) "See Johnson, W. E., *l. c.*, p. 1. Johnson's definition of the proposition is what I have here given as the definition of a statement of fact."
8. (Curry's footnote) "The term idea is used here to denote an object, not a process of thought."
9. (Curry's footnote) "*I. e.*, their meaning may be partly conventional."
10. (Curry's footnote) "Strictly speaking we should consider in the theory not only statements that an entity is an assertion, but also statements that such and such combinations are entities. But the latter are, in simple cases at least, of so trivial a nature that it is not necessary to give them special prominence."
11. (Curry's footnote) "Otherwise the rule would make possible the addition of only a finite number of constituents, and these could just as well be added explicitly to the preceding categories of the primitive frame."
12. (Curry's footnote) "In writing the foregoing account I have naturally made use of any ideas I may have gleaned from reading the literature. The writings of Hilbert are fundamental in this connection. I hope that I have added clearness to certain points where the existing treatments are obscure."
13. This is one point on which Curry's ideas have changed since 1929. As a result of Gödel's theorems, it is no longer possible to believe that one formal system can ever be "adequate for the whole of logic". Today Curry would speak instead of being able to find a formal system for any particular part of logic in which one might be interested.

14. (Curry's footnote) "See the paper of Schönfinkel cited below."
15. (Curry's footnote) "'Ueber die Bausteine der mathematischen Logik,' *Mathematische Annalen*, Vol. 92 (1924), pp. 305-316."
16. (Curry's footnote) "In this presentation I have changed Schönfinkel's formulation in some matters of detail."
17. This is taken in the form $\vdash \Pi(W(CQ))$. I am counting this as illative rather than pure because of the universal quantifier. The system is like those of Curry and Feys (1958a) Chapter 7 in that equations between terms (of the form X = Y) are expressed using Q (as in $\vdash$ QXY). Note that in this early work 'X = Y' exists only as an abbreviation, whereas in later work = is a relation with much the same logical status as $\vdash$.
18. The fact that Curry postulated the elimination rule for the universal quantifier (Π) and implication (P) and the introduction rule for conjunction (Λ) is probably due to the fact that none of Gentzen's work had yet appeared.
19. Curry wrote the first of these rules as follows: "if $\vdash$ X and $\vdash$ QXY then $\vdash$ Y". I am using here some later conventions of Curry's in order to coincide with most other people's use of the symbol '$\vdash$'. For Curry every statement of his formal system (and hence every step in any formal proof) must have the form $\vdash$ X, where X is a term (entity, ob) of the system. In the later systems, in which = is a relation, statements can equally have the form X = Y.
20. The word 'prelogic' is Curry's own as a translation for 'Urlogik'.
21. I wish to thank Gerd Jaeger and Karl Gosejacob for helping me by going over with me this translation (from the original German); however, any shortcomings in this translation are my responsibility. Curry once told me that he did not feel that he could translate any of his own works written in other languages without rewriting them. On Curry's own attitude toward foreign languages, see Curry (1963a), pp. vii and 89 (the first of these is on p. v in the 1977 reprint.)
22. (Curry's footnote) "E.g. we must do without implication and the definite article as primitive concepts in the *Principia Mathematica*. The definitions of these primitive concepts given there lead to paradoxical results. In fact, there the King of France is different from the King of France; and we can make no distinction between the two propositions 1) 'If all men had three hands, Bismarck would have three hands,' and 2) 'If all men had three hands, the moon would be made out of green cheese.'"
23. Note that this definition of combinatory logic seems to be compatible with what John Kearns (1980) defines to be combinatory logic, which is something distinct from a theory of combinators. See my remarks later in this paper in §2

just after the quotation from Curry (1934b).

24. This is a reference to the rule of modus ponens.
25. See footnote 13.
26. (Curry's footnote) "It is well worth remarking that a concept by this definition is an object and not a process of thought. Some examples of concepts are Bismarck, Göttingen, animal, umbrella, red, temperature, matter, substance, causality, entity (Etwas), function, the King of France, the greatest cardinal number, etc.

26a. This is a word Curry himself coined to translate "inhaltlich". It means having a meaning understood prior to the definition of the system or theory in question. See Curry and Feys (1958a) p. 21.

27. Because several people suggested "extraneous meanings" for the word "entity", Curry later became dissatisfied with it. For some time he used the word "term", but when discussing logic with quantifiers, he fould himself wanting to use the word "term" where most people use the word "formula"; and so he coined his own word "ob" (from "object") to translate "Etwas". Despite what some people seem to believe, the word has no special connection with combinatory logic, and means nothing more than a formal object of a certain kind of formal system.
28. Those interested in the history of the formulae-as-types notion may be interested in the fact that in this paper, Curry (1934b), the name 'Axiom (PK)' is given to the formula which is there written as '$\vdash (x,y)(x \supset (y \supset x))$', and similarly for (PB), (PC), and (PW). In Curry (1936a), although these names are not given, theorems are proved for each of them such as the following: if $\vdash (x,y)(x \supset (y \supset x))$, then $\vdash (x,y)F'y(F'xy)K$, where F' is a defined operator for functionality defined by $F' \equiv [x,yz](u)(xu \supset y(zu))$.
29. It is worth emphasizing that the axioms and rules of Curry (1930) were not enough to produce a contradiction, and in fact this system is there proved consistent. Kleene and Rosser themselves reported to Curry that they had tried in vain to derive a contradiction in the system of Church (1932) before Curry (1934a) appeared, but this latter paper gave them what they needed. See Curry and Feys (1958a) p. 274 footnote 18.
30. See Curry and Feys (1958a) p. 274.
31. See Curry (1941e).
32. Curry (1942c) derived a contradiction from combinatory completeness and properties of implication. For a more recent exposition of this contradiction, usually called *Curry's Paradox*, see Curry and Feys (1958a) pp. 258-9.
33. See footnote 2.
34. See footnote 2.
35. See the abstracts Curry (1955) and (1956).

36. This can be seen by comparing Chapters 8 and 9 of Hindley, Lercher, and Seldin (1972).
37. Curry (1960a) and (1961b).
38. If '$\alpha \supset \beta$' is used as an abbreviation for '$P\alpha\beta$', then this scheme would be written '$\alpha \supset (\beta \supset \alpha)$'. This, of course, is for a system based on Π (universal quantifier) and P (implication). For a system based on Ξ (restricted generality), similar schemes are needed. See Curry *et al.* (1972) §§15B, 16B, and 16C.
39. See Bunder (1976) and Bunder and Meyer (1978). Actually, the system of Curry *et al.* (1972) §15C is not known to be inconsistent if E is replaced by a more restricted fundamental domain of individuals A (so that L is FAH and $\vdash$ AX does not hold for all terms X). Bunder (1974) showed that a similar system proposed in Seldin (1968) (using E) is inconsistent. See also Bunder (1978).
40. See Seldin (1976) pp. 84-85.
41. The system is proposed in Bunder (1979a), which I saw as rough notes in 1976. The system presented in Seldin (1976) is a simplified version used in Bunder (1979b) and (1979c) as a basis for set theory. The presentation in Seldin (1976) requires one important correction: Bunder did not use E in his system, but used a more restricted domain A of individuals.
42. F can be defined in this system as in Curry and Feys (1958a) p. 266.
43. See Seldin (1976) p. 86.
44. See Scott (1975) and Aczel (1980).
45. They are essentially the same postulates. If one defines A to be WQ and uses these postulates to determine the terms X for which $\vdash$QXX holds, one has an approach similar to that of Goodman (1970) and (1972). The former paper brings out the semantic considerations especially clearly.

ACKNOWLEDGEMENTS

I would like to thank the Johns Hopkins University Press for permission to reproduce the quotations from "An analysis of logical substitution" by H. B. Curry, *American Journal of Mathematics*, Vol. 51 (1929), pp. 363-384, copyright 1929 by the Johns Hopkins University Press and from "Grundlagen der kombinatorischen Logik" by H. B. Curry, *Ibid.*, Vol. 52 (1930), pp. 509-536 and 789-834, copyright 1930 by the Johns Hopkins University Press. I would also like to thank the editors of the Tôhoku Mathematical Journal for permission to reproduce the quotations from "First properties of functionality in combinatory logic" by H. B. Curry, *Tôhoku Mathematical Journal*, Vol. 41 (1936), Part II, pp. 371-401, copyright 1936 by the Tôhoku Mathematical Journal.

REFERENCES

Aczel, P. (1980). Frege structures and the notions of proposition, truth and set. *In* "The Kleene Symposium: A Collection of Papers on Recursion Theory and Intuitionism" (Eds. John Barwise, H. Jerome Keisler, and Kenneth Kunen), forthcoming. North-Holland, Amsterdam.

Bunder, M. W. (1974). Some inconsistencies in illative combinatory logic, *Zeitschr. math. Logik* 20, 199-201.

Bunder, M. W. (1976). The inconsistency of $\mathfrak{F}^*_{21}$, *J. Symbolic Logic* 41, 467-468.

Bunder, M. W. (1978). Equality in $\mathfrak{F}^*_{21}$ with restricted subjects, *Zeitschr. math. Logik* 24, 125-127.

Bunder, M. W. (1979a): Predicate calculus of arbitrarily high finite order, *University of Wollongong, Department of Mathematics Preprint Series* No. 2/79.

Bunder, M. W. (1979b). A one axiom set theory based on higher order predicate calculus, *University of Wollongong, Department of Mathematics Preprint Series* No. 3/79.

Bunder, M. W. (1979c). Set theory in predicate calculus with equality, *University of Wollongong, Department of Mathematics Preprint Series* No. 8/79.

Bunder, M. W. and Meyer, R. K. (1978). On the inconsistency of systems similar to $\mathfrak{F}^*_{21}$, *J. Symbolic Logic* 34, 1-2.

Chauvin, A. (1979). Theory of objects and set theory: introduction and semantics, *Notre Dame J. Formal Logic* 20, 37-54.

Church, A. (1932). A set of postulates for the foundation of logic, *Ann. of Math.* (2) 33, 346-366. Second paper, *Ibid.* 34 (1933), 839-864.

Church, A. and Rosser, J. B. (1936). Some properties of conversion. *Trans. Amer. Math. Soc.* 39, 472-482.

Feferman, S. (1975a). A language and axioms for explicit mathematics. *In* "Proceedings of the 1974 Summer Research Institute, Monash" (Ed. J. N. Crossley), pp. 87-139. *Springer Lecture Notes in Math.*, Vol. 450.

Feferman, S. (1975b). Non-extensional type-free theories of partial operations and classifications, I. *In* "Proof Theory Symposium, Kiel, 1974" (Eds. J. Diller and G. H. Müller), pp. 71-118. *Springer Lecture Notes in Math.*, Vol. 500.

Goodman, N. (1970). A theory of constructions equivalent to arithmetic. *In* "Intuitionism and Proof Theory. Proceedings of the Summer Conference at Buffalo N. Y. 1968" (Eds. A. Kino, J. Myhill, and R. E. Vesley), pp. 101-120. North-Holland, Amsterdam.

Goodman, N. (1972). A simplification of combinatory logic, *J. Symbolic Logic* 37, 225-246.

Hindley, J. R., Lercher, B., and Seldin, J. P. (1972). "Introduction to Combinatory Logic. London Math. Soc. Lecture Note No. 7", University Press, Cambridge.

Kearns, J. T. (1980). Fully explicit deductive systems. *In*, this volume.

Kleene, S. C. and Rosser, J. B. (1935). The inconsistency of certain formal logics, *Ann. of Math.* (2) 36, 630-636.
Schönfinkel, M. (1924). Ueber die Bausteine der mathematischen Logik, *Math. Ann.* 92, 305-316.
Scott, D. S. (1975). Combinators and classes. *In* "λ-Calculus and Computer Science Theory, Rome, 1975" (Ed. C. Böhm), pp. 1-26. *Springer Lecture Notes in Computer Science*, Vol. 37.
Seldin, J. P. (1968). "Studies in Illative Combinatory Logic", Dissertation, University of Amsterdam.
Seldin, J. P. (1975). Equality in $\mathfrak{F}_{22}$. *In* "Logic Colloquium '73. Proceedings of the Logic Colloquium, Bristol, July, 1973" (Eds. H. E. Rose and J. C. Shepherdson), pp. 433-444. North-Holland, Amsterdam.
Seldin, J. P. (1976). Recent advances in Curry's Program, *Rend. Sem. Mat. Univers. Politecn. Torino* 35, 77-88.
Seldin, J. P. (1977a). The Q-consistency of $\mathfrak{F}_{22}$, *Notre Dame J. Formal Logic* 18, 117-127.
Seldin, J. P. (1977b). The predicate calculus in $\mathfrak{F}_{22}$ (Abstract), *J. Symbolic Logic* 42, 153.
Stenlund, S. (1972). "Combinators, λ-Terms, and Proof Theory", Reidel, Dordrecht.
Whitehead, A. N. and Russell, B. (1910). "Principia Mathematica", 3 vols., University Press, Cambridge, 1910-1913. Second edition 1925-1927.

FULLY EXPLICIT DEDUCTIVE SYSTEMS

John T. Kearns

Department of Philosophy, Baldy Hall,
State University of New York at Buffalo,
Buffalo, New York 14260, U. S. A.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

In this paper I will try to understand and describe what Seldin calls Curry's Program (in (1976-77) and (1979)). Curry has long been associated with combinatory logic. He did not invent the subject, but he certainly deserves most of the credit for having developed it. While it is easy to say that Curry has been concerned to develop combinatory logic, I find it very difficult to say just what combinatory logic is. Indeed, if we look at Seldin's account of Curry's work (in the paper in this Festschrift), combinatory logic seems to have been different things at different times. Perhaps it would be better to say that combinatory logic includes many very different topics and areas. But it is not easy to describe combinatory logic in a way that makes clear which topics belong to it, and how they are related to each other.

Not only do I find it hard to describe combinatory logic, I also find it hard to describe my problem with combinatory logic. After all, I can look at the books on combinatory logic and the articles devoted to the subject. I am sometimes required to write abstracts or reviews of these articles, and at least some of those times I succeed in saying what has been accomplished in an article on combinatory logic. What I find difficult to do is describe combinatory logic, and Curry's

Program, in a way that makes clear the point of it all. I cannot explain combinatory logic in terms that provide a basis for judging that a certain result is of great importance while some other is a minor achievment. I cannot determine the extent to which Curry and others have succeeded in carrying out his program, because I am unable to be specific about what would constitute success. For example, one of the central goals of Curry's Program has been to provide a foundation for logic. But what does this mean? One way that a foundation might be provided for subject *B* would be by developing the theory of a different subject *A*, which theory properly contains a theory of subject *B*. This does not seem to apply to logic, for it is hard to see how there can be a different and more basic subject to which logic might be reduced.

My difficulty in describing combinatory logic is not due to a defect of combinatory logic. I believe that a large part of the difficulty can be traced to there not being a conceptual framework available for thinking about and describing Curry's Program. We don't have a system of expressions (and the related concepts) that we can use to characterize this program. For this reason, we are unable to be specific about the problems that are addressed. And this limits our ability to evaluate various accomplishments. If we don't know what is the destination, we can't tell if we are getting closer.

In this paper I will begin to develop a conceptual (and linguistic) framework that is adequate for characterizing Curry's Program. This does not mean that I will set up a new language filled with new expressions. Some new expressions will be helpful. But the basic task is to assemble existing expressions (and concepts) to form a suitable system. In what follows I will construct a description of Curry's Program. I will generally use expressions that are already available. Many of these have not previously been used to describe combinatory logic. My description provides a perspective from which to view Curry's Program. The expressions I employ constitute part of a framework which can accommodate combinatory logic.

Curry's Program as I describe it will not be Curry's Program as carried out by Curry. It will be Curry's Program as understood and adapted by Kearns. (Perhaps we should give this a name like [CPK].) I have tried to determine what is the "essence" of what Curry is up to; my perception of what is essential determines the program I shall describe. This program is both more general and more restricted than Curry's actual program. It is more general because it allows for systems unlike those Curry has developed. It is more restricted because it deals with expressions rather than obs. Some features which I take to be essential may be regarded as incidental by Curry (and vice versa). The program I describe is coherent. It makes sense of much of what Curry and others have done and are doing. This program is reasonable and interesting.

2. COMBINATORY LOGIC

I consider logic to be a study of language and languages. Curry obviously takes logic to be more general than this, for his obs need not be symbols or expressions. I will present my description of what Curry is up to in terms of language and symbols. Anyone who is dissatisfied with this linguistic limitation should have little trouble reformulating my description in more general terms.

When Curry began working on combinatory logic, he investigated the manipulation of symbols. The study of combinators and their reduction relations seems best understood in this way. Schonfinkel used combinators to show that bound variables are unnecessary in systems of logic. (Since free variables can usually be understood as if bound by an initial universal quantifier, they aren't needed either.) Bound variables mark the places in a formula that are affected by a quantifier. A device to achieve the same effect as variables (but without the variables) must be notational. It is true that when combinators are combined with denoting expressions, we can use them to represent functions of the denoted objects. But combinators can be applied to any kinds of expressions; their reduction rules are not affected when they are applied to nonsignificant expressions. Combinators are symbols which "express" rules for manipulating symbols. Curry's initial concern with symbols and their manipulation also shows up in his attempts to analyze the rule of substitution, and to reduce it to rules whose complexity is no greater than that of modus ponens.

Later Curry became more interested in investigating deductive systems than in developing theories of combinators (and the λ-calculus). The systems he studied usually contain combinators, but in illative combinatory logic these deductive systems are much more general than simple theories of combinators. The development and investigation of these deductive systems now seems to be the central ingredient of Curry's Program.

The deductive systems that Curry studies have certain unusual features, but these systems are closely related to standard systems of logic and mathematics. Curry's systems are conceived as replacements for familiar systems rather than as systems for developing the theory of some new subject matter. The unusual features of Curry's systems are due to what I will call their *explicit* character. The central thrust of Curry's Program (of [CPK]) is to develop (and investigate) *fully explicit deductive systems*. These systems will give new formulations to standard systems of logic and (some systems of) mathematics. Their explicit character makes them suitable for the analysis of features and presuppositions of the standard systems.

One feature of fully explicit deductive systems is that much of what would ordinarily be considered to belong to syntax is made an explicit part of the system. Conventional deductive systems usually make syntax preliminary in the sense that formation rules generating well-formed formulas and sentences are given before the axioms and rules of the system are presented. The expressions generated by the formation rules are the significant sentences (or formulas); only these can occur as steps in proofs. In a fully explicit deductive system, much less is preliminary. There is a list of basic (simple) expressions. (If it is finite; otherwise they must be presented some other way. But a finite list is most fully in keeping with a fully explicit deductive system.) And there is a procedure for combining expressions to obtain complex expressions. Ordinarily in Curry's systems, complex expressions are formed by (binary) application. However, this should not be considered essential. The expressions of a fully explicit deductive system could as well be unstructured strings formed by concatenation. (Expressions formed by application are *structured* strings.)

In a conventional formal system, the formation rules ordinarily restrict the expressions that can be combined in a complex expression. For example, we might have a rule (a clause) that says: If A, B are sentences, then $[A \vee B]$ is a sentence. This rule does not authorize us to form '[v v v].' In a fully explicit deductive system, the procedure for obtaining complex expressions is unrestricted. Any expressions can be combined. So '[v v v]' would be allowed. A fully explicit deductive system employs some expressions which aren't significant. In such a system we begin with arbitrary expressions, but the system may enable us to prove that a given expression is a sentence, or belongs to some other category (type) of significant expressions. This is the sense in which syntax can be an explicit part of the theory.

It might be in keeping with the explicit character of these systems to require that the construction of complex expressions be within the theory, not preliminary to it. To develop the system, we would start with a list of basic expressions. Then we would describe a construction process for obtaining complex expressions (a construction process is given by giving rules that take one or more expressions to an expression). The constructions which yield complex expressions need not be regarded as proofs--another feature of fully explicit deductive systems is that they contain other constructions than proofs. For example, if we consider complex expressions generated by binary application, we might illustrate the construction of $((\alpha\beta)(\gamma(\alpha\gamma)))$ this way:

$$\dfrac{\dfrac{\alpha \quad \beta}{(\alpha\beta)} \qquad \dfrac{\gamma \qquad \dfrac{\alpha \quad \gamma}{(\alpha\gamma)}}{(\gamma(\alpha\gamma))}}{((\alpha\beta)(\gamma(\alpha\gamma)))}$$

We could require that no expression appear in the deductive system until it has been authorized by such a construction. If these constructions appeared in the system, then a restricted procedure for obtaining complex expressions could be allowed without depriving the system of its explicit character. When the structure of complex expressions is especially simple (as is the case with concatenation and binary application), there is no advantage to including the constructions in the system. So let us understand this to be an optional feature of fully explicit deductive systems; they may or may not incorporate constructions of complex expressions. (A deductive system is sometimes considered to consist of results given by constructions. I am taking the systems to include the constructions as well.)

The next important feature of fully explicit deductive systems has already been mentioned--they contain constructions other than proofs. A *construction process* is a procedure given by rules for obtaining an expression (the result) from premisses (some word that does not suggest inference would be better).

A particular instance of a construction process is a *construction*. The premisses in a rule associated with a construction process may be expressions or constructions. I will represent constructions by tree structures that work down to the result. A *proof* is a construction that contains only sentences or sentential formulas as (uncancelled) elements, and that is intended to provide support for the result. (The parenthetical qualification is inserted so that a construction like:

$$\dfrac{\dfrac{\dfrac{SKXYZ}{KY(XY)Z}}{YZ}}{SKXYZ \twoheadrightarrow YZ}$$

can be counted as a proof. The result at the bottom "sums up" the reduction sequence which is its premiss.) A *valid* proof is one that is truth-preserving. The constructions (and construction processes) in a fully explicit deductive system can be subdivided into two classes. These are distinguished on the basis of our intentions for them. One class of constructions contains proofs as special cases. They are generated by construction processes which yield constructions that are intended to be valid proofs when every uncancelled element is a sentence (or sentential formula). These proofs may not be valid--if the system is defective--but they are intended to be. Such constructions are *generalized proofs*. (So a proof is a generalized proof.) Constructions in the second class are not proof-like. These constructions are *computations*. An example is provided by a reduction sequence or a numerical calculation.

Curry initially constructed deductive systems that did not incorporate their syntax--they did not possess the resources for establishing that a given expression belongs to a given category (type). When these systems turned out to be inconsistent, Curry decided that inconsistency and paradox result from not distinguishing between significant sentences and other expressions. By incorporating syntactical statements and syntactical results in a deductive system, we can have rules that require premisses stating that various expressions belong to appropriate categories. This "whittles down" the class of generalized proofs so that it comes closer to the class of valid proofs. Curry apparently does not favor going all the way in this direction (neither does Seldin). But it isn't clear to me that there is a (good) reason for having a class of generalized proofs in a system, which class is broader than the class of valid proofs. One might argue that this practice is in keeping with the explicit character of fully explicit deductive systems. Just as we allow more expressions than significant expressions, so we should have more constructions than valid proofs. Since proofs in formal deductive systems can be characterized in syntactic terms, we should deal with a more general class of constructions in order to determine what are the special features of valid proofs. This argument by analogy is not very convincing. When dealing with syntactic features of expressions, we establish (in the system) results in the form of sentences. We don't have similar results stating features of proofs. The practice of allowing generalized proofs also creates semantic problems, which I will discuss in section 4.

We might also ask why it is desirable to allow constructions (computations) which aren't proof-like. We know that it is possible to recast such constructions in the form of proofs. E.g., instead of:

$$\frac{\dfrac{SK(BW)I}{KI(BWI)}}{I}$$

we might have:

$$\frac{SK(BW)I \twoheadrightarrow KI(BWI) \qquad KI(BWI) \twoheadrightarrow I}{SK(BW)I \twoheadrightarrow I}$$

But there are several advantages to having computations rather than proofs. One advantage concerns convenience. The proofs to which computations are reduced are much more cumbersome than the computations. And retaining computations makes it possible to eliminate at least some sentences attributing syntactic features to expressions. In modern logic, it is taken to be desirable that a formal system have an extensional character. This means that expressions which are equivalent in some sense (usually in the sense of denoting the same objects) are interchangeable without affecting the values of expressions in which they occur. I don't understand why extensionality is desirable, unless it is that extensional systems are easier to study than nonextensional systems. But to have an extensional system, we must banish predicates which represent properties possessed by only some members of an equivalence class of expressions. A third advantage of retaining computations is that this can help to achieve systems having a finite number of axioms. When computations are reduced to proofs, it will often be difficult to avoid using axiom schemes.

I shall say that a *fully explicit deductive system* is a deductive system (1) whose expressions are given by listing or (effectively) describing the basic (simple) expressions, and (optionally) describing unrestricted procedures for obtaining complex expressions. This system is generated by construction processes which take expressions as inputs and yield expressions as results. A "concrete" instance of a construction process is a *construction;* the constructions of a fully explicit deductive system include (2) (optionally) constructions generating complex expressions, (3) generalized proofs, and (4) computations. (The two options form an exclusive disjunction.)

Curry's Program, as I understand it, is to develop fully explicit deductive systems to replace standard systems of logic and mathematics, and to further investigate fully explicit deductive systems. I propose that the expression 'combinatory logic' be used as a name for the study of fully explicit deductive systems. Studies of combinators and their reduction relations will belong to the *theory of combinators* (or else they will constitute various theories of combinators). In "The Logic of Calculation" I have argued that theories of combinators belong to the study of algorithms (but in that paper I used 'combinatory logic' as a name for theories of combinators, which practice I now propose to change.) A theory of combinators can be developed by means of a deductive system, but it is not essential that a fully explicit deductive system contain such a theory.

When Curry's Program is construed as I have done, then I can see the point of it. I can determine that a certain result is important or unimportant. And I have a basis for making criticisms and suggestions. Given my understanding, combinatory logic is a foundational study. Combinatory logic does not reduce logic to something else, but it aims at making logical systems as explicit as possible, reducing what is preliminary or presupposed to a minimum.

In the remainder of this paper, I will make some criticisms and suggestions.

3. PARADOXES

It is still not clear how we are to understand the various paradoxes that have arisen in logic and mathematics. One explanation has it that the paradoxes are due to treating an expression which is not a significant sentence as if it were. (Other terminology is sometimes used; Curry speaks of treating nonpropositions like propositions.) It is undoubtedly true that a contradiction can arise from treating a nonsentence like a sentence. But this does not seem to me to account for all paradoxical sentences. Many of them (e.g., 'This very sentence is false') strike me as perfectly significant. I understand them all right, and this is why I find them perplexing. In this section I will make a proposal for understanding "significant" paradoxes; this proposal gives us a framework for dealing with paradoxes, and enables us to catalogue the various procedures for "getting around" the paradoxes. The proposal has consequences for the practice of allowing generalized proofs in fully explicit deductive systems.

Let us consider a system of propositional logic, and deal only with the significant sentences of this system. Such a system will be a proper part of a fully explicit deductive system, but we can focus on it. The system of propositional logic and the language that goes with it will be developed in stages. In the system some sentences are true; their truth consists in a correspondence of some sort with reality. But how are we to understand falsity? We might say that a (significant) sentence is false if it is not true. But this explains falsity by means of negation, and I think that negation should be explained by means of falsity. To account for falsity, I propose that we recognize incompatibility as a fundamental and primitive relation. A sentence is false if, and only if, it is incompatible with true sentences. On this account, truth is an "absolute" property. True sentences owe their truth to the way things are. But falsity is relative to a language. Sentence *A* in language *L* is false with respect to *L* if it is incompatible with true sentences of *L*. A sentence *A* of *L* which is neither true nor false may be false with respect to some extension *L+* of *L*.

To begin with, consider a system of propositional logic that contains just the connectives 'v,' '&,' and '⊃.' (There is no negation.) We want to construct proofs in this system. To understand a proof, we must understand its point. For most proofs, the purpose of the proof is to show either that some sentence *B* is true, or that sentence *B* follows from sentences $A_1, \ldots, A_n$ (or to show analogous results for formulas). Truth and falsity are properties that are independent of epistemic considerations. But proofs and their purposes are not; a proof aims a producing knowledge.

To deal with proofs and their "significance," I will make use of some terminology due to J. L. Austin and his school. A (true or false) sentence can be *asserted* by a speaker or writer. A language user who asserts a sentence expresses her acceptance of the sentence (her acceptance of the truth of the sentence). Asserting, like some other linguistic acts such as promising, threatening, and requesting information, is an *illocutionary act*. When a sentence is used to perform an illocutionary act, that act has an *illocutionary force* which determines it to be a specific kind of illocutionary act. A language contains expressions (e.g., 'I assert that,' 'I promise that,' 'I request that') for making the illocutionary force of an act explicit. Ordinarily these expressions are not used, for the circumstances in which the act is performed make its illocutionary force clear. But in constructing proofs in our system of propositional logic, I will introduce expressions to mark illocutionary force. We only need to consider acts which involve accepting true sentences and rejecting false ones.

A sentence which is asserted will be prefixed with '$\vdash$.' As well as asserting that a sentence is true, we can *suppose* that it is true. If we are supposing that A is true, we prefix it with '—.' It is not only hypotheses that are supposed; so are the consequences we infer from them. Assertion has an opposite, denial; I will use '$\dashv$' to mark a denial. We can also distinguish positive and negative suppositions. If we are supposing that A is false, we write: $\neg A$. In ordinary proofs, sentences (and formulas) without explicit force indicators are conventionally understood to have a positive force (— or $\vdash$). We can adopt this practice later; for the present we require that every step in a proof be accompanied with a force indicator.

It is essential to understand that force-indicating symbols are not predicates. We do not use them to talk about sentences; they show what we are doing with the sentences. Because of their role, it makes no sense to iterate force-indicating devices. So we will not deal with $\dashv \dashv A$ or $— — A$, for example. Similarly force-indicating devices cannot be incorporated in a sentence. There is no sentence $[A \vee \neg A]$.

Any sentence can be asserted by someone, whether the sentence is true or false. Similarly, both true and false sentences can be denied. The assertion of a sentence is *justified* if the sentence is known to be true, or if it is deducible from sentences known to be true. The denial of a sentence is *justified* if the sentence can (by deduction) be shown to be incompatible with sentences whose assertion is justified. (So an assertion can be justified without this being known.) The preceding defines *plain* justification. An assertion is *plainly* justified if it is known by any means to be true, or is deducible from such. There are any number of stronger justification concepts. Some stronger concepts require that a sentence be provable in a certain framework for its assertion to be justified. A very strong requirement insists on constructive proofs (for mathematical sentences). In this paper I shall consider only plain justification. When we construct proofs we are concerned to preserve or to establish justification. So a proof might justify $\vdash A$ or $\dashv A$. A proof might show that the assertion or denial of A is justified relative to certain assumptions.

Our system of propositional logic employs tree proofs. These proofs establish results which are *inference sequences*. An inference sequence consists of zero or more premisses and a conclusion, and will be written this way: $F_1A_1, \ldots, F_nA_n$ / $F_{n+1}B$. Here $A_1, \ldots, A_n$ are the premisses, and B is the conclusion. The F_i's are force-indicating expressions. If $n \geq 1$, then each of $F_1, \ldots, F_{n+1}$ is '$—$' or '$\neg$.' If $n = 0$, then F_1 is '$\vdash$' or '$\dashv$.'

I will not provide a complete list of rules for constructing tree proofs in the deductive system. At present I am discussing the system of propositional logic only because of the light that it sheds on the paradoxes. In the system, each step in a proof must be accompanied with a force indicator; rules would be illustrated with suppositional force indicators in front of premisses and conclusion. For example, one of the rules v Introduction is illustrated:

$$\frac{— A}{—[A \vee B]}$$

It is understood that such a rule remains applicable if a premiss which is a supposition is replaced by the corresponding assertion or denial (i.e., if — A is replaced by $\vdash A$, or $\neg A$ by $\dashv A$). If all the premisses of an instance of a rule are assertions or denials, then the conclusion is replaced by the corresponding assertion or denial. So an instance of v Introduction can look like this:

$$\frac{\vdash A}{\vdash[A \vee B]}$$

The rule $\supset$ Introduction has a subproof for a premiss, and cancels a hypothesis of the subproof. Such a rule is illustrated like this:

$$\begin{array}{c}[— A]\\ \underline{\quad — B \quad}\\ —[A \supset B]\end{array}$$

The cancelled hypothesis, together with its force indicator, is enclosed in brackets. If the conclusion of a tree proof depends on no uncancelled hypotheses, then it can be prefixed with '$\vdash$' or '$\dashv$.'

Rules which involve negative force indicators in an essential way are the following:

Contradiction Elimination $\qquad\qquad$ $\neg$ Introduction

$$\frac{— A \qquad \neg A}{— B} \qquad \frac{— A \qquad \neg A}{\neg B} \qquad\qquad \begin{array}{c}[— A]\\ \underline{\quad \neg A \quad}\\ \neg A\end{array}$$

We cannot adopt this rule:

$\neg$ Elimination

$$\frac{\begin{array}{c}[\neg\ A] \\ \vdash A\end{array}}{\vdash A}$$

unless we know that every sentence is true or false.

The system as we have it so far has no expression for saying (or indicating) that a sentence is false. The force-indicating expressions '$\neg$' and '$\dashv$' are neither predicates nor connectives; they are not used to "say" anything. It is desirable to have an expression in the language to "represent" falsity. Let us introduce the connective '$\sim$' to serve this purpose. It is our intention that a sentence $\sim A$ will be true if, and only if, A is false. We want to adopt appropriate rules of inference for '$\sim$,' but, before we do this, we will consider a problem that may arise. Our propositional system is conceived as part of a larger system whose details we are ignoring. But if the larger system contains suitable resources, we may find after introducing '$\sim$' that there is a sentence A such that both $\vdash A$ / $\vdash \sim A$ and $\vdash \sim A$ / $\vdash A$ can be established as results in our system. If this happens, we will say that both A, $\sim A$ are *paradoxical sentences (with respect to '$\sim$')*. By my account of falsity, both A and $\sim A$ will be false. This is *not* a disastrous result. It shows that in a sufficiently rich system we cannot realize our intention for '$\sim$.' We cannot have an expression which "characterizes" exactly the false sentences. (We can introduce a negating expression which characterizes those sentences which are "already" false. The new expression generates new sentences; it is not adequate for characterizing all of the new false sentences.)

For reasoning with '$\sim$,' we can adopt these rules:

$\neg$ Introduction #1

$$\frac{\vdash \sim A}{\neg\ A}$$

$\neg$ Introduction #2

$$\frac{\vdash A}{\neg \sim A}$$

Either can be used to derive the other (in the presence of a reasonable set of rules for the other connectives). Given these rules, if A and $\sim A$ are paradoxical sentences, we can establish both $\dashv A$ and $\dashv \sim A$. We cannot (in general) adopt the following as rules:

$\sim$ Introduction

$$\frac{\neg A}{- \sim A}$$

$\sim$ Elimination

$$\frac{- \sim A}{\neg A}$$

$\sim$ Introduction is acceptable for nonparadoxical sentences, even when (i) a concept of justification stronger than plain justification is under study, or (ii) some sentences are neither true nor false. $\sim$ Elimination is acceptable for nonparadoxical sentences when plain justification is considered, for a language in which every sentence is true or false.

Without the forbidden principles $\sim$ Introduction, $\sim$ Elimination, our system is not adequate for making inferences which involve '$\sim$.' I will say that a sentence A is *manageably false* if A is false but not paradoxical. The practice of logicians (and mathematicians) in modern times has been to restrict their attention to a class of false sentences that is a subset of the manageably false sentences. (In the course of doing this, they also end up dealing with less than all the true sentences.) This restriction can be accomplished either by constructing a language which contains no paradoxical sentences, or by allowing paradoxical sentences in the language but finding some device for keeping them "at bay."

In this paper I will not develop a procedure for dealing with paradoxical sentences. I am only concerned to understand and explain what is going on with significant paradoxical sentences. Those paradoxical sentences which are significant are false. It isn't possible to have a negating expression (whether this is a connective like '$\sim$,' a predicate like 'is false,' or some other kind of expression) which is used to "characterize" just the false sentences--in a sufficiently rich language. We can always deny, or reject, a false sentence. We cannot always use a specified expression to say that it is false. If we consider all the sentences in a sufficiently rich language/deductive system, we can't conveniently codify the correct inferences involving '$\sim$' (or whatever negating expression we employ). To obtain a convenient codification, it is necessary to restrict the sentences to which (some of) the deductive principles apply. Once we are dealing with a class of sentences A such that $\sim A$ is true iff A is false, we can drop the negative illocutionary force expressions. Instead of $\urcorner A$, we can use $— \sim A$; and for $\dashv A$ we can use $\vdash \sim A$. But then we can drop illocutionary force expressions altogether, for the situation in which a sentence occurs will make it clear enough whether the force is — or $\vdash$.

Since combinatory logic (as I am understanding it) is a foundational study, it is appropriate that logical paradoxes be investigated in combinatory logic. (Curry has often made similar remarks.) But the practice of dealing with generalized proofs whose steps are not significant sentences makes it difficult to carry out an investigation of the paradoxes. We need to distinguish between those paradoxes which result from mistaking an expression for a significant sentence and the paradoxes which are generated by significant sentences. The first class is less interesting and less important. But to deal carefully with the second class, we must restrict our attention to valid proofs. We cannot expect a "proof" containing steps which are not significant sentences to help us understand or cope with significant paradoxical sentences. The preceding discussion of assertion, denial, and supposition has no place in the framework of generalized proofs. Real proofs have conclusions which are not simply sentences; the conclusions are illocutionary acts of asserting or denying sentences (or they are "suppositional" acts). Premisses leading to these conclusions are also asserted, denied, or supposed. We can only assert or deny a significant sentence; it makes no sense to speak of asserting an arbitrary expression.

One reason it is possible to think the study of generalized proofs is helpful for understanding real proofs is that in real proofs we do not ordinarily employ expressions to make illocutionary force explicit. When such expressions are supplied, they cannot be used with all the steps of generalized proofs. This makes it clear that generalized proofs have only a superficial resemblance to real ones. Constructing a deductive system which contains generalized proofs is a barrier to an effective study of logical (and semantic) paradoxes. There seems to be no compensating advantage.

4. SEMANTICS

Not much attention has been devoted to semantics in combinatory logic. (Constructing a model of a theory of combinators does not count as providing a semantic account for a fully explicit deductive system.) There are several difficulties in giving a semantic account for the language in a fully explicit deductive system. One difficulty is due to the fact that the language is not given prior to the deductive system. We do not begin with a class of significant expressions or significant sentences. Instead there is a large class of expressions; not all of them are significant. It may be necessary to establish results in the deductive system to determine which expressions are significant. This feature of fully explicit deductive systems precludes the standard semantic approach in which a semantic account is given prior to the deductive system; the deductive system then provides a syntactic characterization of semantically distinguished items. This first difficulty does not rule out a semantic account for a fully explicit deductive system; it simply calls for a different approach than the standard one.

A second barrier to giving a semantic account is that some predicates (and relation symbols) are most naturally interpreted as attributing syntactic properties to expressions, while others attribute properties to objects denoted by expressions. Seldin has suggested that confusing the two types of predicates may have led to inconsistent systems. It might be possible to avoid this difficulty by eliminating syntactic predicates. But it seems in keeping with the explicit character of a fully explicit deductive system to allow both kinds of predicates. They simply require different semantic treatments.

The most serious difficulty in providing a semantic account for fully explicit deductive systems is that the meanings of many expressions are determined by their roles in the deductive systems. There are no clear and distinct concepts which these expressions are intended to capture (or express). While it may be possible to provide informal explanations of the expressions' intended significance, these explanations are not specific enough to allow us to determine (in every case) that a rule is correct. Instead of using the significance of the expressions as a basis for judging the rules, the rules are used to give significance to the expressions.

It is plausible to identify the meaning of any expression with its "place" in a system of language. This place is partly determined by relations of entailment and incompatibility linking sentences which contain the expression to other sentences. It is partly determined by relations to experience or features of experience. (For some expressions their relations to experience are of great importance; for others, such as logical expressions, relations to experience are not important in determining their significance.) There is nothing "wrong" with an expression whose meaning is constituted by its role in a deductive system. In logic we ordinarily provide an independent account of the meaning we intend for an expression to have. This may be done by giving the truth conditions of sentences containing the expression; some other way may be possible. The independent characterization is sufficiently clear and precise that we can appeal to it to show that a rule is sound. We can appeal to a class of characterizations to show that a system is complete. When an independent characterization of intended meaning can't be given, a deductive system will have a decidedly experimental character. In setting up the system, we "make up" meanings as we go along. This is a trial-and-error procedure, for it is possible to give significance to expressions in such a way that a system is incoherent. It is also possible to give significance in such a way that a system is not useful. Even in such an experimental system, a preliminary semantic account is required. We must begin with some significant sentences before we can give meanings to other expressions by providing them with deductive roles. The development of an experimental system which gives meanings to various expressions is made more difficult if generalized proofs (other than valid proofs) are allowed. The significance of an expression may consist in its inferential role, but inferences can only be made with sentences or sentential formulas.

The third difficulty in providing a semantic account for a fully explicit deductive system effectively blocks a semantic account sufficient to support soundness and completeness results. But it is important to ask whether a system *should* contain expressions whose significance is given only by giving rules of inference. Perhaps it is always the best strategy to provide a clear, precise semantic account which is independent of a deductive system that codifies inferences. I cannot prove that this is the best strategy, but I would like to see the issue discussed by combinatory logicians.

REFERENCES

Kearns, J. T. (1977). The Logic of Calculation, *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik* 23, 45-58.

Seldin, J.P. (1976-77). Recent Advances in Curry's Program, *Rend. Sem. Mat. Univers. Politecn. Torino* 35, 77-88.

Seldin, J.P. (1979). Curry's Program. This volume.

THE NATURALNESS OF ILLATIVE COMBINATORY LOGIC AS A BASIS FOR MATHEMATICS

Martin W. Bunder
Mathematics Department
University of Wollongong
N.S.W. 2500
Australia

Dedicated to H.B. Curry on the occasion of his 80th Birthday

INTRODUCTION

The naturalness and simplicity of illative combinatory logic lies in 3 of its basic aspects: (1) The nature of the combinators. This allows the representation of the part of Mathematics (including logic) which is purely combinatorial by nature.
(2) The possibility of interpreting the application operator so that a statement $\vdash XY$ can stand for both "Y is an element of the class X" and "the predicate X holds of Y". The paradoxes can be avoided here at the cost of certain formation rules that can be expressed either within or outside the system. This identification of class notation and unary predicate calculus notation constitutes a major simplification in set theory in particular. (3) The use of the illative constant Ξ both as a subclass notation and a restricted form of universal quantification, and also the fact that Ξ can include significance considerations (see later). This last point means that a logic formulated using Ξ is more general than one using implication and universal quantification, while the rules and/or axioms required to specify Ξ are fewer, similar to and in a sense simpler than those required for $\supset, \sim$ and $(\forall x)$ in setting up predicate calculus. Finally these rules and axioms are sufficiently strong to give, in a natural manner, a large amount of set theory.

In this paper we will give an outline of the main concepts of illative combinatory logic while emphasising each of the above three points mentioned above.

FORMAL SYSTEMS

Curry has defined (in Chapter 1 §C of Curry (1963)) Mathematics as the science of formal methods. These would include methods of proof within particular formal systems as well as discussions regarding the relations of formal systems to one another. As we will be setting up combinatory logic as a basic formal system for mathematics, we briefly explain the notion of formal system. (For details see Curry and Feys (1958).)

A formal system is concerned with a set of objects. Usually it has certain primitive objects (constants and variables) and certain operations which allow the formation of new objects.

Next a formal system must contain certain primitive predicates and some rules which allow the formation of statements using the primitive predicates and objects.

Finally a set of statements is given as the set of axioms of the system and a set of rules which allow us to derive statements called theorems from the axioms.

COMBINATORY LOGIC AS A BASIC FORMAL SYSTEM

The Objects. The objects of any formal system of combinatory logic are called obs. The set of primitive obs includes the combinators K and S and may include other obs depending on the use to which the system is to be put. Then if X and Y are obs, so is (XY). This operation for forming new obs is called application.

We should note that the set of obs contains (and needs) no variables, but the system (i.e. the set of primitive obs) can be extended to include them.

We do however need what Curry and Feys (1958) calls U-variables. These are like variables in a meta language and are needed to state rules in this language about the system.

Let us now consider the terms and formulas of any arbitrary mathematical system. These are formed using the primitive constants and variables of the system, so if these constants are also primitive obs of the combinatory logic or can be defined within it, it is clear that all such terms and formulas are obs in a combinatory logic extended to include variables.

The Statements

The primitive predicates of combinatory logic are normally taken to be $\vdash$ (it is assertible that) and = (equals). If however we add a constant Q for equality to the list of primitive obs and add appropriate postulates for Q we can make do with $\vdash$ alone.

The statements of combinatory logic are of the form $\vdash X$ and $X = Y$ where X and Y are obs (or simply of the form $\vdash X$ where X is an ob).

If $\vdash$ is not one of the primitive predicates of a given mathematical system, we can introduce it as the only primitive predicate by translating the system into combinatory logic in the following way: if A is an arbitrary primitive n-ary predicate of the system which gives a statement $A(t_1,\ldots,t_n)$ when applied to terms $t_1,\ldots,t_n$, we replace this statement by $\vdash((\ldots(At_1)t_2\ldots)t_n)$[1.] in the combinatory logic, where A is an ob corresponding to the predicate A.

In many mathematical systems the rules which allow the formation of statements from the objects and predicates of the system are more restricted than the two (or more) for combinatory logic that are mentioned above. When however we want to base such a system on combinatory logic we can add to the list of postulates the rules or axioms which allow us to pick out the special statements of the applied system from those of the underlying combinatory logic.

The shifting of the formation rules of a system from its metatheory to the system itself has the advantage of generality in that objects, that are not in some sense well formed, can appear and also a theory with minimal metatheoretic restriction is easier to handle when for example proving consistency.

In this way any mathematical system can be represented in the notation of combinatory logic. This representation only depends on the primitive obs that the system has in addition to K and S and the postulates concerning these.

The Postulates

If = is a primitive predicate of the combinatory logic the rules of the system will include all the usual ones for equality such as symmetry, transitivity, replacement and:

$$\text{If } X = Y \text{ and } \vdash X \text{ then } \vdash Y \qquad \ldots (1)$$

Also the postulates will include the following axiom schemes which specify K and S:

$$KXY = X$$

$$SXYZ = XZ(YZ)$$

as well as

$$X = X$$

If the system has the constant Q and not the predicate =, the postulates become:

$$\text{If } \vdash QXY \text{ then } \vdash QYX \qquad \ldots (2)$$

$$\text{If } \vdash QXY \text{ and } \vdash QYZ \text{ then } \vdash QXZ$$

$$\text{If } \vdash QXY \text{ then } \vdash Q(UX)(UY) \qquad \ldots (3)$$

$$\text{If } \vdash QXY \text{ and } \vdash X \text{ then } \vdash Y \qquad \ldots (4)$$

$$\vdash Q(KXY)X$$

$$\vdash Q(SXYZ)(XZ(YZ))$$

$$\text{and } \vdash QXX \qquad \ldots (5)$$

The importance of the combinators K and S lies in the fact that given an ob X, possibly involving the free variable u, it allows us to define λu.X as the function which when applied to u gives us X. This is done as follows:

$$\lambda u.X = KX \quad \text{if} \quad u \text{ is not free in } X$$
$$\lambda u.X = I\ (=SKK) \quad \text{if} \quad X = u$$
$$\lambda u.(XY) = S(\lambda uX)(\lambda uY)$$

Thus if we let $C = \lambda z.\lambda y.\lambda x.xyz$
and $\Psi = \lambda w.\lambda z.\lambda y.\lambda x.x(yz)(yw)$
(these are also called combinators) we have:

$$CXYZ = XZY$$
and
$$\Psi XYZW = X(YZ)(YW).$$

These will allow us to rewrite (2) and (3) as:

$$\text{If} \quad \vdash QXY \quad \text{then} \quad \vdash CQXY \qquad \ldots (6)$$
and
$$\text{If} \quad \vdash KQUXY \quad \text{then} \quad \vdash \Psi QUXY.$$

When we examine the deductive rules in any mathematical system we find that, simple ones at least, follow the pattern of (2) and (3) in that the premise and conclusion share several common constants and U-variables (such as X and Y) but possibly in a different arrangement.

Using the combinators we have now rewritten (2) and (3) in such a way that the variables appear in exactly the same place and order in both the premise and conclusion so that the content of the rules is now contained in the constant parts of the statements involved and the U-variables X and Y have become appendages significant only in their number if at all. It is therefore reasonable to look for a means of leaving them out altogether, this we can do using the illative constant Ξ.

Before going onto Ξ however it is worth pointing out that natural numbers and the arithmetical operations can be defined in terms of pure combinatory logic (see Chapter 13 of Curry, Hindley and Seldin (1972)), however to define a set (or class) of all natural numbers Ξ is needed. Also, at the expense of using somewhat complicated metatheoretic definitions, it is possible to develop predicate calculus and some set theory using only an ob Q for equality and the predicate $\vdash$ in addition to pure combinatory logic (see Bunder (1978)).

THE CONSTANT OB Ξ

The primitive ob Ξ has a basic rule:

<u>Rule Ξ</u> If $\vdash \Xi XY$ and $\vdash XU$ then $\vdash YU$.

As we will interpret $\vdash XU$ as "U is an element of the class X" as well as "the predicate X holds of U" it is clear that $\vdash \Xi XY$ can be interpreted as "X is a subclass of Y" or as "YU holds for all U's in X".

The second interpretation allows far more generality than does any interpretation of $\vdash (\forall U)(XU \supset YU)$. In this latter case it is necessary for XU and YU to make sense (or be well formed) for all U's in the range of quantification, and that irrespective of the truth values of XU and YU for various U's $XU \supset YU$ must hold. In $\vdash \Xi XY$ we are only saying that $\vdash YU$ must hold whenever $\vdash XU$ holds, so that we are making no assumptions regarding the significance or truth of $XU \supset YU$ or YU when $\vdash XU$ does not hold or is not significant.

We now show how Ξ can allow us to do without variables in mathematics. (This is not of value in a practical sense as formulas without variables are generally harder to interpret, but the fact that a system can be represented without variables is valuable when various metatheorems are proved).

We can, for example, for a particular X write (6) as

$$\vdash \Xi (QX)(CQX),$$

and using combinators W and Φ with properties:

$$WXY = XYY$$

and $$\Phi XYZW = X(YW)(ZW)$$

as well as (5), we can write it as:

$$\vdash \Xi(WQ)(\Phi\Xi Q(CQ)). \qquad \ldots (7)$$

The use of Ξ allows us to replace all the rules for equality that we need except one ((1) or (4)) by axioms that contain no variables.

In a similar way the formation rules mentioned previously can be stated as axioms. In the case of propositional calculus for example, if we let "HX" stand for "X is a proposition" two of our formation rules could be:

$$X \vdash HX^{2}.$$

and
$$HX, HY \vdash H(X \supset Y).$$

These can be written as

$$\vdash \Xi IH$$

and
$$\vdash \Xi H(B(\Xi H)P)$$

where $X \supset Y$ is written as PXY and B is a combinator with the property

$$BXYZ = X(YZ).$$

Implication itself can be defined in terms of Ξ, using

$$PXY = \Xi(KX)(KY).$$

Modus ponens is then a special case of Rule Ξ.

Statements that are normally axioms of propositional calculus, such as

$$\vdash X \supset .\ Y \supset X \qquad \ldots (8)$$

become derivable rules in the system, such as:

$$HX, HY \vdash X \supset .\ Y \supset X \qquad \ldots (9)$$

or theorems such as:

$$\vdash \Xi H(B(B(\Xi H)(SP))P). \qquad \ldots (10)$$

The restrictions on X and Y in (9) are to be expected as the class of obs in combinatory logic is a much larger one than the class of propositions. Simple unrestricted propositional calculus-type statements such as (8) can lead to contradictions.

For ease of reading we will often write $X \supset_u Y$ for $\Xi(\lambda uX)(\lambda uY)$ and $X \supset Y$ for PXY. (10) then becomes:

$$\vdash Hx \supset_x .\ Hy \supset_y (y \supset .\ x \supset y) \qquad \ldots (11)$$

As an example of possible axioms for Ξ we list those of Bunder (1974):

$$\vdash Lx \supset_{x,y} :\ xu \supset_u .\ yuv \supset_v xu^{3} \qquad \ldots (12)$$

$$\vdash Lx \supset_{x,t} :\ xu \supset_u yu(tu) \supset_y .((xu \supset_u (yuv \supset_z zuv)) \supset_z (xu \supset_u zu(tu)) \qquad \ldots (13)$$

where L is a definable constant which can be considered to be the class of "significant" first order predicates.

The similarity between these and the usual axioms of intuitionistic implicational logic is obvious, the advantage over the axioms for universal quantification (which can be derived from them) is that no restrictions on bound and free variables are necessary as there are none.

Universal quantification $(\forall u)Zu$ can be defined by $\Xi(WQ)Z$, however rather than quantify over all obs it is usually convenient to quantify over a smaller class A of "individuals" so that $(\forall u)Zu$ is ΞAZ.

Conjunction, disjunction and negation can also be defined in terms of the combinators, Ξ and H (see Bunder (1974)), so that all the primitive terms of first order predicate calculus are present in this system.

(12) and (13) together with some axioms or rules for H such as the ones mentioned earlier are sufficient to give all of intuitionistic predicate calculus.

Classical predicate calculus can be obtained (less naturally perhaps) by adding an additional axiom. However, as Kuzichev (1973) and (1974) show, if multiple succedent sequents are used classical logic results as naturally as intuitionistic logic does from the above axioms or from rules for sequents with single succedents.

SOME FURTHER GENERALISATIONS

It may seem that when we introduced Ξ we were not being completely general when we considered only a rule involving one premise and one variable even though we were able to cover more general cases using (5) (see (6) and (7)).

If however we have a system without any universal class such as WQ and have to consider a more general form of rule which involves several premises and variables, we can use a generalised version of Ξ introduced in Bunder (1979a).

Another notion of fundamental importance in Mathematics is that of a function or mapping from one set or class to another. We can represent "Z is a function from X to Y" by "$\Xi X(BYZ)$", which we write as "FXYZ". Rule Ξ then gives us:

$$FXYZ, XU \vdash Y(ZU).$$

FXY is then the class of all functions from X to Y, so in particular, FAA is the class of all functions over individuals, FAH is the class of all functions from individuals to propositions (i.e. the class of first order predicates) and FHH is the class of unary propositional connectives. In (10) and (11) we used quantification over H, so it is not surprising that we can also quantify over FAH, FAA, FHH, F(FAH)H etc. and hence develop a predicate calculus of arbitrarily high order. This is done by taking various values such as FAH, FHH, F(FAH)H etc. for L in (12) and (13), (see Bunder (1978b)). A small part of such a higher order predicate calculus in turn forms a natural basis for set theory (see Bunder (1979c)).

CONCLUSION

Scott in his papers Scott (1975a,b) has stated that "there is as yet no combinatory logic" and "the combinators...have not yet given us an ultimate foundation...for logic...". Further he has stated in Scott (1973) that the claim for naturalness, at least of the system of combinatory logic Church (1932-3) was "awfully weak".

The system indicated above however is clearly a logic based on combinators, i.e. a combinatory logic, as are various systems due to Curry, Church, Fitch, Cogan, Seldin, Kuzichev, Goodman and Scott himself. (These systems of course vary in the degree of naturalness.)

It is of course debatable whether any system can ever be said to form an "ultimate foundation" for logic (or mathematics), but in the same way there is probably no ultimate foundation for set theory or category theory. On the other hand it seems natural that any foundation should be based on combinatory logic.

FOOTNOTES

1. We will from here on in this paper assume association to the left so that this ob will be written $At_1t_2 \ldots t_n$.

2. This is an abbreviation for "If $\vdash X$ then $\vdash HX$", generally "$X_1, X_2, \ldots X_n \vdash Y$" is an abbreviation for "If $\vdash X_1, \vdash X_2 \ldots$ and $\vdash X_n$ then $\vdash Y$".

3. "$X \supset_{u,v} Y$" is an abbreviation for "$WQv \supset_v \lambda v(X \supset_u Y)$".

REFERENCES

Bunder, M.W., (1974). Propositional and predicate calculuses based on combinatory logic, *Notre Dame Journal of Formal Logic*, Vol. XV, pp. 25-34.

Bunder, M.W., (1978a). Predicate calculus of arbitrarily high order, *The University of Wollongong, Department of Mathematics Preprint Series* No. 2/78.

Bunder, M.W., (1978b). Predicate calculus and naive set theory based on pure combinatory logic, *The University of Wollongong, Department of Mathematics Preprint Series* No. 12/76. To appear in *Archiv fur Mathematische Logik und Grundlagenforschung*.

Bunder, M.W., (1979a). Generalised restricted generality, *Notre Dame Journal of Formal Logic*, Vol. XX, pp. 620-624.

Bunder, M.W., (1979b). A one axiom set theory based on higher order predicate calculus, *The University of Wollongong, Department of Mathematics Preprint Series* No. 3/79.

Bunder, M.W., (1979c). Set theory in predicate calculus with equality, *The University of Wollongong, Department of Mathematics Preprint Series* No. 8/79.

Church, A., (1932-3). A set of postulates for the foundations of logic, *Annals of Mathematics*, Vol. 33, (1932), pp. 346-366. and Vol. 34 (1933), pp. 839-864.

Curry, H.B., (1963). "Foundations of Mathematical Logic", McGraw-Hill, New York.

Curry, H.B., and Feys, R., (1958). "Combinatory Logic, Vol. 1", North Holland, Amsterdam.

Curry, H.B., and Hindley, J.R., and Seldin, J.P. (1972). "Combinatory Logic Vol. 2", North Holland, Amsterdam.

Kuzichev, A.S., (1973). Consistent extensions of pure combinatory logic, *Moscow Universitet Mathematics Bulletin*, Vol. 28 (3), pp. 66-72.

Kuzichev, A.S., (1974). Deductive operations of combinatory logic, *Moscow Universitet Mathematics Bulletin*, Vol. 29 (3), pp. 8-14.

Scott, D.S., (1973). Models for various type-free calculi, *In* Suppes et al. "Logic, Methodology and Philosophy of Science IV", Amsterdam, pp. 157-188.

Scott, D.S., (1975a). "Combinators and classes" *In* Springer Lecture Notes in Computer Science, No. 37, pp. 1-26.

Scott, D.S., (1975b). "Some philosophical issues concerning theories of combinators" *In* Springer Lecture Notes in Computer Science, No. 37, pp. 346-366.

FOUNDATIONAL THEORIES

CONSISTENCY OF THE UNRESTRICTED ABSTRACTION PRINCIPLE USING AN INTENSIONAL EQUIVALENCE OPERATOR

by

Peter Aczel* and Solomon Feferman§

**Department of Mathematics*
The University of Manchester M13 9PL England

§Departments of Mathematics and Philosophy
Stanford University Stanford Ca 94305 USA

Dedicated to H.B. Curry on the occasion of his 80th birthday and to F.B. Fitch for his 70th birthday.

"Le paradoxe n'est pas toujours une faussetế" - Diderot

1. INTRODUCTION The usual explanation of '$\{x|\phi(x)\}$' is that it is supposed to denote a class whose members are just those y for which $\phi(y)$ holds. This is here regarded as an *explanation by definition*. Instead of expressing the unrestricted abstraction (comprehension) principle in the usual inconsistent form

(1) $y \in \{x|\phi(x)\} \longleftrightarrow \phi(y)$

we express it here in the form

(2) $y \in \{x|\phi(x)\} \equiv \phi(y)$.

This makes use of a new connective $\equiv$ in place of the truth-functional biconditional $\longleftrightarrow$. In general, $(\phi \equiv \psi)$ is read as expressing that ϕ *is equivalent to* ψ *in consequence of given basic definitions* (in this case (2)). A logic is presented here for $\equiv$ which is simple and plausible under this reasoning. For example, $\equiv$ acts as an equivalence relation which preserves all the operators of the language; furthermore, the principle

(3) $(\phi \equiv T) \rightarrow \phi$

is taken, where T is an identically true statement. This allows us to draw ordinary conclusions from certain statements of equivalence. The full list of laws for $\equiv$ will be found in 2.1 below. It is convenient there to write $T(\phi)$ for $\phi \equiv T$ and $F(\phi)$ for $\phi \equiv F$ where $F = \neg T$. Then we write $D(\phi)$ for $T(\phi) \vee F(\phi)$ and (informally) call ϕ *determinate* if $D(\phi)$ holds, otherwise *indeterminate*.

With each system S in ordinary logic is associated a system S* based on the laws of $\equiv$ and the abstraction principle (AP) of (2) above, together with S. The main result is that: *if* S *has only infinite models then* S* *is a conservative extension of* S. As a corollary one has that: *for empty* S *the theory* S* *is consistent.* (The assumption that S has only infinite models can be dropped using a simple variant of S*.)

There is nothing to prevent us from positing basic definitions that are inherently self-contradictory. Indeed, pursuit of the Russell paradox in S* easily leads to a sentence ϕ for which

(4) $\phi \equiv \neg \phi$

is provable. Nevertheless the main theorem shows that (4) does not have contradictory consequences. The reason is that ϕ is indeterminate in the sense explained above. This example is discussed further at the end of §2. A consequence is that $\equiv$ is an *intensional* operator, i.e. we do *not* have closure under the rule: from $(\phi \equiv \psi)$, $(\phi \leftrightarrow \phi')$, $(\psi \leftrightarrow \psi')$ infer $(\phi' \equiv \psi')$.

The system S* is due to Feferman. It was first presented in a talk he gave at Yale University (April 1, 1977) for a Colloquium in honor of Professor F.B. Fitch on the occasion of his retirement at the age of 70.[1] The proof of the main result has evolved through several stages. A significant error was found by Aczel in the first detailed proof which had been given by Feferman.[2] This led the latter to a revision of his argument, but only of a weaker result (namely conservativeness of a system without function symbols or abstracts, based on the comprehension principle in the form $\exists a \forall x[x\epsilon a \equiv \phi(x)]$ for arbitrary ϕ). The proof given here of the full result makes use of substantial improvements and simplifications found by Aczel.

The contents of the paper are as follows. §2 gives the laws of $\equiv$ and the syntax of the language $\mathcal{L}^*$ of $S^* = S + (AP)$. §3 contains a Church-Rosser (CR) Theorem for a general class of infinitary languages $\mathcal{L}_M(P)$ with reduction relation $\geq$ between sentences. Here P is a set of atomic sentences and the sentences of $\mathcal{L}_M(P)$ are built up by the connectives, $\neg$, $\equiv$ and $\bigwedge_M$ where M is a given set. The reduction relation is generated from basic reductions $p \geq \chi_p$ $(p\epsilon P)$. In consequence of the CR theorem, it is shown that the interpretation of $\phi \equiv \psi$ as $\exists\theta[\phi \geq \theta$ and $\psi \geq \theta]$ satisfies the laws of $\equiv$ in $\mathcal{L}_M(P)$. Finally in §4, given any model $\mathfrak{M} = (M, \ldots)$ of S, we take P to be the set of atomic sentences $(\bar{m}\epsilon\bar{n})$ of $\mathcal{L}^*(M)$ (which contains a constant $\bar{m}$ for each $m\epsilon M$). Then $\mathcal{L}^*(M)$ is translated into $\mathcal{L}_M(P)$ by a map $\phi \mapsto \phi^+$ and we take as basic reductions (essentially) $(\bar{m} \ \epsilon \ \{x|\phi(x)\})^+ \geq \phi(\bar{m})^+$ for $m\epsilon M$. This serves to satisfy the Abstraction Principle as well.

The paper concludes with two Appendices. The first one, §5, is written by Feferman. 5.1 presents his reasons for being interested in systems like S* and his views of their possible utility. In particular, the lack in S* of the extensionality axiom for ϵ is argued not to be disturbing. 5.2 discusses what may be considered to be a peculiar feature of S*. Finally in 5.3 there is raised the question whether the main result can be derived from a CR theorem for a suitable system of λ-calculus. A definite obstacle to this is found in the work of Klop (1977). For this reason, the CR theorem given in §3 for $\mathcal{L}_M(P)$ should be of independent interest.

The second appendix, §6 is written by Aczel. It contains a simplifying reformulation of the notion of interpretation for $\mathcal{L}^*$ used in §4. This reformulation came too late to be incorporated into the main body of the paper. It involves an auxiliary semantics, where sentences are assigned values in a set Ω of possible truth values as well as the ordinary two-valued semantics. The auxiliary semantics is used to interpret $\equiv$ in the ordinary semantics as follows: $\phi \equiv \psi$ is true if and only if ϕ and ψ have the same value in the auxiliary semantics. 6.3 and 6.4 reformulate the results of §3 and §4 respectively. Finally, in 6.5 a modification of the main theorem is proved. The modification involves a change in one of the $\equiv$-laws according to which $T(\phi \equiv \psi)$ holds whenever $\phi \equiv \psi$; the modified law requires $T(\phi \equiv \psi)$ to hold only when ϕ, ψ are determinate and $\phi \equiv \psi$. Thus $T(y \epsilon \{x | \phi(x)\} \equiv \phi(y))$ does not always hold in the modified system, though it does in the original S^*. It turns out that the main theorem can be proved for the modified system without the use of the Church-Rosser argument of §3 and instead by a fixed point construction for a three-valued semantics.

Because of the way the paper evolved it has been dedicated to both Professors Curry and Fitch. We wish to express our great admiration for the rare and persistent optimism and determination shown by each of them in pursuit of his life's work.

§2. A THEORY OF CLASSES WITH $\equiv$

2.1 THE LAWS OF $\equiv$. Let $\mathcal{L}$ be any given countable language in (single-sorted) 1st order predicate calculus with equality. The logic is assumed to be classical, and the basic logical connectives are taken to be $\neg$, $\rightarrow$ and $\forall$ with the remaining operators $\wedge$, $\vee$, $\leftrightarrow$, $\exists$ defined from these as usual. T denotes a fixed identically true sentence of $\mathcal{L}$, e.g. $T = (\theta \rightarrow \theta)$ for some closed θ; F is defined as $\neg T$.

We shall consider a language $\mathcal{L}^*$ which extends $\mathcal{L}$ both in basic vocabulary and by adjunction of a new binary operator E on formulas. Thus whenever ϕ, ψ are formulas of $\mathcal{L}^*$ so also is $\phi \mathrm{E} \psi$; we shall write $(\phi \equiv \psi)$ for $\phi \mathrm{E} \psi$. In the specific cases that interest us, this will serve to express equivalence by definition. We also write $(\phi \not\equiv \psi)$ for $\neg(\phi \equiv \psi)$.

Let S* be a set of formulas in $\mathcal{L}^*$. The relation $S^* \vdash \phi$ is defined by augmenting the axioms and rules of the predicate calculus (now to be applied to arbitrary formulas of $\mathcal{L}^*$) by the laws (1)-(4) of $\equiv$ given below.

In these we use the abbreviations

$$T(\phi) =_{\text{def}} (\phi \equiv T)$$
$$F(\phi) =_{\text{def}} (\phi \equiv F)$$
$$D(\phi) =_{\text{def}} (T(\phi) \vee F(\phi)).$$

Again, in the cases that interest us, $T(\phi)$ and $F(\phi)$ mean that ϕ reduces by definition to truth and falsity respectively. $D(\phi)$ is read: ϕ is *determinate*. While ordinary logic gives $[(\phi \leftrightarrow T) \vee (\phi \leftrightarrow F)]$ for every ϕ, we should not expect $D(\phi)$ to hold in general. Furthermore, when a sentence ϕ happens to be true in an interpretation of $\mathcal{L}^*$, we should not expect $T(\phi)$ to be true, i.e. $\phi \rightarrow T(\phi)$ should not be expected to be generally valid. The axioms will guarantee that $D(\phi)$ and $(\phi \leftrightarrow T(\phi))$ hold for every $\mathcal{L}$-formula ϕ. However in general we can only assert $T(\phi) \rightarrow \phi$.

(1) $\equiv$ is an equivalence relation.

(2) $\equiv$ is preserved by $\neg$, $\rightarrow$, $\equiv$ and $\forall$.

(3) (i) $T(\theta) \leftrightarrow \theta$, if θ is atomic or is F.

(ii) $F(\theta) \leftrightarrow \neg\theta$, if θ is an atomic $\mathcal{L}$-formula.

(4) (i) $T(\neg\phi) \leftrightarrow F(\phi)$

(ii) $F(\neg\phi) \leftrightarrow T(\phi)$

(iii) $T(\phi \rightarrow \psi) \leftrightarrow D(\phi) \wedge D(\psi) \wedge (F(\phi) \vee T(\psi))$

(iv) $F(\phi \rightarrow \psi) \leftrightarrow T(\phi) \wedge F(\psi)$

(v) $T(\forall x \phi(x)) \leftrightarrow \forall x T(\phi(x))$

(vi) $F(\forall x\phi(x)) \leftrightarrow \forall x(D(\phi(x))) \wedge \exists xF(\phi(x))$

(vii) $T(\phi \equiv \psi) \leftrightarrow (\phi \equiv \psi)$

(viii) $F(\phi \equiv \psi) \leftrightarrow D(\phi) \wedge D(\psi) \wedge (\phi \not\equiv \psi)$.

The following explains (1) and (2) in more detail. (1) consists of the schemata $(\phi \equiv \phi)$, $(\phi \equiv \psi) \to (\psi \equiv \phi)$ and $(\phi \equiv \psi) \wedge (\psi \equiv \theta). \to (\phi \equiv \theta)$. For (2) let O be one of the n-ary ($n = 1,2$) propositional operations $\neg$, $\to$ or $\equiv$. The statement that $\equiv$ is preserved by O is given by the scheme:

(2a) $(\phi_1 \equiv \psi_1) \wedge \dots \wedge (\phi_n \equiv \psi_n) \to O(\phi_1, \dots, \phi_n) \equiv O(\psi_1, \dots, \psi_n)$.

The statement that $\equiv$ is *preserved by* $\forall$ is given by the scheme:

(2b) $\forall x(\phi(x) \equiv \psi(x)) \to [(\forall x\phi(x)) \equiv (\forall x\psi(x))]$.

LEMMA. The following are theorems of S*:

(5) $F \not\equiv T$

(6) $(\neg F) \equiv T$

(7) $D(\theta)$ for each atomic θ of $\mathcal{L}$

(8) D is strongly closed w.r. to $\neg$, $\to$ and $\forall$, i.e.

$$D(\neg\phi) \leftrightarrow D(\phi),$$
$$D(\phi \to \psi) \leftrightarrow D(\phi) \wedge D(\psi), \text{ and}$$
$$D(\forall x\phi(x)) \leftrightarrow \forall xD(\phi(x))$$

(9) (i) $T(\theta) \to \theta$

(ii) $F(\theta) \to \neg\theta$.

Proof. (5), (6), (7) and (8) are immediate consequences of (3)(i), 4(i), (3) and (4)(i)-(vi) respectively. For (9),(i) and (ii) are proved simultaneously by induction on θ. For atomic θ use (3)(i) and for non-atomic θ use the appropriate part of (4) and the induction hypothesis.

When $D(\phi)$ holds we say that the *truth-value of* ϕ *under* $\equiv$ *is* T *or* F according as $T(\phi)$ or $F(\phi)$ holds (only one of which can happen by (5)). Thus the truth-value of determinate ϕ under $\equiv$ is the same as its ordinary truth-value. Moreover, by (4), this is related to the truth-values of its immediate subformulas in the standard way when ϕ has outermost operation $\neg$, $\to$ or $\forall$.

Finally, for determinate ϕ, ψ, the truth-value of $(\phi \equiv \psi)$ is T when ϕ, ψ have the same truth-value, otherwise F. Note that for ϕ atomic we have $T(\phi) \leftrightarrow \phi$ by (3)(i); however this does not tell us that ϕ is determinate unless ϕ is true.

2.2 THE LANGUAGE $\mathcal{L}$ *. The specific extension we use is obtained from $\mathcal{L}$ by adjoining a binary relation symbol ϵ and an abstraction operator $\{\cdot|-\}$ as well as the operator E on formulas. Thus *formulas of* $\mathcal{L}$ * are generalised simultaneously as follows:

(1) every variable is a term

(2) if f is an n-ary function symbol of $\mathcal{L}$ $(n \geq 0)$ and $t_1, \ldots, t_n$ are terms then $f(t_1, \ldots, t_n)$ is a term

(3) if ϕ is a formula and x a variable then $\{x|\phi\}$ is a term

(4) if R is an n-ary relation symbol of $\mathcal{L}$ and $t_1, \ldots, t_n$ are terms then $R(t_1, \ldots, t_n)$ is an (atomic) formula

(5) if s, t are terms then $(s \epsilon t)$ and $(s = t)$ are (atomic) formulas

(6) if ϕ, ψ are formulas then so also are

$$\neg\phi,\ \phi \rightarrow \psi,\ \phi \equiv \psi \text{ and } \forall x\phi.$$

The terms $\{x|\phi\}$ are called *abstracts*.

By the *Abstraction Principle* we mean the following scheme in $\mathcal{L}$ *:

(AP) $\quad \forall y\, [y \epsilon \{x|\phi(x)\} \equiv \phi(y)]$

for arbitrary ϕ. Here and below we use the following notations and conditions for substitution. $\phi(x)$ is supposed to be a formula ϕ presented with specified free occurrences of x (possibly none); ϕ may contain additional free variables. In forming $\phi(t)$ where t is a term, it is assumed that t is free for x at all the given occurrences, i.e. no variable y occurring free in t will be substituted in the scope of a quantifier $\forall y$ in ϕ.

2.3 STATEMENT OF THE MAIN RESULT Let S be any set of sentences in $\mathcal{L}$. We define

$$S^* = S + (AP)$$

MAIN THEOREM If S has only infinite models then S* is a conservative extension of S.

COROLLARY If S has an infinite model then S* is consistent. In particular, (AP) is consistent (in the logic of $\equiv$).

The proof of the main theorem will occupy §§3 and 4. The reason for the hypothesis that S has only infinite models can be explained simply as follows. Given any model $\mathfrak{M}$ of S we shall expand $\mathfrak{M}$ to a model $\mathfrak{M}^*$ of S*, i.e. the interpretation of $\mathcal{L}$ is unchanged. In particular the domain M will remain fixed. To define $\mathfrak{M}^*$ we must assign values to abstracts $\{x|\phi\}$ in M; the method of proof requires this assignment to be one-one on a certain infinite class of formulas ϕ.[3] This is also the source of the reason why the axiom of extensionality cannot be included as well. (cf. 5.1 for further discussion of that).

Naturally the first question relative to the above consistency results is to ask how trouble is avoided with the Russell class. Let $r = \{x|x \notin x\}$. Then (AP) gives

$$\forall x[x \in r \equiv x \notin r]$$

so $(r \in r \equiv r \notin r)$. For the atomic sentence $\phi = (r \in r)$ it follows that ϕ is not determinate, for otherwise we would have $T \equiv F$. Thus there is no direct contradiction.

2.4 THE INTENSIONALITY OF $\equiv$. Let $\phi = (r \in r)$ with $r = \{x|x \notin x\}$ as just defined. Since ϕ is atomic we have $\phi \leftrightarrow (\phi \equiv T)$. We shall use this to prove $\neg\phi$ (from (AP) alone). For if ϕ holds then $(\phi \equiv T)$. But $\phi \equiv \neg\phi$ so $(\neg\phi) \equiv T$ and then $(\neg\neg\phi) \equiv F$. From $(\neg\phi) \equiv (\neg\neg\phi)$ we conclude that $T \equiv F$, which contradicts 2.1(5).

From provability of $\neg\phi$ it also follows that the operation $\equiv$ is *intensional*, i.e. we do not have closure under the rule of inference

$$\frac{(\phi \equiv \psi),\quad (\phi \leftrightarrow \phi'),\quad (\psi \leftrightarrow \psi')}{\phi' \equiv \psi'.}$$

For the counter-example take

$$\frac{(\phi \equiv \neg\phi),\quad (\phi \leftrightarrow \phi),\quad (\neg\phi \leftrightarrow T)}{\phi \equiv T\,.}$$

The conclusion fails because otherwise we would derive ϕ and hence an inconsistency.

§3. A CHURCH-ROSSER THEOREM FOR INFINITARY LANGUAGES WITH AN EQUIVALENCE OPERATOR.

3.1 MOTIVATION As indicated in the introduction, the method of proof of our main result will proceed as follows. With each model $\mathfrak{M}$ of S in $\mathcal{L}$ is to be associated an expansion $\mathfrak{M}^*$ of $\mathfrak{M}$ satisfying S*(in $\mathcal{L}^*$). An infinitary language $\mathcal{L}_M$ (P) will be defined, having a symbol $\bar{m}$ for each $m \in M$. $\mathcal{L}^*$ is embedded in $\mathcal{L}_M(P)$ by translating $\forall x\, \phi(x)$ into $(\forall x\, \phi(x))^+ = \bigwedge_{m \in M} \phi(\bar{m})^+$. A reduction relation $\phi \geq \psi$ will be defined among formulas of $\mathcal{L}_M(P)$. The basic reductions are $(\bar{m} \in \bar{n}) \geq \phi(\bar{m})^+$ for $m, n \in M$ and $n = \{x | \phi(x)\}_M$. It is shown by the results of this section that $\geq$ has the Church-Rosser property. Then we define $\phi \equiv \psi$ to hold if there exists θ with $\phi \geq \theta$ and $\psi \geq \theta$. Finally, $(\phi \,E\, \psi)$ is taken to hold in $\mathcal{L}^*$ when $\phi^+ \equiv \psi^+$. This will be verified to have the properties of the laws of $\equiv$ together with the Abstraction Principle, in consequence of the basic reductions. To ensure that $(\phi \,E\, \phi)$ always holds in the model we also take $(\phi \,E\, \phi) \geq T$ as a second basic reduction.

Remark. The situation for the Church-Rosser property here is analogous to that with the interpolation theorem, which fails for ω-logic but works for logic with infinitely long conjunctions. For if $\phi(\bar{m}) \geq \psi_0(\bar{m})$, $\phi(\bar{m}) \geq \psi_1(\bar{m})$ for each m and we know the CR property holds for all these triples we can only assume existence of a θ_m with $\psi_0(\bar{m}) \geq \theta_m$ and $\psi_1(\bar{m}) \geq \theta_m$. The "diamond" for $\forall x\, \phi(x)$ has to be completed by $\bigwedge_{m \in M} \theta_m$.

3.2 THE LANGUAGE $\mathcal{L}_M(P)$ It turns out that the required Church-Rosser theorem can be derived from a more general one for a language $\mathcal{L}_M(P)$ whose sentences are built up from an arbitrary set P of atomic sentences together with T, F, and where one takes *any* set of principal reductions $p \geq \chi_p$. The operations of $\mathcal{L}_M(P)$ are $\neg$, E and $\bigwedge_M$ where M is any fixed set with at least two elements.

In describing the generation of sentences of $\mathcal{L}_M(P)$, it is convenient to deal separately with the *truth-functional sentences*, i.e. those whose only atoms are T and F. Thus the sentences of $\mathcal{L}_M(P)$ are generated as follows:

(1) T and F are sentences

(2) each $p \in P$ is a sentence

(3) if ϕ is a sentence different from T, F then $\neg\phi$ is a sentence

(4) if ϕ, ψ are sentences and $\{\phi\ \psi\} \nsubseteq \{T,F\}$ then $\phi E \psi$ is a sentence

(5) if ϕ_m is a sentence for each $m \in M$ and $\{\phi_m\}_m \nsubseteq \{T,F\}$ then $\bigwedge_m \phi_m$ is a sentence.

We here write $\{\phi_m\}_m$ for $\{\phi_m\}_{m \in M}$ and $\bigwedge_m \phi_m$ for $\bigwedge_{m \in M} \phi_m$. Each sentence is uniquely obtained by one of (1)-(5).

Now the operations $\neg$, E, $\bigwedge_M$ are extended to arbitrary sentences as follows:

(6) $(\neg T) = F$ and $(\neg F) = T$

(7) $(T E T) = (F E F) = T$ and $(T E F) = (F E T) = F$

(8) $\bigwedge_m \phi_m = T$ if each $\phi_m = T$
and $\bigwedge_m \phi_m = F$ if each $\phi_m = T$ or F and some $\phi_m = F$.

3.3 THE REDUCTION RELATION Fix a sentence χ_p for each $p \in P$. We also write p^0 for χ_p. The relation $\geq$ is the least reflexive, transitive relation between sentences which makes $p \geq p^0$ $(p \in P)$ and $(\phi E \phi) \geq T$ (each ϕ) and which preserves $\neg$, E and $\bigwedge_M$. In other words, $\geq$ is generated by the following rules:

(1) $p \ge p^0$ $(p \in P)$

(2) (i) $\phi \ge \phi$

(ii) $\phi \ge \psi$ and $\psi \ge \theta \Rightarrow \phi \ge \theta$

(3) $\phi \ge \phi' \Rightarrow \neg\phi \ge \neg\phi'$

(4) $\phi_i \ge \phi'_i$ $(i = 0,1) \Rightarrow (\phi_0 \mathrm{E} \phi_1) \ge (\phi'_0 \mathrm{E} \phi'_1)$

(5) $\phi_i \ge \phi'_i$ $(i = 0,1)$ and $\phi'_0 = \phi'_1 \Rightarrow (\phi_0 \mathrm{E} \phi_1) \ge \mathrm{T}$

(6) $\phi_m \ge \phi'_m$ $(m \in M) \Rightarrow \bigwedge_m \phi_m \ge \bigwedge_m \phi'_m$.

Note that there is no restriction on the sentences involved, i.e. those which appear in the conclusions of (3)-(5) may be truth-functional combinations. Also (5) is used in place of $(\phi \mathrm{E} \phi) \ge \mathrm{T}$ for the purposes below; the resulting relation is of course the same.

LEMMA If $\phi \ge \theta$ and ϕ is T or F then $\phi = \theta$.

Proof By induction on the generation of $\ge$. As just noted, $\mathrm{T} \ge \theta$, resp. $\mathrm{F} \ge \theta$ may appear as the conclusion of any of (2)-(6). In case (2), say $\mathrm{T} \ge \theta$ by $\mathrm{T} \ge \psi$ and $\psi \ge \theta$; then $\mathrm{T} = \psi$ and hence $\mathrm{T} = \theta$ by I.H. In case (3), if $(\neg\phi) = \mathrm{T}$ this can only be if $\phi = \mathrm{F}$ so then $\phi' = \mathrm{F}$ and $\neg\phi' = \mathrm{T}$; similarly if $(\neg\phi) = \mathrm{F}$. In cases (4) and (5) if $(\phi_0 \mathrm{E} \phi_1) = \mathrm{T}$ this can only happen if ϕ_0, ϕ_1 are both T or both F, so then the same holds for ϕ'_0, ϕ'_1. The remaining cases are treated similarly.

3.4 CHURCH-ROSSER PROPERTIES Let $\succcurlyeq$ be any relation between sentences. We define $\phi \succcurlyeq (\theta', \theta'')$ to mean $\phi \succcurlyeq \theta'$ *and* $\phi \succcurlyeq \theta''$; similarly $(\theta', \theta'') \succcurlyeq \psi$ means $\theta' \succcurlyeq \psi$ *and* $\theta'' \succcurlyeq \psi$. By CR($\succcurlyeq$) is meant the property:

$$\forall\phi, \theta', \theta''[\phi \succcurlyeq (\theta', \theta'') \Rightarrow \exists\psi(\theta', \theta'') \succcurlyeq \psi]$$

This is depicted by solution of the diagram

$$\begin{tikzcd} & \phi \arrow[dl, no head] \arrow[dr, no head] & \\ \theta' \arrow[dr, dashed, no head] & & \theta'' \arrow[dl, dashed, no head] \\ & \psi & \end{tikzcd}$$

TC($\succcurlyeq$) denotes the reflexive transitive closure of $\succcurlyeq$, so ϕ is in the TC($\succcurlyeq$) relation to θ just in case there exist $\theta_1, \ldots, \theta_k$

with

$$\phi = \theta_1 \succcurlyeq \ldots \succcurlyeq \theta_k = \theta$$

The following is readily proved:

LEMMA

(i) If $CR(\succcurlyeq)$ then $CR(TC(\succcurlyeq))$

(ii) If $\{\succcurlyeq_i\}_{i \in I}$ is a chain of relations ordered by inclusion and $CR(\succcurlyeq_i)$ holds for each $i \in I$ then $CR(\bigcup_{i \in I} \succcurlyeq_i)$.

3.5 LAYERED REDUCTIONS Our aim is to prove $CR(\geq)$, where $\geq$ is the reduction relation for $\mathcal{L}_M(P)$ defined in 3.3. The method of proof will be to stratify $\geq$ into layers $\geq_\alpha$ and prove $CR(\geq_\alpha)$ by induction on α. The layering follows roughly the ordinal stages of the inductive definition; however reflexive transitivity is dealt with separately by alternating reflexive transitive closure with the other generation steps. This method is similar to that used by Maass (1975) to prove the CR property for a system of infinite terms in the λ-calculus. However, the principal reductions $\phi E \phi \geq T$ cause problems here which do not permit one to follow his arguments too closely.

Assume $\geq_\beta$ has been defined for each $\beta < \alpha$ where α is an ordinal. Let $(\to_\alpha) = TC(\bigcup_{\beta < \alpha}(\geq_\beta))$. Then generate $\geq_\alpha$ by the following rules:

(1) $p \geq_\alpha p^0$

(2) $\phi \to_\alpha \psi \quad \Rightarrow \quad \phi \geq_\alpha \psi$

(3) $\phi \to_\alpha \phi' \quad \Rightarrow \quad \neg\phi \geq_\alpha \neg\phi'$

(4) $\phi_i \to_\alpha \phi_i' \ (i = 0,1) \quad \Rightarrow \quad (\phi_0 E \phi_1) \geq_\alpha (\phi_0' E \phi_1')$

(5) $\phi_i \to_\alpha \phi_i' \ (i = 0,1)$ and $\phi_0' = \phi_1' \quad \Rightarrow \quad (\phi_0 E \phi_1) \geq_\alpha T$

(6) $\phi_m \to_\alpha \phi_m' \ (m \in M) \quad \Rightarrow \quad \bigwedge_m \phi_m \geq_\alpha \bigwedge_m \phi_m'$.

The following is easily proved.

LEMMA

(i) If $\beta < \alpha$ then $\geq_\beta$ is contained in $\geq_\alpha$.

(ii) If $\phi \to_\alpha \phi'$ then for some $\phi_1',\ldots,\phi_k'$ and some $\beta < \alpha$ we have $\phi = \phi_1' \geq_\beta \ldots \geq_\beta \phi_k' = \psi'$

(iii) The relation $\geq$ is $\cup_\alpha(\geq_\alpha)$.

3.6 ANALYSIS OF REDUCTION POSSIBILITIES To make up for the loss of refined control provided by Maass' method we add an analysis of possibilities in $\geq_\alpha$. The idea for this goes back to Girard (1973), which first established CR for the λ-calculus with infinite terms (without layered reductions).

LEMMA (A.P. Theorem for $\geq_\alpha$) Suppose $\phi \geq_\alpha \theta$. Then

(i) if $\phi = T$ or $\phi = F$ then $\theta = \phi$;

(ii) if $\phi = p$ $(p \in P)$ then $\theta = p$ or $p^0 \to_\alpha \theta$;

(iii) if $\phi = \neg\phi_0$ then there exists ϕ_0' with $\phi_0 \to_\alpha \phi_0'$ and $\theta = \neg\phi_0'$;

(iv) if $\phi = (\phi_0 \mathrm{E} \phi_1)$ then there exist $\phi_i'(i=0,1)$ with $\phi_i \to_\alpha \phi_i'$ and either $\theta = (\phi_0' \mathrm{E} \phi_1')$ or else $\phi_0' = \phi_1'$ and $\theta = T$;

(v) if $\phi = \bigwedge\!\!\bigwedge_m \phi_m$ then there exist ϕ_m' $(m \in M)$ with $\phi_m \to_\alpha \phi_m'$ and $\theta = \bigwedge\!\!\bigwedge_m \phi_m'$.

Proof Note that the result is clear if $\phi = \theta$. So assume $\phi \neq \theta$. (i) is immediate from the lemma of 3.3 since $\geq_\alpha$ is contained in $\geq$. For the remaining cases we induct on α.

(ii) $p \geq_\alpha \phi'$ can hold only by (1) and (2) in the definition (3.5) of $\geq_\alpha$. In case (1) the conclusion is immediate. In case (2) we have $p \to_\alpha \theta$ so there exist $\theta_1,\ldots,\theta_k$ and $\beta < \alpha$ with $p = \theta_1 \geq_\beta \ldots \geq_\beta \theta_R = \theta$. As $p \neq \theta$, $k > 1$ and so by I.H. we have $\theta_2 = p$ or $p \to_\beta \theta_2$. In the latter case $p^0 \to_\alpha \theta$. In case $\theta_2 = p$ we can apply induction on k to get the desired conclusion. (iii) $\phi_0 \geq_\alpha \theta$ can hold only by (2) or (3) of 3.5. The conclusion is immediate for (3). In case (2) we have some $\theta_1,\ldots,\theta_k$ and $\beta < \alpha$ with

$$\neg\phi_0 = \theta_1 \geq_\beta \ldots \geq_\beta \theta_k = \theta.$$

As $\phi_0 \neq \theta$, $k > 1$, so that by I.H. there exists ϕ_1 with $\theta_0 \to_\beta \phi_1$ and $\theta_2 = \neg\phi_1$. We can apply induction on k to get the conclusion for θ_k.

(iv) $(\phi_0 \mathrm{E} \phi_1) \geq_\alpha \theta$ can hold only by (2), (4) or (5). Cases (4) and (5) are immediate. In case (2) there exists $\theta_1,\ldots,\theta_k$ and $\beta < \alpha$ with

$$(\phi_0 \mathrm{E} \phi_1) = \phi_1 \geq_\beta \dots \geq_\beta \theta_k = \theta.$$

As $(\phi_0 \mathrm{E} \phi_1) \neq \theta$, $k > 1$ so that by I.H. there exist $\tilde{\phi}_i$ $(i = 0,1)$ with $\phi_i \to_\beta \tilde{\phi}_i$ and either $\theta_2 = \tilde{\phi}_0 \mathrm{E} \tilde{\phi}_1$ or $\tilde{\phi}_0 = \tilde{\phi}_1$ and $\theta_2 = \mathrm{T}$. In the latter case $\theta_k = \mathrm{T}$ and we have the conclusion for θ. In the former case we can proceed by induction on k.

(v) $\bigwedge_m \phi_m \geq_\alpha \theta$ can hold only by (2) or (6). Again only (2) needs consideration. Here there exist $\theta_1,\dots,\theta_k$ and $\beta < \alpha$ with

$$\bigwedge_m \phi_m = \theta_1 \geq_\beta \dots \geq_\beta \theta_k = \theta.$$

As $\bigwedge_m \phi_m \neq \theta$, $k > 1$ and hence by I.H. there exist $\tilde{\phi}_m$ $(m \in M)$ with $\phi_m \to_\beta \tilde{\phi}_m$ and $\theta_2 = \bigwedge_m \tilde{\phi}_m$. This is transmitted to θ_k by the induction on k.

COROLLARY (A.P. Theorem for $\geq$). The same result holds if we replace both $\geq_\alpha$ and $\to_\alpha$ throughout by $\geq$.

Proof. $\geq$ equals both $\geq_\alpha$ and $\to_\alpha$ for sufficiently large α.

3.7 THE CHURCH-ROSSER THEOREM FOR $\mathcal{L}_M(P)$.

THEOREM CR($\geq_\alpha$) holds for each α.

Proof By induction on α. If CR($\geq_\beta$) holds for each $\beta < \alpha$ then it holds for $\bigcup_{\beta<\alpha}(\geq_\beta)$ and so also for $(\to_\alpha) = \mathrm{TC}(\bigcup_{\beta<\alpha}(\geq_\beta))$. We now prove it for α. All diagrams shown are in the $\to_\alpha$ relation only; thus transitivity holds in them.

(1) Given $\phi \geq_\alpha (\theta',\theta'')$ with $\phi = \mathrm{T}$ or $\phi = \mathrm{F}$ we have $\theta' = \theta'' = \phi$.

(2) If $p \geq_\alpha (\theta',\theta'')$ with $p \in P$, then by the A.P. Theorem for $\geq_\alpha$ $\theta' = p$ or $p^0 \to_\alpha \theta'$ and similarly for θ''. If $\theta' = p$ then $(\theta',\theta'') \geq_\alpha \theta''$. Similarly if $\theta'' = p$ then $(\theta',\theta'') \geq_\alpha \theta'$. Finally if $p^0 \to_\alpha (\theta',\theta'')$ then we can apply CR($\to_\alpha$) to find ψ with $(\theta',\theta'') \to_\alpha \psi$ and hence $(\theta',\theta'') \geq_\alpha \psi$.

(3) If $\neg\phi \geq_\alpha (\theta',\theta'')$ by the A.P. Theorem we have existence of ϕ', ϕ'' with $\phi \to_\alpha (\phi',\phi'')$ and $\theta' = \neg\phi'$, $\theta'' = \neg\phi''$.
Hence by CR($\to_\alpha$) there exists ψ with $\phi' \to_\alpha \psi$ and $\phi'' \to_\alpha \psi$, so that $\neg\phi' \geq_\alpha \neg\psi$ and $\neg\phi'' \geq_\alpha \neg\psi$.

(4) If $(\phi_0 \mathrm{E} \phi_1) \geq_\alpha (\theta',\theta'')$, by the A.P. Theorem there exist ϕ_i', ϕ_i'' $(i = 0,1)$ with $\phi_i \to_\alpha (\phi_i',\phi_i'')$
and $\theta' = \phi_0' \mathrm{E} \phi_1'$ or $\phi_0' = \phi_1'$ and $\theta' = \mathrm{T}$
and $\theta'' = \phi_0'' \mathrm{E} \phi_1''$ or $\phi_0'' = \phi_1''$ and $\theta'' = \mathrm{T}$.

By $CR(\to_\alpha)$ we can complete the diagrams

$$\begin{tikzcd}[column sep=small]
 & \phi_0 \arrow[dl, dash] \arrow[dr, dash] & \\
\phi_0' \arrow[dr, dashed, dash] & & \phi_0'' \arrow[dl, dashed, dash] \\
 & \psi_0 &
\end{tikzcd}
\qquad
\begin{tikzcd}[column sep=small]
 & \phi_1 \arrow[dl, dash] \arrow[dr, dash] & \\
\phi_1' \arrow[dr, dashed, dash] & & \phi_1'' \arrow[dl, dashed, dash] \\
 & \psi_1 &
\end{tikzcd}$$

In case both $\theta' = \phi_0' \mathrm{E} \phi_1'$ and $\theta'' = \phi_0'' \mathrm{E} \phi_1''$ we obtain $(\theta',\theta'') \geq_\alpha (\psi_0 \mathrm{E} \psi_1)$. The case $\theta' = \theta'' = T$ is trivial. Next, in case $\phi_0' = \phi_1'$ and $\theta' = T$ and, on the other hand, $\theta'' = \phi_0'' \mathrm{E} \phi_1''$ we rearrange the picture above and apply $CR(\to_\alpha)$ once more to obtain

$$\begin{tikzcd}[column sep=small]
 & \phi_0 \arrow[dl, dash] \arrow[dr, dash] & & \phi_1 \arrow[dl, dash] \arrow[dr, dash] & \\
\phi_0'' \arrow[dr, dash] & & \phi_0' \arrow[dl, dash] \arrow[dr, dash] & & \phi_1'' \arrow[dl, dash] \\
 & \psi_0 \arrow[dr, dash] & & \psi_1 \arrow[dl, dash] & \\
 & & \chi & &
\end{tikzcd}$$

Thus $\phi_0'' \to_\alpha \chi$, $\phi_1'' \to_\alpha \chi$ and so $(\phi_0'' \mathrm{E} \phi_1'') \geq_\alpha T$ which shows $(\theta',\theta'') \geq_\alpha T$. The symmetric case in θ',θ'' is treated in the same way.

(5) If $(\bigwedge_m \phi_m) \geq_\alpha (\theta',\theta'')$ by the A.P. Theorem there exist ϕ_m', ϕ_m'' with $\phi_m \to_\alpha (\phi_m',\phi_m'')$ for each $m \in M$ and $\theta' = \bigwedge_m \phi_m'$, $\theta'' = \bigwedge_m \phi_m''$. Then by $CR(\to_\alpha)$ there exist ψ_m with $(\phi_m', \phi_m'') \to_\alpha \psi_m$ for $m \in M$, hence $(\bigwedge_m \phi_m', \bigwedge_m \phi_m'') \geq_\alpha \bigwedge_m \psi_m$.

COROLLARY CR holds for the $\geq$ relation of $\mathcal{L}_M(P)$.

3.8 THE EQUIVALENCE RELATION We take

$$\phi \equiv \psi \quad \underset{\text{def}}{\Leftrightarrow} \quad \exists \chi [(\phi, \psi) \geq \chi].$$

LEMMA (i) $\equiv$ is an equivalence relation

(ii) $(\phi \equiv T) \Leftrightarrow \phi \geq T$, and $(\phi \equiv F) \Leftrightarrow \phi \geq F$

(iii) $(\phi \equiv \psi) \Leftrightarrow (\phi \mathrm{E} \psi) \geq T \Leftrightarrow (\phi \mathrm{E} \psi) \equiv T$

Proof For (i) the only point to check is transitivity. If $\phi \equiv \psi$, $\psi \equiv \theta$ then for some χ, χ' we have $(\phi, \psi) \geq \chi$ and $(\psi, \theta) \geq \chi'$. By CR there exists χ^* with $(\chi, \chi') \geq \chi^*$ so $\phi \equiv \theta$. (ii) is then immediate from the A.P. Theorem $(\geq)$ for

T, F. For (iii) if $\phi \equiv \psi$ then for some χ we have $\phi \geq \chi$, $\psi \geq \chi$ so $(\phi \mathrm{E} \psi) \geq (\chi \mathrm{E} \chi) \geq \mathrm{T}$. Conversely by the A.P. Theorem ($\geq$) for E, if $(\phi \mathrm{E} \psi) \geq \mathrm{T}$ then there exist ϕ', ψ' with $\phi \geq \phi'$ and $\psi \geq \psi'$ and $(\phi' \mathrm{E} \psi') = \mathrm{T}$ or $\phi' = \psi'$; but $(\phi' \mathrm{E} \psi') = \mathrm{T}$ also implies $\phi' = \psi'$ so $\phi \equiv \psi$ in either case.

The relation $\equiv$ is non-trivial by $\mathrm{T} \not\equiv \mathrm{F}$ since $\mathrm{T} \geq \mathrm{F}$ does not hold. Note also that the property of being determinate, $\phi \equiv \mathrm{T}$ or $\phi \equiv \mathrm{F}$, is equivalent to $\phi \geq \mathrm{T}$ or $\phi \geq \mathrm{F}$.

3.9 SEMANTICS FOR $\mathcal{L}_M(P)$. By a structure for $\mathcal{L}_M(P)$ we mean a pair $\mathfrak{a} = (P_{\mathfrak{a}}, \equiv_{\mathfrak{a}})$ where $P_{\mathfrak{a}} \subseteq P$ and $\equiv_{\mathfrak{a}}$ is a relation on $\mathcal{L}_M(P)$ such that $\mathrm{T} \equiv_{\mathfrak{a}} \mathrm{T}$, $\mathrm{F} \equiv_{\mathfrak{a}} \mathrm{F}$ and $\mathrm{T} \not\equiv_{\mathfrak{a}} \mathrm{F}$. For any structure $\mathfrak{a}$, the satisfaction relation $\mathfrak{a} \models (\cdot)$ is defined by:

$$\mathfrak{a} \models \mathrm{T}, \quad \mathfrak{a} \not\models \mathrm{F}$$

$$\mathfrak{a} \models p \Leftrightarrow p \in P_{\mathfrak{a}}$$

$$\mathfrak{a} \models (\phi \mathrm{E} \psi) \Leftrightarrow (\phi \equiv_{\mathfrak{a}} \psi)$$

$$\mathfrak{a} \models (\neg\phi) \Leftrightarrow \mathfrak{a} \not\models \phi$$

$$\mathfrak{a} \models \bigwedge_m \phi_m \Leftrightarrow \forall m (\mathfrak{a} \models \phi_m).$$

By hypothesis, this is in accord with the operations extended to truth-functional sentences.

3.10 A CONSISTENCY RESULT The laws of equivalence of 2.1 are adapted to $\mathcal{L}_M(P)$ as follows. We write $(\phi \mathrm{E} \psi)$ instead of $(\phi \equiv \psi)$. Put $\phi \wedge \psi = \bigwedge_m \phi_m$ where $\phi_0 = \phi$, $\phi_m = \psi$ for $m \neq o$ (o some fixed element of M), and then $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$ and $(\phi \to \psi) = (\neg\phi \vee \psi)$. Finally put $\mathrm{T}(\phi) = (\phi \mathrm{E} \mathrm{T})$, $\mathrm{F}(\phi) = (\phi \mathrm{E} \mathrm{F})$ and $\mathrm{D}(\phi) = (\mathrm{T}(\phi) \vee \mathrm{F}(\phi))$, Now the schemata of 2.1 become schemata of $\mathcal{L}_M(P)$ except for the following changes:

In (2) the schemata for $\forall$ preserving $\equiv$ must be replaced by one for $\bigwedge_M$ preserving E. i.e.

$$\bigwedge_m (\phi_m \mathrm{E} \psi_m) \to (\bigwedge_m \phi_m) \mathrm{E} (\bigwedge_m \psi_m).$$

(3) must be replaced by

$$p \mathrm{E} \mathrm{T} \leftrightarrow p \qquad (p \in P \cup \{\mathrm{F}\}).$$

(4) (v) and (vi) must be replaced by

$$T(\bigwedge_m \phi_m) \leftrightarrow \bigwedge_m T(\phi_m)$$

$$F(\bigwedge_m \phi_m) \leftrightarrow \bigwedge_m D(\phi_m) \wedge \neg \bigwedge_m T(\phi_m)$$

Finally (4) (iii) and (iv) are here unnecessary.

Given any $p \mapsto p^0$, define $\geq$ and $\equiv$ for $\mathcal{L}_M(P)$ as in 3.3 and 3.8. Then put $P_{\mathfrak{A}} = \{p \in P | p \equiv T\}$, and $\mathfrak{A} = (P_{\mathfrak{A}}, \equiv)$. Even if we posit prima-facie "contradictory" basic equivalences, such as $p\,E\,(\neg p)$ for $p \in P$, the following result guarantees their consistency.

THEOREM $\mathfrak{A}$ is a model of the above specified schemata and all sentences $(p\,E\,p^0)$ for $p \in P$.

Proof This is mostly quite routine by definition of $\mathfrak{A}$ and the properties we now have for $\geq$ and $\equiv$. We consider only (4)(i) and 4(viii). For 4(i), if $\mathfrak{A} \models T(\neg\phi)$ then $\neg\phi \equiv T$ and hence $\neg\phi \geq T$. By the A.P. Theorem ($\geq$) for $\neg$ we conclude that $\phi \geq \phi'$ where $(\neg\phi') = T$ and hence $\phi' = F$ so that $\phi \geq F$ and hence $\mathfrak{A} \models F(\phi)$. Conversely if $\mathfrak{A} \models F(\phi)$ then $\phi \equiv F$ and hence $(\neg\phi) \equiv T$ so that $\mathfrak{A} \models T(\neg\phi)$. For 4(viii) if $\mathfrak{A} \models F(\phi\,E\,\psi)$ then $(\phi\,E\,\psi) \geq F$ and hence by the A.P. Theorem ($\geq$) for E there are ϕ', ψ' such that $\phi \geq \phi'$, $\psi \geq \psi'$ and $(\phi'\,E\,\psi') = F$, so that $\phi' = T$ and $\psi' = F$ or else $\phi' = F$ and $\psi' = T$. Hence either $\phi \equiv T$ and $\psi \equiv F$ or else $\psi \equiv T$ and $\phi \equiv F$. In either case we get $\mathfrak{A} \models D(\phi) \wedge D(\psi) \wedge (\phi \not{E} \psi)$. Conversely if $\mathfrak{A} \models D(\phi) \wedge D(\psi) \wedge (\phi \not{E} \psi)$ we have $\phi \equiv T$ and $\psi \equiv F$ or else $\phi \equiv F$ and $\psi \equiv T$, and in either case $(\phi\,E\,\psi) \equiv F$ so that $\mathfrak{A} \models F(\phi\,E\,\psi)$.

§4. PROOF OF THE MAIN RESULT

4.1 THE LANGUAGE $\mathcal{L}^*(M)$ Throughout §4, $\mathfrak{M} = (M, \ldots)$ is any countably infinite structure for the language $\mathcal{L}$. The interpretation of a relation symbol R in $\mathfrak{M}$ is denoted by $R_{\mathfrak{M}}$, and that of a function symbol f by $f_{\mathfrak{M}}$. It is assumed that $=_{\mathfrak{M}}$ is the equality relation on M. By $\mathcal{L}^*(M)$ we mean the language $\mathcal{L}^*$ of 2.2 augmented by a constant symbol $\bar{m}$ for each

$m \in M$.

4.2 REPRESENTATION OF TERMS IN M. We wish to assign to each closed term t of $\mathcal{L}^*(M)$ an element t^* of M in such a way that

(1) $$t_1^* = t_2^* \Rightarrow s(t_1)^* = s(t_2)^*.$$

For this purpose, call a term or formula of $\mathcal{L}^*(M)$ *simple* if it has no closed subterms except those of the form $\bar{m}$ for $m \in M$. By a *simple abstract* we mean a closed term of the form $\{x|\phi\}$ where ϕ is simple. Fix a bijection $t \mapsto t_M$ of all simple abstracts t with M; this is possible by the infinitude of M. Now t^* is defined inductively as follows for all closed t:

(2) (i) $(\bar{m})^* = m$

(ii) $f(t_1,\ldots,t_n)^* = f\,(t_1^*,\ldots,t_n^*)$

(iii) $\{x|\phi\}^* = \{x|\phi^\wedge\}_M$

where $\phi^\wedge$ is obtained from ϕ by replacing each maximal closed subterm t of ϕ by t^+, with

(3) $$t^+ =_{\mathrm{def}} \overline{t^*}.$$

That (1) is satisfied is now guaranteed by the following:

LEMMA If $s(y)$ and t are terms such that t and $s(t)$ are closed then $s(t)^* = s(t^+)^*$.

Proof By induction on the term $s(y)$ that contains at most y free. If $s(y)$ is $\bar{m}$ the result is clear, and if $s(y)$ is y it follows from $t^* = (t^+)^*$. If $s(y)$ is $f(s_1(y),\ldots,s_n(y))$ then $s(t)^* = f(s_1(t),\ldots,s_n(t))^*$ and $s(t^+)^* = f(s_1(t^+),\ldots,s_n(t^+))^*$ so that

$$\begin{aligned} s(t)^* &= f_{\mathcal{M}}(s_1(t)^*,\ldots,s_n(t)^*) \\ &= f_{\mathcal{M}}(s_1(t^+)^*,\ldots,s_n(t^+)^*) \\ &= s(t^+)^*. \end{aligned}$$

Finally, if $s(y)$ is $\{x|\phi(x,y)\}$ where ϕ has at most x, y free then $\phi(x,y)$ can be written as $\psi(x,\ s_1(y),\ldots,s_k(y))$ where $s_1(y),\ldots,s_k(y)$ is a pairwise distinct list of the terms occurring in $\phi(x,y)$ that contain at most y free and are maximal with respect to the property. Then by the induction hypothesis, $s_i(t)^+ = s_i(t^+)^+$ for $i = 1,\ldots,k$ and hence

$$\begin{aligned}\phi(x,t)\hat{} &= \psi(x, s_1(t),\ldots,s_k(t))\hat{} = \psi(x, s_1(t)^+,\ldots,s_k(t)^+) \\ &= \psi(x, s_1(t^+)^+,\ldots,s_k(t^+)^+) = \psi(x, s_1(t^+),\ldots,s_k(t^+))\hat{} \\ &= \phi(x, t^+)\hat{}.\end{aligned}$$

It follows that

$$s(t)^* = \{x|\phi(x,t)\}^* = \{x|\phi(x,t)\hat{}\}_M = \{x|\phi(x,t^+)\hat{}\}_M = s(t^+)^*.$$

4.3 SEMANTICS OF $\mathcal{L}^*(M)$ To expand $\mathfrak{M}$ to a structure $\mathfrak{M}^*$ for $\mathcal{L}^*(M)$ we need to specify

(1) a binary relation $\varepsilon_{\mathfrak{M}}$ on M and

(2) a binary relation $\equiv_{\mathfrak{M}}$ between the sentences of $\mathcal{L}^*(M)$, for which

(3) $\phi(t) \equiv_{\mathfrak{M}} \psi(t) \Leftrightarrow \phi(t^+) \equiv_{\mathfrak{M}} \psi(t^+)$.

This last condition is needed to insure that the laws of = and substitution are valid in $\mathfrak{M}^*$, where the semantics is determined as follows for all closed ϕ of $\mathcal{L}^*(M)$:

(4) $\mathfrak{M}^* \models R(t_1,\ldots,t_n) \Leftrightarrow R_{\mathfrak{M}}(t_1^*,\ldots,t_n^*)$ for each relation symbol R of $\mathcal{L}^*$

$$\mathfrak{M}^* \models \neg\phi \Leftrightarrow \mathfrak{M}^* \not\models \phi$$

$$\mathfrak{M}^* \models (\phi \rightarrow \psi) \Leftrightarrow [\mathfrak{M}^* \not\models \phi \text{ or } \mathfrak{M}^* \models \psi]$$

$$\mathfrak{M}^* \models (\phi \equiv \psi) \Leftrightarrow \phi \equiv_{\mathfrak{M}} \psi$$

$$\mathfrak{M}^* \models \forall x\phi(x) \Leftrightarrow \mathfrak{M}^* \models \phi(\bar{m}) \text{ for all } m \in M.$$

LEMMA For each $\phi(x)$, t with t, $\phi(t)$ closed we have:

(i) $\mathfrak{M}^* \models \phi(t) \Leftrightarrow \mathfrak{M}^* \models \phi(t^+)$

(ii) $\mathfrak{M}^* \models \forall x\phi(x) \rightarrow \phi(t)$

(iii) $\mathfrak{M}^* \models s = t \wedge \phi(s) \rightarrow \phi(t)$.

Proof (i) follows by induction from the lemma of 4.2 and the hypothesis (3) above. Given t let $t^* = m$ so $t^+ = \bar{m}$ and by (i) $\mathfrak{M}^* \models \phi(t) \leftrightarrow \phi(\bar{m})$. From this we obtain (ii) and (iii).

COROLLARY If $\phi(x_1,\ldots,x_n)$ is any instance of a law of logic in $\mathcal{L}^*$ then $\phi(\bar{m}_1,\ldots,\bar{m}_n)$ is true in $\mathcal{M}^*$.

In this sense the laws of logic of $\mathcal{L}^*(M)$ are valid in $\mathcal{M}^*$. Obviously, the formulas valid in $\mathcal{M}^*$ are closed under modus ponens and universal generalization.

4.4 EMBEDDING OF $\mathcal{L}^*(M)$ IN AN INFINITARY LANGUAGE Now assume $\mathcal{M}$ is a model for S. Let P be the set of sentences $(\bar{m} \in \bar{n})$ for m, n $\in$ M. We shall expand $\mathcal{M}$ to a model $\mathcal{M}^*$ of S* by intermediate use of a translation $\phi \mapsto \phi^+$ sending each sentence ϕ of $\mathcal{L}^*(M)$ into a sentence ϕ^+ of $\mathcal{L}_M(P)$, given as follows, where R is a relation symbol of $\mathcal{L}$ or is equality:

$$
\begin{aligned}
&(1)\qquad R(t_1,\ldots,t_n)^+ = \begin{cases} T & \text{if } R_{\mathcal{M}}(t_1^*,\ldots,t_n^*) \\ F & \text{otherwise} \end{cases}\\
&\qquad (s \in t)^+ = (s^+ \in t^+)\\
&\qquad (\neg\phi)^+ = \neg\phi^+\\
&\qquad (\phi \to \psi)^+ = (\phi^+ \to \psi^+)\\
&\qquad (\phi \equiv \psi)^+ = (\phi^+ \mathrm{E}\, \psi^+)\\
&\qquad (\forall x\phi(x))^+ = \bigwedge_m \phi(\bar{m})^+
\end{aligned}
$$

LEMMA 1. For all $\phi(x)$, t in $\mathcal{L}^*(M)$ with both t and $\phi(t)$ closed we have $\phi(t)^+ = \phi(t^+)^+$.

Proof The statement follows from the Lemma of 4.2 in the atomic case. It is then established for all ϕ by induction.

If $\mathfrak{a} = (P_{\mathfrak{a}}, \equiv_{\mathfrak{a}})$ is any structure for $\mathcal{L}_M(P)$, we obtain a structure $\mathcal{M}^*$ for $\mathcal{L}^*(M)$ expanding $\mathcal{M}$ by taking

$$
\begin{aligned}
&(2)\qquad m_1 \in_{\mathcal{M}} m_2 \Leftrightarrow (\bar{m}_1 \in \bar{m}_2) \in P_{\mathfrak{a}}\\
&\qquad \phi \equiv_{\mathcal{M}} \psi \Leftrightarrow (\phi^+ \equiv_{\mathfrak{a}} \psi^+).
\end{aligned}
$$

By the preceding lemma, this immediately satisfies the required substitution property 4.3(3).

LEMMA 2. Suppose $\mathfrak{a}$ is any structure for $\mathcal{L}_M(P)$. Then for any closed ϕ of $\mathcal{L}^*(M)$ we have

$$\mathfrak{M}^* \models \phi \Leftrightarrow \mathfrak{A} \models \phi^+.$$

4.5 A MODEL FOR S* We continue to assume that $\mathfrak{M} \models S$. Now consider the map $p \mapsto p^0$ of members of P defined by:

$$(\bar{m} \varepsilon \bar{n})^0 = \phi(\bar{m})^+ \text{ if } n = \{x|\phi(x)\}_M \text{ with } \phi \text{ simple.}$$

Since, by assumption, every $n \in M$ is the value of a unique $\{x|\phi(x)\}_M$ with ϕ simple, $(\ )^0$ is well-defined.

Choose $\mathfrak{A}$ *to be a model of the laws of equivalence in* $\mathcal{L}_M(P)$ *and all sentences* $(p\,E\,p^0)$ *for* $p \in P$, as guaranteed by the theorem of 3.10.

Finally, define $\mathfrak{M}^*$ from $\mathfrak{A}$ as described in the preceding section 4.4.

THEOREM $\mathfrak{M}^*$ is a model of S*.

Proof By lemma 2 of 4.4 we need only show that $\mathfrak{A} \models \phi^+$ for each substitution instance in M of an instance of one of the laws (1)-(4) of equivalence of 2.1 and the Abstraction Principle. Almost all the work is already done. First, except for 3(ii) all the laws (1)-(4) of 2.1 translate into the corresponding laws for $\mathcal{L}_M(P)$ in 3.10. (3)(ii) itself translates into one of

$$TEF \leftrightarrow \neg T \quad \text{or}$$

$$FEF \leftrightarrow \neg F.$$

Both of these clearly always hold.

As for the Abstraction Principle, consider any instance in $\mathcal{L}^*(M)$:

$$\bar{m} \varepsilon \{x|\phi(x)\} \equiv \phi(\bar{m})$$

where $\phi(\bar{m})$ is closed. The translation of this is

$$(\bar{m} \varepsilon \bar{n})\ E\ \phi(\bar{m})^+$$

where $n = \{x|\phi(x)\}^* = \{x|\phi^{\wedge}(x)\}_M$. So we have to show that

$$\mathfrak{A} \models (\bar{m} \varepsilon \bar{n})\ E\ \phi(\bar{m})^+.$$

Now by the definition of $(\bar{m} \varepsilon \bar{n})^0$ and the fact that

$\mathfrak{a} \models (p\,E\,p^0)$ for $p \in P$ we know

$$\mathfrak{a} \models (\bar{m} \in \bar{n})\ E\ \phi\hat{}(\bar{m})^+.$$

It thus suffices to show that $\phi(\bar{m})^+ = \phi\hat{}(\bar{m})^+$. So let $t_1,\ldots,t_k$ list the distinct (disjoint) occurrences of closed terms in $\phi(x)$ so that $\phi\hat{}(x)$ is $\psi(x, t_1^+,\ldots,t_k^+)$ where $\phi(x)$ is $\psi(x, t_1,\ldots,t_k)$ for a suitable $\psi(x, y_1,\ldots,y_k)$. Then by repeated application of Lemma 1 of 4.4

$$\begin{aligned}\phi(\bar{m})^+ &= \psi(\bar{m},t_1,\ldots,t_k)^+ = \psi(\bar{m},t_1^+,t_2,\ldots,t_k)^+ = \ldots\\ &= \psi(\bar{m},t_1^+,\ldots,t_k^+)^+ = \phi\hat{}(\bar{m})^+.\end{aligned}$$

The Main Result of 2.3 is now a corollary. For if $S^* \vdash \phi$ where ϕ is a closed sentence of $\mathcal{L}$ and $S \nvdash \phi$ then S has a countable model $\mathfrak{M}$ in which ϕ fails; $\mathfrak{M}$ must be infinite by hypothesis. By the preceding theorem $\mathfrak{M}$ can be expanded to a model $\mathfrak{M}^*$ of S^*, so also $\mathfrak{M}^* \models \neg\phi$. Since the laws of logic and logical inference in $\mathcal{L}^*$ are valid in $\mathfrak{M}^*$ we must have $\mathfrak{M}^* \models \phi$, a contradiction.

§5. APPENDIX (by S. Feferman)

5.1 DISCUSSION OF AIMS RELATIVE TO S* The theory S* admits classes which belong to themselves, the simplest being $V = \{x \mid x = x\}$. Such availability of self-applicative notions would appear to be of mathematical utility, e.g., in a foundation for the informal theory of categories. In previous publications (Feferman (1975), Feferman (1977)), I have set up a variety of systems for such purposes and discussed their aims at length. Due to limitations of space, comparison of what may be achieved via S* with what had been achieved by the previous work is deferred to another occasion.

However, I think it is useful for orientation to repeat the basic point of view at which I have arrived in this direction. (i) There is nothing currently on the horizon which gives hope of obtaining anything like the Frege-Russell (pre-PM) or

Church-Curry programs for a foundation of mathematics in a type-free theory. These programs are first of all monistic ("everything is a class" or "everything is a function") and secondly attempt to extract all of mathematics from some few relatively simple principles of a very general "logical" character.
(ii) Instead these programs are more profitably reversed. We should start with a part S of mathematics that is already accepted, or at least reasonably well-understood - be it number theory, or analysis, or even stronger theories. S should then be extended to a theory $\hat{S}$ which admits instances of self-application not available in S. This should be done as a *matter of convenience*; thus the extension should be *conservative*.
(iii) There are a number of cases in the history of mathematics which provide paradigms of such extensions, among them:

(a) expansion of the reals to the complex numbers;

(b) expansion of the Euclidean plane to the projective plane;

(c) expansion of ordinary analysis to infinitesimal analysis.

Each such expansion gives some advantage of freedom to deal with the new "ideal" objects, but also in each case a price has to be paid - some principles or structure has to be given up (in the above: order, metric, and induction on "external" properties, resp.). In some cases the price seems to outweigh the advantages. The choice of expansion for given purposes need not be uniquely determined, and criteria of simplicity come into play.
(iv) Among the principles which people have considered giving up for a type-free theory are those of ordinary logic. For example, there is a frequent move made to some sort of three-valued logic. A little experimentation shows this to be extremely debilitating for mathematical practice, and I believe this kind of move to be a great mistake. Instead I have looked for systems which involve an *expansion of ordinary logic* (using "partial classes" in Feferman (1975), or by new intensional operators $\Box$ in Feferman (1977) and $\equiv$ here).

(v) On the other hand, great efforts have been expended to preserve the extensionality axiom for classes: $\forall x(x \in a \leftrightarrow x \in b) \rightarrow a = b$. However, there is really no need for such on mathematical grounds; one simply deals systematically with classes equipped with an "equality relation", as had always been done before set-theory became fashionable. (Indeed, in constructive mathematics, extensionality is simply excluded by the basic point of view, and without mathematical loss.) Extensionality is the main familiar principle which I am prepared to sacrifice, and with no qualms.

Remark. In view of this, the reader may prefer to think of S* as a theory of *properties* and to read $y \in x$ as: the property x applies to y.

(vi) A more technical point that I have elaborated in the previous publications is the need to treat *operations* independently of and prior to *classes* for a sufficiently rich mathematical theory. Thus the theory S* presented here only gives a new method to carry out the second stage.

While it is quite possible that the extensionality axiom is consistent with S* (and this is an open question), it becomes inconsistent if S is a theory of operations, and abstraction $\{x|\phi(x, y_1,\ldots,y_n)\}$ is uniformly given as a function of its parameters (cf. Feferman (1975)).

5.2 FURTHER DISCUSSION OF THE RUSSELL CLASS EXAMPLE In 2.4 it was shown that $S^* \vdash \neg\phi$ where $\phi = (r \in r)$ is such that $S^* \vdash (\phi \equiv \neg\phi)$. Myhill found this feature of S* objectionable. My own view was that this was peculiar, but a reasonable peculiarity. Kreisel has expressed the view that there is nothing peculiar about the matter at all and that it corresponds to the familiar convention in ordinary cumulative type theory which declares an atomic sentence *false* where previously (in Russell's type theory) it had been treated as *senseless*. There is, however, one novel point. The idea of $\equiv$ in this context is that $(\phi \equiv \psi)$ is to hold if the equivalence of ϕ and ψ follows from (AP) considered as a basic equivalence by definition. It would seem, on this informal reading, that if we establish (AP) $\vdash (\phi \leftrightarrow F)$ we ought to accept $(\phi \equiv F)$. There is no consistent way to do this with the axioms given here.

5.3 RELATED SYSTEMS IN THE λ-CALCULUS In a first version of this paper (1977) I formulated a system analogous to S* for the λ-calculus with an "equality" operator and conjectured its consistency. Roger Hindley pointed out the connection with the CR problem for the λ-δ calculus. Indeed it was shown soon after in Klop 1977 that there is a counterexample to CR for that calculus, from which it followed that my conjecture was wrong. It seems the best one can do is something as follows. Consider a system of terms containing constants T, F, E, such that whenever s,t are terms then so also are st and $\lambda x.t(x)$. A reduction relation $\geqslant$ is defined from which $\equiv$ is defined as: having a common reduct. The predicate D(t) is taken to hold just in case t is reducible to normal form. Consider the following laws:

(1) $\equiv$ is an equivalence relation

(2) $\equiv$ preserves application and abstraction

(3) D(T) and D(F)

(4) $D(\lambda x\, t) \Leftrightarrow D(t)$

(5) $D((\lambda x\, t(x))\, s) \leftrightarrow D(t(s))$

(6) $D(E\, st) \leftrightarrow D(s) \wedge D(t)$

(7) $D(s) \wedge D(t) \wedge s \equiv t \rightarrow (E\, st) \equiv T$

(8) $D(s) \wedge D(t) \wedge s \not\equiv t \rightarrow (E\, st) \equiv F$

(9) $T \not\equiv F$

(10) $(\lambda x\, t(x))s \equiv t(s)$.

Hindley has drawn my attention to Church's theorem giving CR for his λδ-calculus (Church 1941 pp.62-68), from which the consistency of (1)-(10) follows using an interpretation as suggested above.

The main weakening of the present system from the earlier one I had formulated is that we do *not* have the law $s \equiv t \rightarrow (Est) \equiv T$ for arbitrary s,t - for then Klop's counterexample would apply. The system can be augmented by operators N and A, corresponding to $\neg$ and $\forall$, so as to embed logic. Even with a consistency result for such an extension, it is not obvious how to extract a consistency result for S* where the law $(\phi \mathrm{E} \phi) \equiv \mathrm{T}$ holds for arbitrary ϕ.

§6. APPENDIX (by P. Aczel)

6.1 INTRODUCTION At a late stage in preparing this paper I noticed that $\mathcal{L}$* can be interpreted more elegantly in terms of a many-valued semantics. In this approach the two-valued semantics is combined with an auxiliary semantics that assigns to each sentence a truth value taken from a set Ω of possible truth values. Then $\phi \equiv \psi$ is taken to hold (in the standard two-valued semantics) just in case ϕ and ψ are assigned the same truth value in Ω in the auxiliary semantics.

In this appendix the main result is reviewed in the light of the above indicated ideas, and a prima-facie slight variant of the main theorem of 2.3 is given a simple proof using a three valued semantics. However the theorem as initially stated still seems to require use of a Church-Rosser argument.

6.2 M-VALUATION SYSTEMS Ω The auxiliary semantics is based on the following notions, relative to an infinite set M.

An *M-valuation system* $\Omega = (\Omega, \mathrm{T}, -, \supset, \bigwedge_M, \mathrm{E})$ consists of a set Ω, $\mathrm{T} \in \Omega$ and functions $-: \Omega \to \Omega$, $\supset: \Omega \times \Omega \to \Omega$, $\bigwedge_M: \Omega^M \to \Omega$ and $\mathrm{E}: \Omega \times \Omega \to \Omega$ such that $\mathrm{T} \neq \mathrm{F}$ where $\mathrm{F} = -\mathrm{T}$.

An M-valuation system Ω will be said to be *regular* if the following conditions hold.

(i) $-d = \mathrm{T} \quad \Leftrightarrow \quad d = \mathrm{F}$

(ii) $-d = \mathrm{F} \quad \Leftrightarrow \quad d = \mathrm{T}$

(iii) $(d_1 \supset d_2) = \mathrm{T} \quad \Leftrightarrow \quad d_1, d_2 \in \{\mathrm{T},\mathrm{F}\}$ and $d_1 = \mathrm{F}$ or $d_2 = \mathrm{T}$

(iv) $(d_1 \supset d_2) = \mathrm{F} \quad \Leftrightarrow \quad d_1 = \mathrm{T}$ and $d_2 = \mathrm{F}$

(v) $\bigwedge_{m \in M} d_m \mathrm{T} = \mathrm{T} \quad \Leftrightarrow \quad \{d_m | m \in M\} = \{\mathrm{T}\}$

(vi) $\bigwedge_{m \in M} d_m = \mathrm{F} \quad \Leftrightarrow \quad \{\mathrm{F}\} \subseteq \{d_m | m \in M\} \subseteq \{\mathrm{T},\mathrm{F}\}$

(vii) $d_1 = d_2 \in \{T,F\} \Rightarrow E(d_1, d_2) = T \Rightarrow d_1 = d_2$

(viii) $E(d_1, d_2) = F \Leftrightarrow \{d_1, d_2\} = \{T, F\}$

Ω is *maximal regular* if (vii) is strengthened to

$(\text{vii})_{\max}$ $E(d_1, d_2) = T \Leftrightarrow d_1 = d_2$,

and Ω is *minimal regular* if (vii) is strengthened to

$(\text{vii})_{\min}$ $E(d_1, d_2) = T \Leftrightarrow d_1 = d_2 \in \{T, F\}$

EXAMPLES

(1) The standard M-valuation system with $\Omega = \{T,F\}$ is the unique one that is both maximal and minimal regular.

(2) If we consider M-valuation systems with $\Omega = \{T, F, U\}$ where U is a third truth value then it is easy to see that exactly two of them are regular, one being maximal, the other minimal.

(3) The construction of §3 may be seen to generate maximal regular M-valuation systems in the following way. Given any assignment $p \mapsto \chi_p (p \in P)$ let $\equiv$ be the equivalence relation on sentences of $\mathcal{L}_M(P)$ that is defined in 3.8.

Now let Ω be the set of equivalence classes $\phi/\equiv$ of sentences ϕ of $\mathcal{L}_M(P)$ under this equivalence relation. The M-valuation system Ω may now be obtained by the following schemes

$$T^{\Omega} = T/\equiv$$

$$-(\phi/\equiv) = (\neg\phi)/\equiv$$

$$(\phi/\equiv) \supset (\psi/\equiv) = (\phi \rightarrow \psi)/\equiv$$

$$\bigwedge_{m\in M} (\phi_M/\equiv) = (\bigwedge_{m\in M} \phi_m)/\equiv$$

$$E(\phi/\equiv, \psi/\equiv) = E(\phi, \psi)/\equiv.$$

These schemes make sense as defining Ω and it is easily seen that Ω is maximal regular.

6.3 A REFORMULATION OF THE MAIN RESULT OF §3 Given a regular M-valuation system Ω and an assignment $\alpha : P \rightarrow \Omega$, we may interpret the language $\mathcal{L}_M(P)$ by assigning a value $\|\phi\|^{\Omega}_{\alpha} \in \Omega$ to each sentence ϕ:

$$\|T\| = T,$$
$$\|F\| = F,$$
$$\|p\| = \alpha(p), \quad (p \in P)$$
$$\|\neg\phi\| = -\|\phi\|,$$
$$\|\bigwedge_{m\in M} \phi_m\| = \bigwedge_{m\in M} \|\phi_m\|,$$
$$\|E(\phi, \psi)\| = E(\|\phi\|, \|\psi\|).$$

Note that the regularity of Ω ensures that there is no ambiguity in the above schemes.

The main result of §3 can now be rephrased in the following way.

THEOREM. For any assignment $p \mapsto \chi_p (p \in P)$ there is a maximal regular M-valuation system Ω and an assignment $\alpha : P \to \Omega$ such that for all $p \in P$

(*) $\quad \alpha(p) = \|\chi_p\|^{\Omega}_{\alpha}$.

Proof Ω is taken as specified in (3) of 6.2. Then $\alpha : P \to \Omega$ is defined by

$$\alpha(p) = p/\equiv \quad (p \in P).$$

An easy induction on ϕ shows that

$$\|\phi\|^{\Omega}_{\alpha} = \phi/\equiv$$

for all sentences ϕ, and hence as $p \equiv \chi_p$ $(p \in P)$ we get (*).

6.4 Ω-EXPANSIONS. Given an infinite structure $\mathcal{M}$ for $\mathcal{L}$ and an M-valuation system Ω, an *Ω-expansion* $(\mathcal{M}, \varepsilon^{\Omega}_{M})$ of $\mathcal{M}$ is determined by

$$\varepsilon^{\Omega}_{M}: M \times M \to \Omega.$$

Relative to such an expansion the auxiliary semantics assigns $\|\phi\|^{\Omega} \in \Omega$ to each sentence ϕ of $\mathcal{L}^*(M)$ as follows:

$$\|R(t_1,\ldots,t_n)\| = \begin{cases} T & \text{if } R_{\mathcal{M}}(t_1^*,\ldots,t_n^*) \\ F & \text{otherwise} \end{cases}$$

if R is an n-place relation symbol of $\mathcal{L}$ or is equality.

$$\|t_1 \,\varepsilon\, t_2\| = \varepsilon_M^{\Omega}(t_1^*, t_2^*)$$

$$\|\neg\phi\| = -\|\phi\|$$

$$\|\phi \to \psi\| = \|\phi\| \supset \|\psi\|$$

$$\|\forall x\phi(x)\| = \bigwedge_{m\varepsilon M} \|\phi(\bar{m})\|$$

$$\|\phi \equiv \psi\| = E(\|\phi\|, \|\psi\|)$$

Associated with each Ω-expansion $(\mathcal{M}, \varepsilon_M^{\Omega})$ is an expansion $\mathcal{M}^*$ of $\mathcal{M}$ in the sense of 4.3, where

(1) $(m \,\varepsilon_{\mathcal{M}}\, n) \Leftrightarrow \varepsilon_M^{\Omega}(m, n) = T$, for $m, n \,\varepsilon\, M$,

(2) $\phi \equiv_{\mathcal{M}} \psi \Leftrightarrow \|\phi\|^{\Omega} = \|\psi\|^{\Omega}$.

Condition (3) of 4.3 is easily seen to hold. Moreover we have the following result.

THEOREM. $\mathcal{M}^*$ is always a model of (1), (2) and (3) of 2.1 and is a model of (4) if Ω is maximal regular.

We now review the construction in §4 of an enlargement $\mathcal{M}^*$ of $\mathcal{M}$ that is a model of (AP). If $\mathcal{M}^*$ is to be constructed via an Ω-expansion $(\mathcal{M}, \varepsilon_M^{\Omega})$ it suffices to find a maximal regular Ω and $\varepsilon_M^{\Omega}\colon M \times M \to \Omega$ such that

(†) $\qquad \varepsilon_M^{\Omega}(m,n) = \|\phi(\bar{m})\|^{\Omega}$

whenever $n = \{x|\phi(x)\}_M$.

If P is as in 4.4 and $p \mapsto p^0$ as in 4.5 then let Ω and $\sigma\colon P \to \Omega$ be as given by the theorem of 6.3. Then

(*) $\qquad \sigma(\bar{m} \,\varepsilon\, \bar{n}) = \|\phi(\bar{m})^+\|_{\sigma}^{\Omega}$

whenever $n = \{x|\phi(x)\}_M$.

Using σ we may define ε_M^{Ω} by

$$\varepsilon_M^{\Omega}(m,n) = \mathfrak{a}\,(\bar{m}\ \varepsilon\ \bar{n}) \text{ for } m,n\ \varepsilon\ M.$$

By a straightforward induction on sentences ϕ of $\mathcal{L}^*(M)$

$$||\phi||^{\Omega} = ||\phi^+||^{\Omega}_{\mathfrak{a}}.$$

Hence by (*), if $n = \{x|\phi(x)\}_M$

$$\begin{aligned}\varepsilon_M^{\Omega}(m,n) &= \mathfrak{a}\,(\bar{m}\ \varepsilon\ \bar{n})\\ &= ||\phi(\bar{m})^+||^{\Omega}_{\mathfrak{a}}\\ &= ||\phi(\bar{m})||^{\Omega},\end{aligned}$$

which is (†).

6.5 A MODIFICATION OF THE MAIN RESULT Modify the $\equiv$-laws in 2.1 by replacing (4)(vii) by

(vii)' $T(\phi \equiv \psi) \leftrightarrow D(\phi) \wedge D(\psi) \wedge (\phi \equiv \psi)$.

With this change the lemma of 2.1 still holds. In addition we also have the result that D is strongly closed with respect to $\equiv$, i.e.

$$D(\phi \equiv \psi) \leftrightarrow D(\phi) \wedge D(\psi).$$

We show that the main result of the paper, stated in 2.3, still holds. We shall follow the same pattern of proof as before, except that in the theorem of 6.4 'maximal' should now be 'minimal' and the theorem of 6.3 must be replaced by the following theorem. Instead of requiring a Church-Rosser theorem to find a suitable maximal regular Ω, we here simply use the unique minimal regular M-valuation Ω_0 on the three element set $\Omega_0 = \{T, F, U\}$.

THEOREM $\mathfrak{a}$ For any assignment $p \mapsto \bar{\chi}_p$ ($p\ \varepsilon\ P$) there is an assignment $\mathfrak{a} : P \to \Omega_0$ such that

$$\mathfrak{a}(p) = ||\chi_p||^{\Omega_0}_{\mathfrak{a}} \text{ for all } p\ \varepsilon\ P.$$

Proof Let $\sqsubseteq$ be the unique partial ordering of Ω_0 such that $U \sqsubseteq T, F$ and $T \not\sqsubseteq F$, $F \not\sqsubseteq T$.

This induces an inductive partial ordering (i.e. every chain has a lub) of Ω_0^P: For α, α': $P \to \Omega_0$

$$\alpha \sqsubseteq \alpha' \Leftrightarrow \alpha(p) \sqsubseteq \alpha'(p) \qquad (p \in P)$$

Notice that the operations of Ω_0 are all monotone, so that by induction on the way a sentence ϕ of $\mathcal{L}_M(P)$ is built up,

$$\alpha \sqsubseteq \alpha' \text{ implies } \|\phi\|_\alpha \sqsubseteq \|\phi\|_{\alpha'}.$$

Now given $p \mapsto \chi_p$ $(p \in P)$, for each $\alpha : P \to \Omega_0$ define $\alpha^*: P \to \Omega_0$ by

$$\alpha^*(p) = \|\chi_p\|_\alpha \text{ for } p \in P.$$

The assignment $\alpha \mapsto \alpha^*$ is a monotone operation on the inductive partial ordering of Ω_0^P, so that by the Tarski fixed point theorem it has a fixed point α. So $\alpha = \alpha^*$ and hence

$$\alpha(p) = \alpha^*(p) = \|\chi_p\|_\alpha \text{ for all } p \in P.$$

FOOTNOTES

1) A write-up of that talk was circulated in 1977 under the title "Theories of properties and functions with intensional equality operators."

2) This was written up and circulated under the same title as the present paper, except that 'comprehension' was used for 'abstraction'. The error there was in the definition 3.3(vi), which is not founded.

3) One can formulate a conservation result for arbitrary S by modifying S* to a two-sorted theory in which both $x \in Y$ and $X \in Y$ would be allowed.

ACKNOWLEDGEMENTS The research for and preparation of this paper by the second author were supported in part by the National Science Foundation (USA), as well as the Mathematical Institute, Oxford University while a Visiting Fellow at All Souls College.

REFERENCES

A. Church (1941). "The calculi of λ-conversion". Princeton Univ. Press.

S. Feferman (1975). Non-extensional type-free theories of partial operations and classifications. *Proof Theory Symposium, Kiel* 1974. *Lecture Notes in Maths.* v.500 (eds. J. Diller, G.H. Müller) 73-118, Springer Verlag.

S. Feferman (1977). Categorical foundations and foundations of category theory. *Logic, Foundations of Mathematics and Computability Theory* (eds. R.E. Butts, J. Hintikka), 149-169. D. Reidel Co.

J.Y. Girard (1973). Théorème de Church-Rosser pour un système de termes infinis. (Unpublished notes).

J. Klop (1977). A counterexample to the CR property of λ-calculus + DMM → M. (Unpublished notes, Mathematisch Instituut, Budapestlaan 6, Utrecht).

W. Maass (1975). Church-Rosser-Theorem für λ-Kalküle mit unendlich langen Termen. *Proof Theory Symposium, Kiel* 1974. *Lecture Notes in Maths.* 500 (eds. J. Diller, G.H. Müller) 257-263, Springer Verlag.

COMBINING UNRESTRICTED ABSTRACTION WITH UNIVERSAL QUANTIFICATION

P. C. Gilmore

Department of Computer Science
University of British Columbia
Vancouver, B.C. V6T 1W5
Canada

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

Since the discovery of an inconsistency in Frege's formal foundation for mathematical reasoning there has been a continuing effort to develop an intuitively sound and consistent set theory which would permit unrestricted abstraction within a full first order predicate logic. Curry's contributions to that effort have been considerable through his work on illative combinatorial logic, which, as explained in Chapter 8 of Curry and Feys (1958) was developed for this purpose.

My attempts at contributions to this effort have been made over a number of years beginning just prior to and during the years 1955-58 when I had the privilege of sharing an office and much time with my generous senior colleague, Haskell Curry, at Penn. State. He encouraged and criticized my attempts. In this paper I wish to report on a success that has previously been described only in the unpublished reports Gilmore(1971) and (1972). It uses methods very different from those used in Gilmore (1974).

In this paper the goal of a consistent set theory with unlimited abstraction within a framework of first order logic has not been achieved, but it has been achieved within a monadic second order logic. The power of the resulting formal theory

NST is considerable, being at least that of second order arithmetic. Also one of the two innovations needed to maintain the consistency of NST relates very closely to the method employed in illative combinatorial logic for maintaining consistency, in particular to the method used to maintain the consistency of the logic described in §21 of Church (1941). Thus although the implications of the results of this paper for the lambda calculus are not immediate they nevertheless provide a method for extending that formal logic to one with full universal quantification, or of extending NST to a logic with the full functional abstraction capability of the lambda calculus. A sketch of the method of extension is given in §5.

Although the original motivation for this work was set theory it received a second motivation with the urging of McCarthy(1963) for the development of a mathematical theory of computation satisfying some broadly stated requirements. It is clear that a formal set theory which is to provide a simple interpretation for a programming language like LISP and be powerful enough for proofs of correctness must have the character of a naive set theory in the tradition of Frege. For the unlimited functional abstraction of LISP, like that of the lambda calculus, permits the definition of functions which can take themselves as arguments, and reasoning about them requires full universal quantification. A hint of one kind of application of NST for correctness proofs of programs is given in §5.

In §2 an informal introduction to the theory NST is given. There the main difficulties in interpreting formulas of a naive set theory are discussed and one resolution suggested. The discussion relies heavily upon non-technical examples since such examples illustrate in a simple way the kind of interpretation to be given to NST.

It will be evident from this section that it is technically incorrect to call NST a set theory since no axiom of extensionality is introduced. Indeed it is plausible to conjecture that an extensionality axiom is inconsistent with NST as it was shown to be with the theory described in Gilmore(1974). The theory can therefore be more correctly called a theory of attributes.

Apart from the move to second order from first order logic, two technical innovations are needed to permit unlimited abstraction to consistently co-exist with universal quantification. The first of these is justified in §2. The motivation for it comes from a nominalistic interpretation of simple assertions like 'red is a color' in which a property is asserted of an abstraction like red. The innovation identifies the circumstances in which an abstraction term may be used autonomously as a name for itself and thereby appear to the left of ε.

The informal discussion of the semantics of NST is made precise in §3 and a lemma needed in the consistency proof of NST is proved.

Finally in §4 the formal proof theory of NST is described and proved consistent. The theory is presented as a Gentzen sequent calculus but could equally well be presented in any other of the so-called natural deduction systems of logic like Smullyan(1968) or Beth(1959). Here the second of the two innovations needed to permit unlimited abstraction to consistently co-exist with universal quantification is described. Briefly it limits the formulas S of the axioms $S \rightarrow S$ of the Gentzen(1934) sequent calculus to being atomic. Such a limitation does not introduce any restriction on the calculus when it is used to define first or second order logic, but it is essential to the consistency of NST. This restriction relates very closely to the means by which the theory described in §21 of Church(1941) is maintained consistent.

Another response to the need described by McCarthy(1963) is the work of Strachey, Scott and others described, for example, in Stoy(1977). The extent to which NST provides a fully formal means for studying the semantics of programming languages remains to be investigated, although a hint is provided in §5.

Two other technical questions also deserve investigation: Whether Henkin's proof of the completeness of second order logic, either in the form implicit in Henkin (1950 and 1953), or in the cut-elimination version of Prawitz(1967), can be adopted to prove the completeness of NST; and whether an intuitionistic version of NST, formalized within an intuitionistic Gentzen calculus, can provide foundations for a constructive analysis discussed in several of the papers of Myhill(1969).

2. INTERPRETING THE FORMULAS OF A NAIVE SET THEORY

This section is intended as an informal introduction to the theory NST. Here the main difficulties in interpreting formulas of a naive set theory are discussed and a resolution of them suggested.

The most elementary interpreted sentences of NST take the form

individual name ε attribute name

and are called atomic sentences. Such a sentence is understood to assert that the individual named by the individual name possesses the attribute named by the attribute name. For example, three such sentences taken from Quine(1953) are:

(1) Boston ε populous

(2) Boston ε disyllabic

(3) 'Boston' ε disyllabic.

The attribute names are 'populous' and 'disyllabic' while the individual names are 'Boston' and "Boston", the latter being a name for the word 'Boston'. The sentences are respectively true, false and true because the city of Boston is populous but is not disyllabic while the word 'Boston' is disyllabic.

An interpretation of NST assumes the existence of a collection I of individual names and a collection A of attribute names. Further it assumes that each atomic sentence formed by taking an individual name from I and an attribute name from A has been assigned a truth value true or false. Since interpretations of NST rather than the formal language NST are being discussed here it is not necessary to assume that the collections I and A are denumerable although there will never be a need in this paper for non-denumerable I.

The device of signed sentences will be used to abbreviate assertions about the truth or falsehood of a sentence. Rather than the letters T and F used by Smullyan (1968), which were suggested by the semantic tableaux of Beth(1959), the signs + and - will be used. Thus to assert $+a\varepsilon A$ is to assert that the truth value true has been assigned to the atomic sentence $a\varepsilon A$ while to assert $-a\varepsilon A$ is to assert that the truth value falsehood has been assigned. The notation permits the compact notation $\pm$.

Through semantic rules truth values are assigned to sentences other than the atomic. For example the semantic rule for the alternate denial connective $|$ is:

2.1. For any sentences K and L, $\pm(K|L)$ can be asserted if and only if one of $-K$ and $-L$, or respectively both of $+K$ and $+L$, can be asserted. Here $\pm(K|L)$ should be read $+(K|L)$ or $-(K|L)$.

Semantic rules for propositional connectives like negation $\sim$, disjunction $\vee$, conjunction &, conditional $\rightarrow$, and biconditional $\leftrightarrow$ can be inferred from their definition in context: $\sim K$ abbreviates $(K|K)$, $(K\vee L)$ abbreviates $(\sim K|\sim L)$, $(K\&L)$ abbreviates $\sim(K|L)$, $(K\rightarrow L)$ abbreviates $(\sim K\vee L)$ and $(K\leftrightarrow L)$ abbreviates $((K\rightarrow L)\ \&\ (L\rightarrow K))$.

As in second order logic two kinds of universal quantifiers are available in NST. The first kind employs lower case variables ranging over the elements of I while the second employs upper case variables ranging over the elements of A. To state the semantic rules for these quantifiers substitution operators $[a/x]$ and $[A/X]$ are used. The effect of these operators is to replace each free occurrence of x, respectively X, by a, respectively A. The semantic rules are then:

2.2. For any sentence (x)K, ±(x)K can be asserted if and only if +[a/x]K can be asserted for every a of I, or respectively -[a/x]K can be asserted for some a of I.

2.3. For any sentence (X)K, ±(X)K can be asserted if and only if +[A/X]K can be asserted for every A of A, or respectively -[A/X]K can be asserted for some A of A.

Semantic rules for existential quantifiers can be inferred from their definitions in context: (Ex)K abbreviates ∿(x)∿K and (EX)K abbreviates ∿(X)∿K.

The examples (1), (2), and (3) were used by Quine in an excellently brief and clear exposition of the distinction between use and mention. The word 'Boston' in (1) and (2) is being used while in (3) it is being mentioned. Enlarging on Quine's discussion note that 'populous' in (1) and 'disyllabic' in (2) and (3) are being used. Consider now the sentences

(4) 'disyllabic' ε disyllabic

(5) 'multisyllabic' ε multisyllabic

In (4) the first occurrence of 'disyllabic' is being mentioned while the second is being used. Similarly for 'multisyllabic' in (5).

For (4) and (5) to be atomic sentences in an interpretation with individual names I and attribute names A it is necessary that 'disyllabic' and 'multisyllabic' be members of A while "disyllabic" and "multisyllabic" are members of I.

Single quotes around an attribute name is a device for creating an individual name which denotes the attribute name. The device is not applicable to second order variables. For example

(6) (Y)(∿'Y'ε disyllabic ∨'Y'ε disyllabic)

is meaningless unless 'Y' is to be taken as an individual name for the letter Y and the quantifier in front of (6) is taken to be vacuous. However the use of quotes around Y in (6) suggests something else, namely that every attribute name in A has an individual name in I. But that suggestion must in general be rejected since A may have cardinality greater than that of I. But more important the use of quotes in (6) is an abuse of the quote convention: 'Y' is somehow to be a function with arguments from A and values in I while no indication has been given as to how such a function can be defined. Church(1956) emphasizes this point in footnote 136 on page 62.

Under the quote convention the name of an attribute appearing to the left of ε in an atomic sentence must appear within single quotes since only individual names may appear in that position. Because there are no exceptions to this edict the quote convention for attribute names can be replaced by a position convention: An attribute name appearing to the left of ε in an atomic sentence is used autonomously, that is as a name for itself. For example under this convention (4) and (5) become

(7) disyllabic ε disyllabic

(8) multisyllabic ε multisyllabic

The first occurrence of 'disyllabic' in (7) is used autonomously as a name for itself while the second occurrence is used as an attribute name. Similarly for 'multisyllabic' in (8).

Under this position convention any attribute name in A which is to be given an individual name is simply added to I since an attribute name in I is an individual name for itself. But as was noted before not every attribute name in A can be assumed to have an individual name in I so that I need not include A. For this reason

$$(Y)(\sim Y\varepsilon \text{ disyllabic} \vee Y\varepsilon \text{ disyllabic})$$

has no more meaning than (6) and will not be admitted as a well formed sentence.

It would appear that there is a class of elementary sentences similar to the atomic sentences which cannot be expressed using the position convention. Consider for example the sentence

(9) red is a color.

Conventionally interpreted this sentence asserts that the attribute named by the word 'red' possesses the attribute of being a color. Using 'color' as a name for the attribute of being a color one might try to express it as

(10) red ε color.

But then (10) is false for under the position convention it asserts that the word 'red' is a color.

The difficulty of expressing (9) as (10) arises only because the conventional interpretation of (9) is platonic, while under a nominalistic interpretation of (9) the difficulty vanishes.

The word 'red' appearing in atomic sentences

(11) a ε red

is used as a name for an attribute. Through experience with sentences like (11) the correct use of the word 'red' is learned. Another interpretation of (9) asserts, therefore, that it expresses something about the use of the word 'red' rather than something about the attribute red. Under this interpretation (9) has the same meaning as

(12) 'red' is a color word.

Now with 'color' used as a name for the attribute of being a color word, (9) is expressed by (10).

There are, of course, profound objections to a nominalist interpretation (12) of a sentence like (9). However this is not the place to review the disputes between platonic realists and nominalists which are almost as ancient as philosophy. The models to be proposed for NST can be understood without touching upon these disputes. A reader wishing a less superficial discussion may consult several chapters of Quine (1953), or consult Henkin (1953) for a technical paper describing interpretations for second order and higher order logics related to the interpretations provided here for NST.

Lower and upper case variables have been introduced as variables over the domains I and A respectively. An attribute name replacing an occurrence of an upper case variable in a formula is used in the context in which it appears, used that is as an attribute name. An attribute name replacing an occurrence of a lower case variable in a formula is mentioned in the context in which it appears since it occurs as a name for itself. For this reason lower case variables were referred to in Gilmore (1971) as variables of mention and upper case variables as variables of use. However to emphasize the connections with second order logic they will here be referred to as first and second order variables. One last kind of variable, a variable of abstraction, remains to be introduced.

A name of an attribute from A is thought of as an irreducible linguistic expression whose correct use is learned from experience with atomic sentences in which the attribute name is used, in a manner similar to the way in which the correct use of 'red' is learned from experience with sentences like (11). However common to language in general and mathematics in particular are complex attribute expressions whose use is indicated by their structure. One such expression is $\{\mu:(X)(\mu\varepsilon X\leftrightarrow\mu\varepsilon X)\}$, which will be abbreviated by V, while another is $\{\mu:\sim\mu\varepsilon\mu\}$, which will be abbreviated by R. Here μ is being used as a variable of abstraction which is allowed to occur on either side of ε. More generally if K is any formula and μ any variable of abstraction then $\{\mu:K\}$ is a complex attribute expression called an abstraction term. The use of abstraction terms is prescribed by the following semantic rule:

(13) For any a in I, $\pm a\varepsilon\{\mu:K\}$ can be asserted if and only if $\pm[a/\mu]K$, respectively, can be asserted.

Like the semantic rules 2.1, 2.2 and 2.3, the rule (13) is to be understood as giving conditions under which a complex signed sentence, in this case $\pm a\varepsilon\{\mu:K\}$, can be asserted, the conditions being expressed in terms of simpler signed sentences, in this case $\pm[a/\mu]K$. For example it quickly follows that +Boston ε V, +disyllabic ε V and +multisyllabic ε V can all be asserted. Further it follows that -disyllabic ε R and +multisyllabic ε R can be asserted assuming (7) is false and (8) is true. However neither +Boston ε R nor -Boston ε R can be asserted since Boston ε Boston is not an atomic sentence and therefore neither true nor false. Thus in general an abstraction term like R differs from an attribute name like 'disyllabic' in as much as $+(x)(x\varepsilon R \vee \sim x\varepsilon R)$ cannot be asserted. Further it must be carefully noted that the introduction of abstraction terms does not change the semantic rule 2.3 since abstraction terms are not members of A.

The autonomous use convention for attribute names to the left of ε can be extended to abstraction terms like V and R in which no second order variables occur free and in which only names from I occur. In effect such abstraction terms are admitted into every I as names for themselves. Thus sentences like $V\varepsilon V$, $V\varepsilon R$ and $R\varepsilon R$ can now be formed.

Since $+(X)(V\varepsilon X\leftrightarrow V\varepsilon X)$ can be asserted it follows that $+V\varepsilon V$ and therefore $-V\varepsilon R$ can be asserted in every interpretation. However neither $+R\varepsilon R$ nor $-R\varepsilon R$ can be asserted in any interpretation, since to assert one it is necessary to be able to assert the other.

Each I is assumed to contain every abstraction term in which no variables occur free and in which only names from I occur. Consequently signed sentences like

$$+(X)(y)(\{\mu:y\varepsilon\mu\}\varepsilon X\leftrightarrow\{\mu:y\varepsilon\mu\}\varepsilon X)$$

can be asserted in every interpretation. However abstraction terms in which a second order variable occurs free may not appear to the left of ε in well formed sentences just as second order variables themselves could not so appear. To allow them to appear would not only introduce difficulties of interpretation like those introduced by (6), but would also render NST inconsistent.

Generalized abstraction terms of a kind not hitherto considered may also be given meaning by a semantic rule which is a generalization of (13). Let t be a term in which only the abstraction variables $\mu_1,\ldots,\mu_n$ occur free. Then for any formula K, $\{t:K\}$ is admitted as an abstraction term with a use prescribed by:

2.4. For any $a_1,\ldots,a_n$ from I,

$\pm([a_1,\ldots,a_n/\mu_1,\ldots,\mu_n]t) \varepsilon \{t:K\}$ can be asserted if and only if $\pm[a_1,\ldots,a_n/\mu_1,\ldots,\mu_n]K$, respectively, can be asserted. Here $[a_1,\ldots,a_n/\mu_1,\ldots,\mu_n]$ is a simultaneous substitution operator equivalent to $[a_1/\mu_1]\ \ldots\ [a_n/\mu_n]$.

The complex attribute expressions $\{\mu:K\}$ previously introduced are examples of these new abstraction terms. Another important class of examples appears when t is taken to be the ordered pair $<\mu_1,\mu_2>$, which can be defined by well-known methods described for example in Quine(1951). Then $\{<\mu_1,\mu_2>:K\}$ is a complex relation expression, that is a complex attribute expression which can be applied only to ordered pairs. The variables μ_1 and μ_2 in $\{<\mu_1,\mu_2>:K\}$ are bound like the variable μ in $\{\mu:K\}$. These generalized abstraction terms permit an easy definition or relational abstraction and thus make it possible to have only one kind of second order variable needed for monadic second order logic rather than the multiplicity of variables usually needed for the full second order quantificational logic Church(1956).

It is important to note that these new abstraction terms are a significant generalization of the simple terms like $\{\mu:K\}$. Consider an example. Let $t(\alpha)$ be a term and $K(\alpha)$ a formula in which the abstraction variable α is the only variable occurring free. Let Q be the term $\{t(\alpha):K(\alpha)\}$. Then to prove $t(a) \varepsilon Q$ it is sufficient to prove $K(a)$, while to prove $\sim t(a) \varepsilon Q$ it is sufficient to prove $\sim K(a)$. Thus Q must be distinguished from the term P defined to be $\{\beta:(Ex)(\beta=t(x)+K(x))\}$, where = has been defined in the usual way for second order logic. For although $t(a) \in P$ can be concluded from $K(a)$, $\sim t(a) \in P$ cannot be concluded from $\sim K(a)$. Thus the generalized terms are symmetric to negation in a way in which terms like P are not. Of course they introduce potential new inconsistencies, but it is demonstrated below that these do not in fact arise.

Given sets $\mathcal{I}$ and $\mathcal{A}$ an atomic sentence is an expression of the form $p\varepsilon A$, with p from $\mathcal{I}$ and A from $\mathcal{A}$. It was assumed that an assignment of truth values to atomic sentences could be made so that each atomic sentence received exactly one truth value. Such an assignment is called a base. Thus a base $\mathcal{B}$ is a set of signed atomic sentences such that for each p from $\mathcal{I}$ and A from $\mathcal{A}$ exactly one of $+p\varepsilon A$ and $-p\varepsilon A$ is in $\mathcal{B}$. Through use of the semantic rules 2.1, 2.2, 2.3 and 2.4 a sentence which is not atomic can be assigned at most one truth value. Given a base $\mathcal{B}$ let $\mathcal{B}^*$ be the set of all signed sentences $\pm K$ for which K is assigned the truth value true, respectively false, by the semantic rules. $\mathcal{B}^*$ is called the semantic closure of $\mathcal{B}$, and is formally defined in the next section.

From experience with first and second order semantics one might expect that $\mathcal{B}^*$ was closed with respect to changes of bound variables. That is if S' is obtained from S by proper changes of bound variables then S' is in $\mathcal{B}^*$ also. Clearly however that need not be the case since $\mathcal{B}$ need not be so closed. It may be that changes of bound variables are not needed for the purposes to which NST is to be put. However in this paper a version of NST is described in which such changes can be made by restricting attention to bases which are closed with respect to changes of bound variables.

Although it has been said that some sets $\mathcal{A}$ may have more members than $\mathcal{I}$ no assumption applying to all sets of attribute names $\mathcal{A}$ has been made, apart from an implicit assumption that they are non-empty. The one assumption about to be stated will finally define the models of NST - they are the bases satisfying the assumption.

Consider particular sets I and A and a particular base B. Each closed abstraction term r, that is each abstraction term in which no variables occur free, defines an attribute which applies to those members p of I for which $+p\varepsilon r$ is in B^*. For a given r the attribute defined by r may or may not have a name in A. More generally there may or may not be an attribute name C in A completing r in the sense that for each p, if $\pm p\varepsilon r$ is in B^* then $\pm p\varepsilon C$, respectively, is in B. A model-base, however, is a base for which each closed abstraction term has an attribute name completing it. Model-bases are analogous to the normal systems of domains for second order logic defined in Henkin(1950) or Church(1956) and to the general models for the theory of types defined in Prawitz(1967).

That there exist model-bases can be established in several ways. However the most immediate way is the following. Given I consider all subsets δ of I closed under changes of bound variables; that is all subsets δ such that if p is in δ and p' is obtained from p by changes of bound variables then p' is in δ. Let A include an attribute name C_δ for each such subset δ. Let B be the set of all signed atomic sentences $\pm p\varepsilon C_\delta$ for which p is, respectively is not, a member of δ. Clearly B is a model-base.

The assumption that given any I the set of all subsets of I can be formed is essential to the consistency proof of NST given below. The assumption introduces a non-constructive step into the consistency proof.

3. THE MODELS OF NST

In the preceding section the reader's experience with first and second order logic was relied upon for an understanding of the syntax and semantics of NST. In this section details will provided and a lemma proved which is needed for the consistency proof of NST given in Section 4.

The primitive symbols of the formal theory NST include denumerably many variables of each of the three kinds, first and second order variables and abstraction variables, and at most denumerably many constants of each of the two orders. A second order constant may or may not be also a first order constant. In addition ε, $|$, (,), {,}, and :, are primitive symbols.

In the last section distinct notations were maintained for first and second order variables and abstraction variables, and for first and second order constants. However for the more formal parts of this paper such distinct notations become cumbersome. In general therefore letters u, v, w, x, y, and z will be used with and without subscripts and superscripts as variables of any kind, and the letters a, b, c, and d with and without subscripts as constants of either kind, and the context will be relied upon to indicate what kind of variable or constant is intended. Occasionally for emphasis upper case letters will be used as second order variables or constants.

3.1. *Definition of term and formula*

1. Any first order variable or constant and any abstraction variable is a first order term. Any second order variable or constant and any abstraction variable is a second order term. Any occurrence of a variable is a free occurrence in itself.
2. If p is a first order term and r a second order term then $p\varepsilon r$ is a formula. Any free occurrence of a variable in p or r is a free occurrence in $p\varepsilon r$.
3. If K and L are formulas then $(K|L)$ is a formula. Any free occurrence of a variable in K or L is a free occurrence in $(K|L)$.
4. If K is any formula and x a first or second order variable

then (x)K is a formula. Any free occurrence of a variable other than x in K is a free occurrence in (x)K.

5. If K is any formula and t any first order term then {t:K} is a second order term. If no second order variable occurs free in K and no second order constant which is not also a first order constant occurs in K then {t:K} is also a first order term. Any free occurrence in K of a variable which is not an abstraction variable also occurring free in t, or any free occurrence of a first order variable in t, is a free occurrence in {t:K}.

3.2. *Definition*

An atomic formula is a formula rεs for which s is a second order variable or constant. A signed formula is a formula prefixed with + or -. A closed formula or term is a formula or term in which no variable occurs free. A sentence is a closed formula.

In the remainder of this section the practice will be continued of extending the language of NST to include elements of sets I and A for the semantics of NST. Elements of I are admitted as first order constants and elements of A as second order constants. It is tedious to have to refer frequently to the sets I and A which most of the time can be taken to be any sets with denumerably and at least denumerably many members respectively. Consequently the terminology introduced in the last section will be continued: A closed first order term of the extended language will be referred to as an individual name while a second order constant (note <u>not</u> a closed second order term) is referred to as an attribute name. For example the next definition presumes that some I and A have been given so that the set of individual names and attribute names is given.

3.3. *Definition*

A base is a set B of signed atomic sentences such that:

1. For each individual name p and attribute name A, exactly one

of $+p\varepsilon A$ and $-p\varepsilon A$ is in $\mathcal{B}$; and

2. For each pair of individual names p and p' for which one is a obtainable from the other by a proper change of bound variables, if $\pm p\varepsilon A$ is in $\mathcal{B}$ then so also is $\pm p'\varepsilon A$ respectively.

The next definition prepares for the definition of semantic closure of a base. Note the correspondence between the four clauses of the definition and the four semantic rules 2.1, 2.2, 2.3 and 2.4.

3.4. *Definition*

The semantic successor of a set $\mathcal{S}$ of signed sentences is the set of all signed sentences S satisfying one of the following conditions.

1. S is $\pm(K|L)$ and one of $-K$ and $-L$, respectively both $+K$ and $+L$, are in $\mathcal{S}$.
2. S is $\pm(x)K$, where x is a first order variable, and $+[p|x]K$ for all individual names p, respectively $-[p\ x]K$ for some individual name p, is in $\mathcal{S}$.
3. S is $\pm(X)K$, where X is a second order variable, and $+[A|X]K$ for all attribute names A, respectively $-[A|X]K$ for some attribute name A, is in $\mathcal{S}$.
4. S is $\pm([a_1,\ldots,a_n/u_1,\ldots,u_n]t)\ \varepsilon\ \{t:K\}$ and $\pm[a_1,\ldots,a_n/u_1,\ldots,u_n]K$, respectively, is in $\mathcal{S}$ where $u_1,\ldots,u_n$ are all the variables of abstraction occurring free in t, and $a_1,\ldots,a_n$ are individual names.

3.5. *Definition*

The semantic closure $\mathcal{B}^*$ of a base $\mathcal{B}$ is the union of all the sets $\mathcal{B}_i$, $0\leq i$, where $\mathcal{B}_o$ is $\mathcal{B}$ and $\mathcal{B}_{i+1}$ is the union of $\mathcal{B}_i$ and the semantic successor of $\mathcal{B}_i$.

It was noted earlier that the semantic rule 2.3 defines the range of the second order variables to be the set A, the set of all attribute names, and does not include the abstraction terms. That these terms are not in A is essential to the proof of the following theorem:

Theorem 3.6. For any base B and integer i, B_i has the following two properties:

.1. Let K be any formula and let x be a first order variable, respectively a second order variable, occurring free in K. Let one of [c/x]±K be in B_i for some individual, respectively attribute, name c. Then for any individual, respectively attribute, name d one of [d/x]±K is also in B_i.

.2. For no sentence K and bound variable variant K' of K are both +K and -K' in the set.

Proof. That B_0 has the two properties follows from the definition of base. That B_{i+1} will have the two properties when B_i has them follows from the definition of semantic successor.[1]

Corollary. The semantic closure B* of a base B is closed under semantic successor and has property 2.

3.7. *Definition*

Let B be a base with semantic closure B* and let r be any closed second order term of B. An attribute name C is a completion of r for B if and only if for every individual name p if ±pεr is in B* then ±pεC, respectively, is in B.

3.8. *Definition*

A base B is a model-base if and only if every closed second order term of B has a completion for B.

At the end of section two a proof was provided for the following lemma; that proof cannot be improved upon here.

Lemma 3.9. There exists a model-base.

In the next section the formal theory NST will be described. Its rules of deduction in all but one case are formal expressions of the semantic rules 2.1, 2.2, 2.3 and 2.4. The one exception is with that part of the rule 2.3 dealing with the assertability of -(X)K. The following lemma provides the connection for model-bases between that part of rule 2.3 and the corresponding rule of deduction of NST.

Lemma 3.10. Let $\mathcal{B}$ be a model-base and $\mathcal{B}^*$ its semantic closure. For any formula K and closed second order term r, if $-[r/X]K$ is in $\mathcal{B}^*$ then so also is $-(X)K$.

Proof. Let C_r be an attribute name which is a completion of r for $\mathcal{B}$. To prove the lemma it is sufficient to prove by induction on i: For any signed formula S if $[r/X]S$ is in $\mathcal{B}_i$ then so also is $[C_r/X]S$. For if $-[C_r/X]K$ is in $\mathcal{B}_i$, then $-(X)K$ is in $\mathcal{B}_{i+1}$ and hence in $\mathcal{B}^*$. Clearly X may be assumed to occur free in S.

1. Let $i = 0$. S must be of the form $\pm p\varepsilon X$, where p is an individual name. r must be an attribute name. Hence if $\pm p\varepsilon r$ is in $\mathcal{B}_0$ so also is $\pm p\varepsilon C_r$ respectively.
2. Assume for all S and r that if $[r/X]S$ is in $\mathcal{B}_i$ then $[C_r/X]S$ is in $\mathcal{B}_i$ also. Let now r be a closed second order term and S a signed formula for which $[r/X]S$ is in $\mathcal{B}_{i+1}$ and consider the 6 possible forms that S can take namely $\pm(K|L)$, $\pm(x)K$ where x is a variable, first or second order, and $\pm p\varepsilon s$. Since the arguments are similar for all these cases only one of them will be given namely for the case that S is $+p\varepsilon s$, where necessarily X occurs free in s and p is an individual name. Consider first the case where s is X. If $[r/X] + p\varepsilon X$ is in $\mathcal{B}_{i+1}$ then, since C_r is a completion of r, $[C_r/X] + p\varepsilon X$ is in $\mathcal{B}_0$, and hence in $\mathcal{B}_{i+1}$.

   There remains to be considered the case where s is of the form $\{t:K\}$. If $[r/X] + p\varepsilon\{t:K\}$ is in $\mathcal{B}_{i+1}$ then p is of the form $[a_1,\ldots,a_n/u_1,\ldots,u_n]t$, where $u_1,\ldots,u_n$ are all the variables of abstraction occurring free in t and $a_1,\ldots,a_n$ are individual names, and $[r/X] + [a_1,\ldots,a_n/u_1,\ldots,u_n]K$ is in $\mathcal{B}_i$. From the induction assumption therefore $[C_r/X] + [a_1,\ldots,a_n/u_1,\ldots,u_n]K$ is in $\mathcal{B}_i$ and hence $[C_r/X] + p\varepsilon\{t:K\}$ is in $\mathcal{B}_{i+1}$.

4. THE FORMAL THEORY NST

In the previous sections the semantics for the theory NST has been developed. The semantics was described in general mathematical terms without any regard to a formal proof theory. For example the definition of semantic successor introduced what are essentially rules of deductions some of which have infinitely many premisses while in a fully formal theory such rules may have only finitely many. The systems described by Fitch in(1967) and (1974) for example, are not fully formal in this sense. In this section a fully formal theory NST will be introduced.

The theory NST is a theory of Gentzen sequents, where a Gentzen sequent is understood to be any finite set of signed formulas. Using the conventional Gentzen(1934) notation, a sequent $K_1,\ldots,K_m \rightarrow L_1,\ldots,L_n$ is the set $\{-K_1,\ldots,-K_m,+L_1,\ldots,+L_n\}$.

For any set δ of signed formulas and any signed formula S, the notation (δ,S) expresses the union of δ and $\{S\}$.

4.1. *Definitions*

A sequent is derivable in NST if and only if it is the last element of a derivation, where a derivation is a finite sequence $\delta_1,\ldots,\delta_n$ of sequents δ_k such that for each k, $1 \leq k \leq n$, one of the following cases apply.

1. For some first order term p, bound variable variant p' of p, and some second order variable or constant s, both $+p\varepsilon s$ and $-p'\varepsilon s$ are in δ_k.
2. For some signed formula S and sequent δ, δ_k is (δ,S) and for indices i and j, $i,j < k$, one of the following is true:
   1. S is $\pm(K|L)$ and δ_i is $(\delta,-K)$ or $(\delta,-L)$, respectively δ_i is $(\delta,+K)$ and δ_j is $(\delta,+L)$.
   2. S is $+(x)K$, and for some variable y of the same order as x free to replace x in K and not occurring free in any member of δ_k, δ_i is $(\delta,+[y/x]K)$.
   3. S is $-(x)K$ and for some term t of the same order as x and free to replace x in K, δ_i is $(\delta,-[t/x]K)$.
   4. S is $\pm([r_1,\ldots,r_n/u_1,\ldots,u_n]t)\ \varepsilon\ \{t{:}K\}$ and δ_i is $(\delta,\pm[r_1,\ldots,r_n/u_1,\ldots,u_n]K)$ respectively, where $u_1,\ldots,u_n$ are all the variables of abstraction occurring free in t

and $r_1,\ldots,r_n$ are any first order terms free to replace $u_1,\ldots,u_n$ in K.

3. For indices i and j, $i,j < k$, and some formula M, δ_i is $(\delta_k,+M)$ and δ_j is $(\delta_k,-M)$.

Note that 4.1.1 contains a slight restriction of the usual Gentzen axiom which allows δ_k to include +S and -S for any formula S, not just for an atomic S as in the case here. This is no restriction for derivability in first or second order logic but it is a critical restriction for NST.

It is now necessary to discuss both formulas of the formal theory NST and formulas of the language extended for semantics. By a formula, signed or not, of NST is meant one of the formal theory NST. By a formula, signed or not, of a base $\mathcal{B}$ is meant one in a language extended by the first and second order constants of $\mathcal{B}$. Similarly a sequent of NST or $\mathcal{B}$ is a finite set of signed formulas of NST or $\mathcal{B}$ respectively. A closed sequent is a sequent of sentences only, where as before a sentence is a formula in which no variable occurs free.

4.2. *Definitions*

Let $\mathcal{B}$ be any base, δ any sequent of NST and δ' any closed sequent of $\mathcal{B}$.

1. δ' is satisfied in $\mathcal{B}$ if and only if some element of δ' is in the semantic closure of $\mathcal{B}$.
2. δ' is a substitution instance of δ if and only if δ' is obtained from δ by replacing each free occurrence of a first order variable in a signed formula of δ by an individual name, each free occurrence of a second order variable by an attribute name, and each free occurrence of a variable of abstraction by a constant which is both first and second order, in each case the same name or constant $\mathcal{B}$ replacing the same variable throughout the signed formulas of δ.
3. δ is valid in $\mathcal{B}$ if and only if each substitution instance δ' of $\mathcal{B}$ is satisfied in $\mathcal{B}$.

Theorem 4.3. Each derivable sequent of NST is valid in every model-base.

Proof. Let $\mathcal{B}$ be any model base, $\mathcal{B}^*$ its semantic closure, and $\delta_1,\ldots,\delta_n$ a derivation. By induction on n, δ_n will be shown to be valid in $\mathcal{B}$.

1. Let n = 1. Then for some first order term p, bound variable variant p' of p, and some second order variable or constant s, both $+p\varepsilon s$ and $-p'\varepsilon s$ are in δ_1. Consequently every substitution instance δ_1' of δ_1 is satisfied in $\mathcal{B}$.
2. Assume that if $1 \leq n < k$ then δ_n is valid in $\mathcal{B}$. Let now $\delta_1,\ldots,\delta_k$ be a derivation and consider the possibilities for δ_k:
   1. δ_k is itself a derivation. Then the argument of 4.3.1 proves δ_k is valid in $\mathcal{B}$.
   2. For some signed formula S and sequent δ, δ_k is (δ,S) and for indices i and j, $i,j < k$, one of the four cases of 4.1.2 applies. Since the arguments for each of these cases is similar only the arguments for the most difficult of the cases will be given, namely 4.1.2.2 and 4.1.2.3 when x is a second order variable. Consider the case 4.1.2.2 first. Let δ_k' be any substitution instance of δ_k and let δ' and +(x)K' be the corresponding substitution instances of δ and +(x)K respectively. If δ' is satisfied in $\mathcal{B}$ then δ_k' is satisfied in $\mathcal{B}$. Assume therefore that δ' is not satisfied in $\mathcal{B}$. By the induction assumption δ_i is valid in $\mathcal{B}$. Consequently $+[A|x]K'$ is in $\mathcal{B}^*$ for every attribute name A so that +(x)K' is in $\mathcal{B}^*$ also. Hence δ_k' is satisfied in $\mathcal{B}$.

      Consider now the case 4.1.2.3. Let δ_k' by any substitution instance of δ_k and let δ', -(x)K' and t' be the corresponding substitution instances of δ, -(x)K and t. Should t' contain free occurrences of variables it will be assumed that they have all been replaced by appropriate constant terms so that t' is a closed second order term. If δ' is satisfied in $\mathcal{B}$ then δ_k is satisfied in $\mathcal{B}$. Assume therefore that δ' is not satisfied in $\mathcal{B}$. By the induction assumption δ_i is valid in $\mathcal{B}$. Consequently $-[t'|x]K'$ is in $\mathcal{B}^*$ and therefore by lemma 3.10 $-(x)K'$ is in $\mathcal{B}^*$ also so that δ_k is satisfied in $\mathcal{B}$.

3. For indices i and j, $i,j < k$, and some formula M, δ_i is $(\delta_k,+M)$ and δ_j is $(\delta_k,-M)$. Let δ_k' and M' be substitution instances of δ_k and M, and let δ_i' and δ_j' be the corresponding substitution instances of δ_i and δ_j. Since both δ_i' and δ_j' are satisfied in $\mathcal{B}$ and since by the corollary to theorem 3.6 not both +M' and -M' can be in $\mathcal{B}^*$, it follows that δ_k' is satisfied in B.

4.4. Corollary to 4.3. NST is consistent in the sense that the empty sequent is not derivable.

Proof. By lemma 3.9 there is a model-base. Since the empty sequent is not satisfied in any model-base it cannot be derived in NST.

As remarked earlier, the only non-constructive step in this proof of consistency is in the proof of lemma 3.9. For that proof it is assumed that a certain power set exists, namely the set of all subsets of closed under changes of bound variables.

Another kind of consistency proof, which at the same time would yield a proof of completeness, would be through the proof of the redundancy of the cut-rule of deduction of NST, the rule 4.1.3. This was the manner in which Gentzen (1934) proved the completeness of first order logic and in which Prawitz (1967) proved the completeness of second order logic.

5. AN APPLICATION AND EXTENSION

To develop arithmetic in second-order logic it is only necessary to assume that there is a first order constant 0 and a first order function ', the successor function. A predicate x is an integer, expressed as $x \in N$, can then be defined:

5.1. $x \varepsilon N$ for $(Y)(0 \varepsilon Y \ \& \ (u)(u\varepsilon Y \supset u'\varepsilon Y) \supset x \varepsilon Y)$

With this definition mathematical induction becomes a theorem of second-order logic.

Within NST 0 and ' can be defined in many ways. One of the simplest is to take 0 to be the null set and x' to be the singleton set of x. Alternatively the counter set of x as defined in paragraph 45 of Quine (1951) can be used as the definition of x'. In any case the definition 5.1, or better the definition of N as an abstraction term in the obvious way, assures that full second-order arithmetic is obtained. Details are provided in Gilmore (1972).

Definition 5.1 is typical of a fixed-point definition that is much used in correctness proofs of computer programs Manna(1974). Because of the device of admitting some second order abstraction terms as first order terms, "successor functions" of greater complexity than ' above can be defined, corresponding fixed-point definitions given and inductive arguments justified within NST without the need for additional axioms.

The most obvious extension needed for NST is the introduction of functional abstraction and application. A common method of introducing these concepts into a set theory is through definition in context, as is done in Quine(1951). The lack of an axiom of extensionality does not hinder the introduction of appropriate definitions in NST, but the fact that the definitions are given in an extended second order logic does affect the power of the resulting concepts.

The definition of functional abstraction is immediate:

5.2. $(\lambda\alpha p)$ for $\{<\beta,\alpha>:\beta=p\}$

Here p is a first order term, β is an abstraction variable distinct from α and any occurring free in q, and identity and ordered pair are assumed defined.

Functional application could be defined in first order contexts as follows:

5.3. $\ldots(P.q)\ldots$ for $(y)((y,q)\varepsilon P \supset \ldots y\ldots)$

Here q is a first order term and P a second order term that is also a first order term, and y is a first order variable not occurring free in q or P.

With these definitions it can be shown that functional abstraction and application interact in the appropriate way. However the definitions do not permit repeated functional applications of the form ((P.q).r) for the removal of (P.q) would then require that a first order variable be used in a second order position. Consequently an alternative is needed to the definition 5.3. The alternative chosen is to introduce functional application as a new primitive of the language:

5.4. If p and q are first order terms then (p.q) is a first order term.

Then the λ-conversion rules of Church (1941) are introduced as new rules of deduction.[2] A consistency proof of the resulting extended theory can be developed from the proof given in this paper.

Another kind of extension of NST is not possible. The technical innovations which distinguish NST from monadic second order logic can be employed in higher order logic as well. However these higher order forms of NST do not admit clear interpretations and indeed in Gilmore (1972) it is shown that the third order form of NST is already inconsistent.

FOOTNOTES

1. A referee has suggested that attention should be drawn to the following fact upon which the proof depends:

   Let S be a formula of the form introduced in 3.4.4. The individual names $a_1,\ldots,a_n$ can be recaptured from S because t can be recaptured from $\{t\varepsilon K\}$ and each u_i must occur free in t, so that each a_i can be recaptured from $[a_1,\ldots,a_n/u_1,\ldots,u_n]t$.

2. This removes the need for the special treatment given the changing of bound variables in the present paper as it can be introduced as one of the conversion rules.

ACKNOWLEDGEMENTS

The research in this paper was supported in part by a grant from the National Science and Engineering Research Council of Canada.

REFERENCES

Beth, E.W. (1959), The Foundations of Mathematics, North-Holland, Amsterdam.

Church, Alonzo (1941), The Calculi of Lambda Conversion, *Annals of Mathematics Studies*, No. 6, Princeton.

Church, Alonzo (1956), Introduction to Mathematical Logic, Princeton University Press, Princeton, N.J.

Curry, Haskell, B. and Feys, Robert (1958), Combinatory Logic Volume I, North-Holland Publishing Company, Amsterdam.

Fitch, Frederic B. (1967), A Complete and Consistent Modal Set Theory, *Journal of Symbolic Logic* 32, 176-210, 93-103.

Fitch, Frederic B. (1974), Elements of Combinatory Logic, Yale University Press, New Haven, Conn.

Gentzen, G. (1934), Untersuchungen über das logische Schliessen, *Mathematische Zeitschrift* 39, 176-210, 405-431.

Gilmore, P.C. (1971), A Consistent Naive Set Theory: Foundations for a Formal Theory of Computation, IBM Research Report RC 3413, June 22.

Gilmore, P.C. (1972), A Consistent Second Order Naive Set Theory, unpublished report.

Gilmore, P.C. (1974), The Consistency of Partial Set Theory Without Extensionality, ACM *Proc. of Symposia in Pure Mathematics* 13, Part II, 147-153.

Henkin, L. (1950), Completeness in the Theory of Types, *Journal of Symbolic Logic* 15, 81-91.

Henkin, L. (1953), Some Notes on Nominalism, *Journal of Symbolic Logic* 18, 19-29.

Manna, Z. (1974), Mathematical Theory of Computation, McGraw-Hill, New York.

McCarthy, John (1963), A Basis for a Mathematical Theory of Computation, in Computer Programming and Formal Systems (P. Broffort and D. Hirschberg editors), North-Holland, Amsterdam, 33-71.

Myhill, J. (editor) (1969), Intuitionism and Proof Theory, *Proceedings of Summer Conference*. North Holland, Amsterdam.

Prawitz, D. (1967), Completeness and Haupsatz for Second Order Logic, *Theoria* 33, 246-258.

Smullyan, Raymond M. (1968), First-Order Logic, Springer-Verlag, New York Inc.

Stoy, Joseph E. (1977), Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory, The MIT Press, Cambridge, Mass.

Van Orman Quine, Willard (1951), Mathematical Logic, Revised Edition, Harvard University Press, Cambridge.

Van Orman Quine, Willard (1953), From a Logical Point of View, Harvard University Press, Cambridge.

AN EXTENSION OF A SYSTEM OF COMBINATORY LOGIC

FREDERIC B. FITCH

*Department of Philosophy, Yale University,
New Haven, Conn. 06520*

Dedicated to H.B. Curry on the occasion of his 80th Birthday

This paper is an expression of the author's great affection and respect for Professor Curry.

1. INTRODUCTION

1.1. A system CG will be presented which is an extension of the system CΓ of Fitch (1980a) in two main respects: (1) Lambda operators of all finite degrees and of infinite degree are included. (2) Infinitely long well-formed formulas are included. The lambda operators of course require the use of variables. The combinatory operators 'O', 'T', 'σ', and 'B' of CΓ will now be definable by way of lambda operators so that Oab = b, Tab = ba, σabc = ab(bc), and Babc = a(bc), as required.

1.2. The details will be analogous to those of CΓ, so that familiarity with the latter might be helpful to the reader. The proof of the consistency of CG, for example, will closely parallel the proof of the consistency of CΓ.

2. FORMATION RULES

2.1. *Convention.* The metalinguistic name of an expression will consist of the total expression obtained by writing the original expression within single quotes provided that the original expression does not contain lower-case letters (possibly having numerical subscripts viewed as part of them). If the original expression does contain lower-case letters (other than subscripts), the result of placing it within single quotes is not the name of it but denotes an arbitrary expression got by replacing each lower-case letter by an arbitrary well-formed formula (if the letter is from the first part of the alphabet) or by an arbitrary variable (if the letter is from the latter part of the alphabet). See below for the definition of the class of variables and the class of well-formed formulas.

2.2. *Primitive symbols.* (1) Primitive well-formed formulas of degree ≥ 1: '$\sim$' (negation), 'E' (non-emptiness), 'N' (class of natural numbers), '$\urcorner$' (inverse of equality). (2) Primitive well-formed formulas of degree ≥ 2: '=' (equality), '$\vee$' (disjunction). (3) Variables: 'V_1', 'V_2', 'V_3', and so on. (4) The lambda prefix: 'λ'. (4) Parentheses: '(', ')'.

2.3. *Definition of the class of wffs (well-formed formulas).* (1) The primitive wffs are wffs. (2) if 'a' and 'b' are wffs (cp. 2.1) then '(ab)' is a wff. (3) If '$\lambda x_1 x_2 x_3 \ldots$' consists of '$\lambda$' followed by a non-empty finite or infinite sequence of distinct variables 'x_1', 'x_2', 'x_3', ... (and the sequence could contain as few as one or two variables) and if '$(\ldots x_1 \ldots x_2 \ldots x_3 \ldots\ldots)$' is an expression containing some or all of these same variables and such that it becomes a wff if all free occurrences in it of these same variables are replaced by wffs, then '$\lambda x_1 x_2 x_3 \ldots (\ldots x_1 \ldots x_2 \ldots x_3 \ldots\ldots)$' is a wff. (4) If some expression is such that for every left-hand occurrence in it of the form '$((\ldots((ab_1)b_2)\ldots)b_n)$' there is a left-hand occurrence in it of the form '$(((\ldots((ab_1)b_2)\ldots)b_n)c)$' for some 'c', then that expression is a wff.
(A *left-hand occurrence* of a wff in a wff is an occurrence in that wff that has no other occurrence of a wff to the left of it in that wff.) (5) '(ba)' will be called a *reversal* of '(ab)'. A *reversal-transform* of a wff is, by definition, the

result of replacing finitely or infinitely many occurrences of wffs in that wff by occurrences of their reversals. Every reversal-transform of a wff is a wff.

2.4. Notice that 2.3(5) guarantees that other infinite expressions than merely those of 2.3(4) are wffs.

2.5. It will be assumed, as is usual in combinatory logic, that 'abc' abbreviates '((ab)c)' and that omitted parentheses are to be inserted as far to the left as possible. It will also be assumed that '[c a b]' (with spaces on each side of 'a') abbreviates 'abc', that is, '((ab)c)'. An expression of the form 'ab_1b_2...' is to be understood as any wff in which 'a' has a left-hand occurrence. Then 'cb_1b_2...' would be understood as the result of replacing this occurrence by 'c'.

3. THEOREMS AND ANTITHEOREMS.

3.1. Formulas of the form '=ab' (which also may be written as '[b = a]' or simply as 'b = a') are called *equations*.

3.2. Given any class of equations $\underset{\sim}{A}$, there exists a system $C\underset{\sim}{A}$ defined relatively to the class $\underset{\sim}{A}$ in a way described below. The class of *theorems* of the system $C\underset{\sim}{A}$ will be designated as $P\underset{\sim}{A}$ and is defined by transfinite induction in 3.4.1 - 3.4.7 below. This is done in such a way that $\underset{\sim}{A}$ turns out to be the class of those equations which are theorems of system $C\underset{\sim}{A}$, so that $\underset{\sim}{A} \subseteq P\underset{\sim}{A}$.

3.3. Every system $C\underset{\sim}{A}$ not only is characterized by a class $P\underset{\sim}{A}$ of theorems, but also by a class $R\underset{\sim}{A}$ of what will be called *antitheorems*. All equations not in $\underset{\sim}{A}$ are antitheorems of $C\underset{\sim}{A}$ by 3.4.1 below and are the only equations that are antitheorems of $C\underset{\sim}{A}$. The class $R\underset{\sim}{A}$ is defined simultaneously with $P\underset{\sim}{A}$ in 3.4.1 - 3.4.7.

3.4. The rules defining $P\underset{\sim}{A}$ and $R\underset{\sim}{A}$ in terms of $\underset{\sim}{A}$ are now presented. The phrase "if and only if" is abbreviated as "iff". The parenthetical parts of the rules give conditions for membership in $R\underset{\sim}{A}$. The expressions appearing in these rules can be understood as finitely long or as infinitely long wffs in accordance with the last two sentences of 2.5.

3.4.1. '$=abc_1c_2\ldots$', which also may be written as '$[b = a]c_1c_2\ldots$', is in $P\underset{\sim}{A}$ (in $R\underset{\sim}{A}$) iff '$=ab$' is in $\underset{\sim}{A}$ (not in $\underset{\sim}{A}$).

3.4.2. '$Nab_1b_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff at least one of (each of) '$=0ab_1b_2\ldots$', '$=(\sigma 0)ab_1b_2\ldots$', '$=(\sigma(\sigma 0))ab_1b_2\ldots$', ..., is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$), where 'σ' is an abbreviation for '$\lambda xyz(xy(yz))$'.

3.4.3. '$\sim ab_1b_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff '$ab_1b_2\ldots$' is in $R\underset{\sim}{A}$ ($P\underset{\sim}{A}$).

3.4.4. '$\dot{V}abc_1c_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff at least one of (each of) '$ac_1c_2\ldots$' and '$bc_1c_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$).

3.4.5. '$Eab_1b_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff, for some (for every) wff 'c', '$acb_1b_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$).

3.4.6. '$\lambda x_1x_2\ldots(\ldots x_1\ldots x_2\ldots\ldots)a_1a_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff '$(\ldots a_1\ldots a_2\ldots\ldots)$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$), where '$\lambda x_1x_2\ldots(\ldots x_1\ldots x_2\ldots\ldots)$' is like '$\lambda x_1x_2x_3\ldots(\ldots x_1\ldots x_2\ldots x_3\ldots\ldots)$' of 2.3(3) and where '$(\ldots a_1\ldots a_2\ldots\ldots)$' is a wff resulting from replacing all free occurrences of 'x_1', 'x_2', ... in '$(\ldots x_1\ldots x_2\ldots\ldots)$' everywhere respectively by 'a_1', 'a_2', ...

3.4.7. If '$=a(=b)$', that is, '$[(=b) = a]$', is in $\underset{\sim}{A}$, then 'ɿ$ac_1c_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff '$bc_1c_2\ldots$' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$).

3.5. Although the rules in 3.4 are stated as giving sufficient *and* necessary conditions for various wffs to be in $P\underset{\sim}{A}$ or in $R\underset{\sim}{A}$, these rules could have been stated equally well as giving merely sufficient conditions (by using "if" instead of "iff"). The necessity of the conditions would then follow from the fact that there is at most one rule giving the condition for a given wff to be in $P\underset{\sim}{A}$ or $R\underset{\sim}{A}$.

4. DEFINITION OF G

4.1. The aim is now to define a class G of equations that contains every equation '$=ab$' (that is, '$b = a$') such that, for every class of equations $\underset{\sim}{X}$ which has G as a subclass, every wff that has a left-hand occurrence of 'a' is in $P\underset{\sim}{X}$ iff (1) the result of replacing that occurrence by an occurrence of 'b' is also in $\underset{\sim}{PX}$ and (2) similarly for $\underset{\sim}{RX}$. (For examples of equations in G, see the examples of equations in Γ presented in 3.7 of Fitch (1980a).) Matters will be so arranged that unrestricted substitution of equals for equals will be allowed in the system CG. Results and terminology will next be presented that are required for defining the class G and the system CG.

4.2. An '=ab'-*transform* of a wff 'c' is by definition a result of replacing any number of occurrences of 'a' by 'b' and (possibly simultaneously) any number of occurrences of 'b' by 'a' in 'c'. (In 4.2 of Fitch (1980a) the replacement was of a single occurrence of 'a' or 'b' by the other, but here provision must be made for infinitely long wffs and therefore for simultaneous replacements of infinitely many occurrences.)

4.3. A *left-hand* '=ab'-transform of a wff 'c' is by definition an '=ab'-transform of 'c' such that there is only one replaced occurrence of 'a' or 'b' in 'c', and this occurrence is a left-hand occurrence in 'c' in the sense defined in 2.3(4).

4.4. A *right-hand* '=ab'-transform of a wff 'c' is by definition an '=ab'-transform of 'c' that does not involve replacement of a left-hand occurrence in 'c'.

4.5. A class $\underset{\sim}{A}$ of wffs is by definition *closed for* '=ab' if every '=ab'-transform of a member of $\underset{\sim}{A}$ is a member of $\underset{\sim}{A}$, in other words, if 'a' and 'b' are intersubstitutable as far as membership in $\underset{\sim}{A}$ is concerned.

4.6. A class $\underset{\sim}{A}$ of wffs is by definition *left-hand closed for* '=ab' if every left-hand '=ab'-transform of a member of $\underset{\sim}{A}$ is a member of $\underset{\sim}{A}$. Thus if a class $\underset{\sim}{A}$ is left-hand closed for '=ab' it must be the case that '$ac_1c_2\ldots$' is in $\underset{\sim}{A}$ iff '$bc_1c_2\ldots$' is in $\underset{\sim}{A}$.

4.7. A class $\underset{\sim}{A}$ of wffs is by definition *right-hand closed for* '=ab' if every right-hand '=ab'-transform of a member of $\underset{\sim}{A}$ is a member of $\underset{\sim}{A}$. If $\underset{\sim}{A}$ is both left-hand and right-hand closed for '=ab', then, clearly, $\underset{\sim}{A}$ is closed for '=ab'.

4.8. Note that every right-hand '=ab'-transform of an equation is itself an equation, so that the class of equations is right-hand closed for its own members.

4.9. A class $\underset{\sim}{A}$ of wffs is by definition *closed for a class* $\underset{\sim}{B}$ of equations if it is closed for every member of $\underset{\sim}{B}$. Also, a class of equations is said to be *self-closed* if it is closed for every one of its own members. Similarly for left-hand closure, right-hand closure, and strong closure (defined in 4.11 below).

4.10. A class $\underset{\sim}{A}$ of equations is by definition *systematically closed* if $P\underset{\sim}{A}$ and $R\underset{\sim}{A}$ are closed for $\underset{\sim}{A}$. This means, in effect, that substitution of equals for equals is permitted in system $C\underset{\sim}{A}$. If a class of equations is systematically closed, it is clearly self-closed by 3.4.1 and therefore right-hand self-closed.

4.11. A class $\underset{\sim}{A}$ of equations is by definition *strongly closed for* '=ab' if the following two conditions as satisfied:

(1) $\underset{\sim}{A}$ is systematically closed.

(2) Every class $\underset{\sim}{X}$ of equations which is right-hand self-closed (4.7, 4.9) and such that $\underset{\sim}{A} \subseteq \underset{\sim}{X}$, is such that $P\underset{\sim}{X}$ and $R\underset{\sim}{X}$ are left-hand closed for '=ab' (4.6).

4.12. The class $\underset{\sim}{G}_0$ is by definition the class of equations of the form '=aa' for all wffs 'a'. It is clear that $\underset{\sim}{G}_0$ is systematically closed and strongly self-closed (4.9 - 4.11).

4.13. If a class $\underset{\sim}{A}$ of equations is right-hand self-closed and strongly closed for '=ab', then the '=ab'-*closure* of $\underset{\sim}{A}$ is by definition the result of adding '=ab' to $\underset{\sim}{A}$ and enough further equations so that the resulting class is right-hand self-closed (4.7, 4.9). If '=ab' is already in $\underset{\sim}{A}$, where $\underset{\sim}{A}$ is right-hand self-closed, then the '=ab'-closure of $\underset{\sim}{A}$ is of course $\underset{\sim}{A}$ itself.

4.14. Let $\underset{\sim}{A}$ be any class of equations which is strongly self-closed and strongly closed for '=ab', and which does not contain '=ab'. Then the '=ab'-closure of $\underset{\sim}{A}$ is strongly self-closed. The proof is given in the outline below. A similar proof will be found in 4.15 of Fitch (1980a). Let $\underset{\sim}{B}$ be the '=ab'-closure of A. Note that since $\underset{\sim}{A}$ is systematically closed, it is right-hand self-closed (4.10). The steps are as follows:

(1) $\underset{\sim}{A} \subseteq \underset{\sim}{B}$, by the definition of $\underset{\sim}{B}$.

(2) Let $\underset{\sim}{X}$ be an arbitrary class of equations which is right-hand self-closed and such that $\underset{\sim}{B} \subseteq \underset{\sim}{X}$. (See 4.11(2). In (7) below it will be shown that $P\underset{\sim}{X}$ and $R\underset{\sim}{X}$ are left-hand closed for $\underset{\sim}{B}$, so that $\underset{\sim}{B}$ can then be shown in (10) to be strongly self-closed, which is the desired result.)

(3) $P\underset{\sim}{X}$ and $R\underset{\sim}{X}$ are left-hand closed for '=ab' and for $\underset{\sim}{A}$ by (1), (2), and 4.11(2).

(4) $P\underset{\sim}{X}$ and $R\underset{\sim}{X}$ are closed for '=ab' and for $\underset{\sim}{A}$. This can be shown, first for '=ab' and then similarly for $\underset{\sim}{A}$. The steps are as follows:

(4a) Every right-hand '=ab'-transform of an equation in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$) is in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$), by (2) and 3.4.1.

(4b) Every left-hand '=ab'-transform of an equation in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$) is in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$), by (3), so that every '=ab'-transform of an equation in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$) is in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$). Similarly for expressions of the form '$=cde_1e_2..$' which act like equations by 3.4.1 and may be called *semi-equations*.

(4c) Starting with semi-equations and proceeding to other members of $P\underset{\sim}{X}$ and $R\underset{\sim}{X}$ by the transfinite induction used in defining $P\underset{\sim}{X}$ and $R\underset{\sim}{X}$, it can be shown that if all '=ab'-transforms of wffs so far assigned, at a given stage, to $P\underset{\sim}{X}$ or $R\underset{\sim}{X}$ are in $P\underset{\sim}{X}$ or $R\underset{\sim}{X}$ respectively, then the right-hand '=ab'-transforms of the next wff to be so assigned must themselves be in $P\underset{\sim}{X}$ or $R\underset{\sim}{X}$ respectively, as can be seen from the form of the rules 3.4.2 - 3.4.7 in terms of which the assignment is made. (Rule 3.4.7 gives slightly more difficulty than the others. See 4.15(4c) of Fitch (1980a)). Furthermore, because of (3), the left-hand '=ab'-transform of any wff in $P\underset{\sim}{X}$ or $R\underset{\sim}{X}$ is in $P\underset{\sim}{X}$ or $R\underset{\sim}{X}$ respectively. Hence, all the '=ab'-transforms of the next wff assigned to $P\underset{\sim}{X}$

or $R\mathbf{X}$ are in $P\mathbf{X}$ or $R\mathbf{X}$ respectively.

(4d) The argument used in (4c) to show that $P\mathbf{X}$ and $R\mathbf{X}$ are closed for '=ab' can also be used to show that $P\mathbf{X}$ and $R\mathbf{X}$ are closed for each equation in $\mathbf{A}$. This is because $\mathbf{X}$ is right-hand closed for its own members by (2) and hence for its subclass $\mathbf{A}$ (just as $\mathbf{X}$ is right-hand closed for '=ab'), and because $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed for $\mathbf{A}$ by (3) (just as $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed for '=ab'). Thus $P\mathbf{X}$ and $R\mathbf{X}$ are closed for $\mathbf{A}$.

(5) Consider any equation '=ef' which is a right-hand '=ab'-transform of an equation '=cd' for which $\mathbf{X}$ is right-hand closed. Then $\mathbf{X}$ must be right-hand closed for '=ef'. This is because $\mathbf{X}$ is right-hand closed for '=ab', as already shown, and because the effect of substitutions of 'e' and 'f' for each other in right-hand occurrences can be gotten by substitutions of 'a' and 'b' for each other and of 'c' and 'd' for each other in right-hand occurrences.

(6) If we take the class $\mathbf{A}$ and repeatedly extend it by adding '=ab'-transforms of wffs already in $\mathbf{A}$ or in the partial extensions of it, then since X is right-hand closed for $\mathbf{A}$ and for '=ab', $\mathbf{X}$ will also be right-hand closed, by (5), for each of these extensions of $\mathbf{A}$, and finally for the union of these extensions, which is easily seen to be $\mathbf{B}$ itself. Thus $\mathbf{X}$ is right-hand closed for $\mathbf{B}$.

(7) Using the description of $\mathbf{B}$ given in (6), it can be shown that $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed for $\mathbf{B}$. This can be seen observing (a) that $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed for '=ab' and $\mathbf{A}$ by (3), and (b) that if $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed for any equation, then they are left-hand closed for any right-hand '=ab'-transformation of that equation. More specifically, in the case of (b), let '=cd' be any equation for which $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed, and let '=ef' be a right-hand '=ab'-transform of '=cd'. It is required to show that $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed '=ef'. From the fact that $P\mathbf{X}$ and $R\mathbf{X}$ are left-hand closed

for '=cd', it follows that if '$cg_1g_2\ldots$' is any wff in which 'c' has a left-hand occurrence and if '$dg_1g_2\ldots$' is a left-hand '=cd'-transform of '$cg_1g_2\ldots$', then:

'$cg_1g_2\ldots$' is in P$\underset{\sim}{X}$ (R$\underset{\sim}{X}$) iff '$dg_1g_2\ldots$' is in P$\underset{\sim}{X}$ (R$\underset{\sim}{X}$); and hence also, since P$\underset{\sim}{X}$ (R$\underset{\sim}{X}$) is closed for '=ab' by (4c):

'$eg_1g_2\ldots$' is in P$\underset{\sim}{X}$ (R$\underset{\sim}{X}$) iff '$fg_1g_2\ldots$' is in P$\underset{\sim}{X}$ (R$\underset{\sim}{X}$), where '$eg_1g_2\ldots$' results from replacing the left-hand occurrence of 'c' by 'e' in '$cg_1g_2\ldots$', and '$fg_1g_2\ldots$' results from replacing the left-hand occurrence of 'd' in '$dg_1g_2\ldots$' by 'f'. Thus P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ are left-hand closed for '=ef'.

(8) The same sort of argument that showed in (4) that P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ are closed for '=ab' and for $\underset{\sim}{A}$ can now be used to show that P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ are closed for $\underset{\sim}{B}$. This is because $\underset{\sim}{X}$ is right-hand closed for $\underset{\sim}{B}$ by (6) and because P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ are left-hand closed for $\underset{\sim}{B}$ by (7).

(9) Since $\underset{\sim}{X}$ is an arbitrary class of equations which is right-hand self-closed by (2) and has $\underset{\sim}{B}$ as a subclass, and since P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ are closed for $\underset{\sim}{B}$ by (8), and since $\underset{\sim}{B}$ is right-hand self-closed by 4.13, it follows that P$\underset{\sim}{B}$ and R$\underset{\sim}{B}$ must be like P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ in being closed for $\underset{\sim}{B}$, that is, it follows that P$\underset{\sim}{B}$ and R$\underset{\sim}{B}$ must be closed for $\underset{\sim}{B}$, so that $\underset{\sim}{B}$ is systematically closed (4.10).

(10) Since $\underset{\sim}{X}$ is an arbitrary class of equations which is right-hand self-closed and has $\underset{\sim}{B}$ as a subclass, and since P$\underset{\sim}{X}$ and R$\underset{\sim}{X}$ are left-hand closed for $\underset{\sim}{B}$ by (7), and $\underset{\sim}{B}$ is systematically closed by (9), it follows by 4.11 that $\underset{\sim}{B}$ is strongly self-closed, as was to be shown.

4.15. Consider a series α of classes of equations such that each member of the series is strongly self-closed and such that of any two members of the series one is a subclass of the other. Then the union $U\alpha$ of the classes in the series is strongly self-closed. This is easy to show. See 4.16 of Fitch (1980a).

4.16. If $\underset{\sim}{A}$ is strongly self-closed and if $\underset{\sim}{B}$ is the result of taking all equations not in $\underset{\sim}{A}$ for which $\underset{\sim}{A}$ is strongly closed, and adding these equations to $\underset{\sim}{A}$ and just enough further equations so that the resulting class is right-hand self-closed, then $\underset{\sim}{B}$ is strongly self-closed. This is because $\underset{\sim}{B}$ can be regarded as obtained from $\underset{\sim}{A}$ by forming successively larger classes in such a way that each (after the first) is the '=ab'-closure (4.13) of the previous class for some equation '=ab' for which $\underset{\sim}{A}$ is strongly closed. These successively larger classes can all be shown to be strongly closed for the same equations for which $\underset{\sim}{A}$ is strongly closed, and each of them is strongly self-closed by 4.14. Hence $\underset{\sim}{B}$, the union of them, is strongly self-closed by 4.15.

4.17. A series γ of classes of equations is next constructed by starting with the class G_0 described in 4.12 as having for members only equations of the form '=aa', and proceeding as follows: Given any member $\underset{\sim}{A}$ of γ, the next member of the series γ is by definition the class $\underset{\sim}{B}$ defined in terms of $\underset{\sim}{A}$ just as $\underset{\sim}{B}$ in 4.16 is defined in terms of $\underset{\sim}{A}$ there. Also, given any infinite series of classes thus obtained, the union of these classes is constructed and added to the series. This process is continued as far as possible and gives, by definition, the series γ. Note that of any two members of γ, one is a subclass of the other and that every member of γ is strongly self-closed by 4.12, 4.16, and 4.15. The last member of γ is the union $\cup\gamma$ of all the members of γ. This union is by definition the class G. There cannot be any next member of the series after G since this would mean that the series γ had not been completed after all. Thus there cannot be an '=ab'-closure of G for any equation '=ab' not in G, and G cannot be strongly closed for any equation not a member of it.

4.18. G is strongly self-closed by 4.12, 4.16, and 4.15, and is therefore systematically closed by 4.11(1), so that substitution of equals for equals is allowed in system CG.

4.19. G is strongly closed for its own members by 4.16, and only for its own members by 4.17, so that a necessary and sufficient condition for an equation '=ab' to be in G (and hence to be a theorem of the system CG) is that G be strongly closed for '=ab'.

4.20. This criterion for membership of '=ab' in G can also be expressed as follows in virtue of 4.11 and the fact that, by 4.18, G is systematically closed: For every class $\underset{\sim}{X}$ of equations which is right-hand self-closed and such that $G \subseteq \underset{\sim}{X}$, and for every wff 'e' in which 'a' has a left-hand occurrence and every wff 'f' which is a left-hand '=ab'-transform of 'e', (1) and (2) as follows both hold:

(1) 'e' is in $P\underset{\sim}{X}$ iff 'f' is in $P\underset{\sim}{X}$.

(2) 'e' is in $R\underset{\sim}{X}$ iff 'f is in $R\underset{\sim}{X}$.

(The notation '$ac_1c_2\ldots$' could have been used for 'e' and the notation '$bc_1c_2\ldots$' for 'f'.)

4.21. Here is an example of an application of the criterion for membership in G given in 4.20: Since '=(=b)(=b)' is in $\underset{\sim}{X}$, where $\underset{\sim}{X}$ is as in 4.20, we have, by 3.4.7: '$\urcorner(=b)c_1c_2\ldots$'is in $P\underset{\sim}{X}$ ($R\underset{\sim}{X}$) iff '$bc_1c_2\ldots$' is. Therefore, by 4.20, '$=(\urcorner(=b))b$' is in G, and hence is a theorem of CG by 3.4.1. This shows that '$\urcorner$' acts as the inverse of equality. Other examples of members of G are the members of Γ presented in 3.7 of Fitch (1980a).

4.22. An example of an equation that can be shown not to be in G (and not to be in Γ) is the equation '0 = [a + 1]' where 'a' is any wff. If the latter equation were in G then it could be shown that

$$
\begin{aligned}
{\sim}[0 = 0] &= 00{\sim}[0 = 0] \\
&= [a + 1]0{\sim}[0 = 0] \\
&= {+}1a0{\sim}[0 = 0] \\
&= 10(a0{\sim})[0 = 0] \\
&= 0(a0{\sim})[0 = 0] \\
&= [0 = 0].
\end{aligned}
$$

Then '~[0 = 0] = [0 = 0]' would be in G, and '~[0 = 0]' and '[0 = 0]' would both be in PG because of the fact that G is systematically closed. But this is impossible because CG is consistent, as will be shown below. (For the properties of '0', '+', and '1' used above, see Fitch (1980a).)

5. SATURATED AND UNSATURATED EXPRESSIONS

5.1. Some further rules will be introduced to guarantee that if '=(ac)(bc)' is in G for every wff 'c', then '=ab' is in G.

5.2. Note that by 2.2, '~', 'E', 'N' and '∩' have been assigned all degrees ≥1, and '=' and 'v' have been assigned all degrees ≥2. Among assigned degrees are all non-negative, finite integral degrees, together with the infinite degree ω. Wffs of the form '$\lambda x_1 x_2 \ldots x_n(\ldots x_1 \ldots x_2 \ldots\ldots x_n \ldots)$', for finite n, have all degrees ≥n. Wffs of the form, '$\lambda x_1 x_2 \ldots(\ldots x_1 \ldots x_2 \ldots\ldots)$', where there are infinitely many distinct variables in the prefix, have degree ω. By a *saturated* wff is meant a finite wff '$ab_1 b_2 \ldots b_n$', where 'a' is of all degrees ≥n, or an infinitely long wff '$ab_1 b_2 \ldots$' (see 2.3) where 'a' is of degree ω (and the sequence 'b_1', 'b_2', ... , is infinitely long). *Unsaturated* wffs are those wffs which are not saturated. For example, '=a' is unsaturated, while '=ab' and '=abc' are saturated.

5.3. Conditions have not been given for membership of unsaturated wffs in $P\underset{\sim}{A}$ or $R\underset{\sim}{A}$. That deficiency will now be remedied by introducing the following rule for unsaturated wffs: If 'a' is unsaturated and if enough (finitely or infinitely many) occurrences of wffs are sequentially added to the right side of 'a' so that the resulting wff is saturated, then 'a' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) iff the resulting wff is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$) for all wffs that are or could be thus added to produce such a saturated wff. (For example, if 'W' is so defined as to have its usual properties, and if '=aa' is in $\underset{\sim}{A}$ for every 'a', then the unsaturated wff 'W=' is in $P\underset{\sim}{A}$ because the saturated wff 'W=a' is in $P\underset{\sim}{A}$ for every 'a'.)

5.4. The above rule for unsaturated wffs will be assumed to be added to 3.4.1 - 3.4.7. All results previously obtained are still obtainable after this rule has been added.

5.5. It is easy to show, using 5.3, that if 'a' is unsaturated then 'a' is in PA (RA) iff, for every wff 'b', 'ab' is in PA (RA). Proof omitted.

5.6. Also in the case of a saturated wff 'a' it can be shown that 'a' is in PA (RA) iff, for every wff 'b', 'ab' is in PA (RA). If 'a' is an equation or semi-equation (4.14(4b)), this is obviously the case by 3.4.1. For other saturated wffs it can be shown to be the case by an induction that parallels the rules 3.4.2 - 3.4.7.

5.7. From 5.5 and 5.6 it follows that, for every wff 'a', 'a' is in PA (RA) iff, for every wff 'b', 'ab' is in PA (RA).

5.8. Of the two statements (1) and (2) below, it can be shown that (2) follows from (1):

(1) For every wff 'c' and every wff 'd', if 'ac' has a left-hand occurrence in 'd', then 'd' is in PA (RA) iff the left-hand '=(ac)(bc)'-transform of 'd' in which 'bc' has replaced the left-hand occurrence of 'ac' is in PA (RA).

(2) For every wff 'd', if 'a' has a left-hand occurrence in 'd', then 'd' is in PA (RA) iff the left-hand '=ab'-transform of 'd' in which 'b' has replaced the left-hand occurrence of 'a' is in PA (RA).

5.9. To show that (2) above follows from (1), observe that (2) is merely a special case of (1) except for the situation in (2) where 'd' is 'a' itself. Thus to demonstrate the required result it suffices to prove that 'a' is in PA (RA) iff 'b' is in PA (RA), and this can be seen to be true since the following four conditions are equivalent by 5.7 and 5.8(1):

(a) 'a' is in PA (RA).

(b) 'ae' is in PA (RA) for every 'e'.

(c) 'be' is in PA (RA) for every 'e'.

(d) 'b' is in $P\underset{\sim}{A}$ ($R\underset{\sim}{A}$).

5.10. If, for every 'c', 'ac = bc' is in G, then 'a = b' is in G, by 4.20 and 5.8.

5.11. The *consistency* of the system CG will now be proved by showing that there is no wff in both PG and RG, so that there is no wff 'p' such that 'p' and '~p' are both in PG (3.4.3). The same proof applies to any system $C\underset{\sim}{A}$ such that $\underset{\sim}{A}$ is systematically closed. (The same proof of consistency is given for CΓ in 5.9 of Fitch (1980a)).

(1) In the order of showing wffs to be in PG, it can be assumed without loss of generality that the first wffs shown to be in PG have been so shown by use of 3.4.1. Every such wff fails to be in RG, as is easily seen from 3.4.1.

(2) Suppose that up to a given stage in showing wffs to be in PG all wffs so far shown to be in PG fail to be in RG (and at the outset they do fail as (1) indicates). Then it can be seen as follows that any further wff shown to be in PG also fails to be in RG, so that no wff in PG is also in RG. Suppose, for example, that 'vabcd' is such a further wff, and so must have been shown to be in PG by rule 3.4.4, assuming the rules to be in "if" form (3.5). Suppose also that 'vabcd' is in RG (as will be shown to be impossible). Since 'vabcd' is in PG by 3.4.4, it must be the case that either 'acd' or 'bcd' has already been shown to be in PG, so that by the hypothesis of the transfinite induction either 'acd' is not in RG or 'bcd' is not in RG. But since 'vabcd' has been assumed to be in RG, it follows by 3.4.4 that both 'acd' and 'bcd' are in RG. Hence, as was to be shown, 'vabcd' cannot be in RG. Cases involving other rules among 3.4.2 - 3.4.6, or involving rule 5.3 for unsaturated wffs, are similar to the above case. When the rule involved is 3.4.7, the situation is slightly more complicated as in the following example. Suppose that 'Ꞁacd' is a further wff shown to be in PG and that, as before, all wffs previously shown to be in PG fail

to be in RG. It is to be shown that '⌐acd' also fails to be in RG. By rule 3.4.7, there must be a 'b' such that '=a(=b)' is in G and such that 'bcd' is in PG. Now if '⌐acd' is in RG (as will be proved impossible), then by 3.4.7 there must be an 'e' such that '=a(=e)' is in G and such that 'ecd' is in RG. But then '=(=b)(=e)' is in G, and from this and the fact that '=(⌐(=b))b' and '=(⌐(=e))e' are in G by 4.21, it follows that '=be' is in G and PG, so that, since G is systematically closed and since 'ecd' is in RG, 'bcd' is in RG. Then, however, 'bcd' is a wff shown to be in PG previously to '⌐acd', but such that it is in RG, contrary to assumption.

6. REMARKS ON DENUMERABILITY

6.1. By allowing infinitely long wffs as in system CG, the assignment of Gödel numbers to wffs apparently becomes impossible, and the class of wffs becomes non-denumerable. One advantage of this is that it is no longer possible to use a diagonal procedure to define an infinite sequence of natural numbers that is not nameable by a wff of the system.

6.2. Suppose that 'b_1', 'b_2', 'b_3', ... is any infinite sequence of wffs. Form the infinitely long wff '$Fb_1b_2b_3\ldots$' by attaching 'b_1', 'b_2', 'b_3', ... sequentially to the right side of 'F', where 'F' is a wff so defined that the equation

$$(Fb_1b_2b_3\ldots)n = b_n$$

is in G for every finite n, as can easily be done. Thus any infinite sequence of wffs may be represented by an infinitely long wff '$Fb_1b_2b_3\ldots$' that acts like the name of a function which for argument n has as its value the thing denoted by the nth term of such a sequence.

ACKNOWLEDGEMENT

I am indebted to Rulon Wells for pointing out to me the difficulties of maintaining a position according to which all infinities are denumerable. He convinced me of the untenability of such a position regarding denumerability because of the incompleteness that results with respect to representing infinite sequences, as shown by the diagonal argument. It is hoped that the position of the present paper overcomes such difficulties about denumerability.

REFERENCES

Fitch, F.B. (1980a) A Consistent Combinatory Logic with an Inverse to Equality, forthcoming in *The Journal of Symbolic Logic*.

Fitch, F.B. (1980b) The Consistency of System Q, forthcoming in *The Journal of Symbolic Logic*.

SEQUENTIAL SYSTEMS OF λ-CONVERSION AND OF COMBINATORY LOGIC

Alexander S. Kuzichev

Department of Mathematical Logic,
Moscow State University, Moscow 117234, USSR

Dedicated to H.B. Curry on the occasion of his 80th Birthday

We consider the question of introducing logical, arithmetical and set theoretical notions into λ-conversion and pure combinatory logic (Schönfinkel 1924; Church 1941; Curry, Feys and Craig 1958; Curry, Hindley and Seldin 1972; Hindley, Lercher and Seldin 1972; Stenlund 1972). Though this problem came up long ago, it has not yet received a satisfactory solution. As is well known, a straightforward extension of a combinatorially complete theory with the aid of the implication operator P having usual intuitionistic properties leads to the Curry paradox.

To solve this problem we have proposed type-free extensions of λ-conversion calculus or pure combinatory logic based on a Gentzen style theory of sequents with a restriction on the cut rule (but unrestricted principle of combinatorial completeness) (for example, see the Á-system in Kuzichev (1977a,b; 1978a,b). The introduced systems have two levels. The basis of each system is the calculus of combinatorially complete theory with sequents of the form $a \to b$, where a and b are obs, and $\to$ is the first level sequent symbol. The second level is constructed as an extension of the first having sequents of the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are sequences (possibly empty) of obs, and $\Rightarrow$ is the second level sequent symbol. Each system has rule of ascent

$$* : \frac{a \to b}{a \Rightarrow b} .$$

The axiom scheme of the Á-system is

$$Q^{\circ} : \Rightarrow Qaa ,$$

where Q is the equality operator. Each system has λ-cut rules (combinatory rules)

$$\lambda* : \frac{\Gamma \Rightarrow \Theta,a; \quad a \to b}{\Gamma \Rightarrow \Theta,b} \text{ and } *\lambda : \frac{a,\Gamma \Rightarrow \Theta; \quad b \to a}{b,\Gamma \Rightarrow \Theta} ;$$

the cut rule:

$$\frac{\Delta \Rightarrow \Lambda, a; \quad a, \Gamma \Rightarrow \Theta}{\Delta, \Gamma \Rightarrow \Lambda, \Theta}$$

is absent (here Γ, Θ, Δ and Λ are sequences of obs, a and b are obs). The unrestricted principle of combinatorial completeness makes it possible to replace in a provable second level sequent any ob a by b, if corresponding first level sequents $a \rightarrow b$ or $b \rightarrow a$ are provable.

Using the Church-Rosser theorem we can prove that each system is absolutely consistent: there exists a sequent $\Gamma \Rightarrow \Theta$ that is not provable in the system. All operators of predicate logic (Π, $\exists$, P, &, v, $\neg$, $\equiv$) and the equality operator Q can be defined in terms of formal implication Ξ or functionality F (Kuzichev 1973a,b; 1974a,b). Since in pure predicate calculus the rule of cut is eliminable, it follows that every provable sequent of that calculus holds in these systems for arbitrary obs.

The properties of propositional connectives, quantifiers and the equality in connection with the principle of combinatorial completeness permit us:

(i) to construct an ob a such that the sequents $\Rightarrow a$ and $\Rightarrow \neg a$ are provable;

(ii) to prove the inadmissibility of the cut rule and modus ponens for the second level;

(iii) to prove sequents $\Rightarrow \neg(Qab)$ (for arbitrary obs a and b) and $\Pi\lambda x.Nx \Rightarrow$, where N is the arithmetical operator,

$$N = \lambda x.\Xi(\lambda y.\&(yZ_0)(\Pi\lambda z.P(yz)(y(\sigma z))))\lambda t.tx,$$
$$Z_0 = KI,$$
$$\sigma = \lambda xyz.y(xyz).$$

To get over these difficulties which are due to the principle of the combinatorial completeness, we introduce additional restrictions on the obs of the second level (for example, see definitions of the N-derivation in the $\mathcal{A}$-system and $\mathcal{R}$-theory in Kuzichev (1976, 1977c)).

The aim of the present work is to show that the apparatus of the proposed systems can be used for the presentation of axiomatic theories and to prove their consistency. In this paper the apparatus of the $\mathcal{A}$-system is used to prove the consistency of formal arithmetic (the calculi H$\mathcal{A}$ and L$\mathcal{A}$ of formal arithmetic in the language of the $\mathcal{A}$-system were formulated in Kuzichev (1977b, 1978a)). The paper has three parts:

I. The proof of the consistency of formal arithmetic;

II. Embedding theorem;

III. A variant of formalization of logical and mathematical theories.

The Church-Rosser theorem is used freely in the following.

By virtue of Gödel's Second Incompleteness Theorem, every proof of the consistency of formal arithmetic must contain methods which are not formalizable within the framework of the system whose consistency is being investigated. Thus, Gentzen's proof is carried out by means of transfinite induction, and in

Gödel's proof functionals of higher types are used. In the proof we offer, we make use of the type-free A-system with a restriction on the cut rule (which uses essentially the notion of convertibility and its negation) and the unrestricted principle of combinatorial completeness. In the class of all derivations of the A-system we define a certain subclass $\mathfrak{M}$ of derivations (called F-derivations), which is closed with respect to the cut rule and modus ponens and is free from contradictions of the form (i) or (iii). From a theorem on the embedding of formal arithmetic in the class $\mathfrak{M}$ and a lemma on the nonderivability of the sequent $\Pi\lambda x.Nx \Rightarrow$ in $\mathfrak{M}$ which corresponds to the 'empty' sequent $\Rightarrow$ of calculus LA follows a theorem on the consistency of formal arithmetic. By virtue of the definition of F-derivations the class of all obs is divided into subclasses. In $\mathfrak{M}$ we have the unrestricted principle of combinatorial completeness.

I. THE PROOF OF THE CONSISTENCY OF FORMAL ARITHMETIC

I.1. Definitions and notations.

The result of substituting the ob c in the ob a in place of the variable x is denoted by $[c/x]a$. The expression '$x \notin a_1, \ldots, a_n$' means that x has no free occurrences in $a_1, \ldots, a_n$. Variables are denoted by the letters x, y, z, t. Obs are denoted by the Latin latters a, ..., h; Γ, Δ, Θ, Λ are sequences of obs.

I.2. Postulates of the A-system of λ-conversion.

(α): $\lambda xa \to \lambda y[y/x]a$, $y \notin a$; (β): $(\lambda xa)b \to [b/x]a$;

$$(\mu): \frac{a \to b}{ca \to cb}; \quad (\nu): \frac{a \to b}{ac \to bc}; \quad (\xi): \frac{a \to b}{\lambda xa \to \lambda xb};$$

$$(\tau): \frac{a \to b; \quad b \to c}{a \to c}; \quad (\sigma): \frac{a \to b}{b \to a}.$$

(α), (β), (μ), (ν), (ξ), (τ) and (σ) are postulates of λ-conversion.

Postulates of A-extension:

1. $\Rightarrow Qaa$;

2. $\dfrac{a \to b}{a \Rightarrow b}$;

3. $\dfrac{\Gamma \Rightarrow \Theta, x; \quad x \notin \Gamma, \Theta}{\Gamma \Rightarrow \Theta}$;

4. $\dfrac{\Gamma \Rightarrow \Theta}{\Gamma \Rightarrow \Theta, a}$;

5. $\dfrac{\Gamma \Rightarrow \Theta}{a, \Gamma \Rightarrow \Theta}$;

6. $\dfrac{\Gamma \Rightarrow \Theta, a, a}{\Gamma \Rightarrow \Theta, a}$;

7. $\dfrac{a, a, \Gamma \Rightarrow \Theta}{a, \Gamma \Rightarrow \Theta}$;

8. $\dfrac{\Gamma \Rightarrow \Lambda, a, b, \Theta}{\Gamma \Rightarrow \Lambda, b, a, \Theta}$;

9. $\dfrac{\Delta, a, b, \Gamma \Rightarrow \Theta}{\Delta, b, a, \Gamma \Rightarrow \Theta}$;

10. $\dfrac{\Gamma \Rightarrow \Theta, a; \quad a \to b}{\Gamma \Rightarrow \Theta, b}$;

11. $\dfrac{a, \Gamma \Rightarrow \Theta; \quad b \to a}{b, \Gamma \Rightarrow \Theta}$;

12. $\dfrac{\Delta \Rightarrow \Lambda, ca; \quad cb, \Gamma \Rightarrow \Theta}{Qab, \Delta, \Gamma \Rightarrow \Lambda, \Theta}$;

13. $\dfrac{ax,\Gamma \Rightarrow \Theta,bx; \quad x \notin a,b,\Gamma,\Theta}{\Gamma \Rightarrow \Theta,\Xi ab}$; 14. $\dfrac{\Delta \Rightarrow \Lambda,ac; \quad bc,\Gamma \Rightarrow \Theta}{\Xi ab,\Delta,\Gamma \Rightarrow \Lambda,\Theta}$;

15. $\dfrac{\Gamma \Rightarrow \Theta,ax; \quad x \notin a,\Gamma,\Theta}{\Gamma \Rightarrow \Theta,\Pi a}$; 16. $\dfrac{ac,\Gamma \Rightarrow \Theta}{\Pi a,\Gamma \Rightarrow \Theta}$;

17. $\dfrac{\Gamma \Rightarrow \Theta,ac}{\Gamma \Rightarrow \Theta,\exists a}$; 18. $\dfrac{ax,\Gamma \Rightarrow \Theta; \quad x \notin a,\Gamma,\Theta}{\exists a,\Gamma \Rightarrow \Theta}$;

19. $\dfrac{a,\Gamma \Rightarrow \Theta,b}{\Gamma \Rightarrow \Theta,Pab}$; 20. $\dfrac{\Delta \Rightarrow \Lambda,a; \quad b,\Gamma \Rightarrow \Theta}{Pab,\Delta,\Gamma \Rightarrow \Lambda,\Theta}$;

21. $\dfrac{a,\Gamma \Rightarrow \Theta}{\Gamma \Rightarrow \Theta,\neg a}$; 22. $\dfrac{\Gamma \Rightarrow \Theta,a}{\neg a,\Gamma \Rightarrow \Theta}$;

23. $\dfrac{\Gamma \Rightarrow \Theta,a; \quad \Gamma \Rightarrow \Theta,b}{\Gamma \Rightarrow \Theta,\&ab}$; 24. $\dfrac{a,\Gamma \Rightarrow \Theta}{\&ab,\Gamma \Rightarrow \Theta}$;

25. $\dfrac{b,\Gamma \Rightarrow \Theta}{\&ab,\Gamma \Rightarrow \Theta}$; 26. $\dfrac{\Gamma \Rightarrow \Theta,a}{\Gamma \Rightarrow \Theta,\vee ab}$;

27. $\dfrac{\Gamma \Rightarrow \Theta,b}{\Gamma \Rightarrow \Theta,\vee ab}$; 28. $\dfrac{a,\Gamma \Rightarrow \Theta; \quad b,\Gamma \Rightarrow \Theta}{\vee ab,\Gamma \Rightarrow \Theta}$.

Derivations in the A-system are constructed in the form of a tree. (Cf. systems in Kuzichev (1973a,b; 1974a,b; 1976; 1979b,c).)

I.3.

Variables and the ob Z_0 are considered to be N-terms; if a and b are N-terms, then the obs $\pi_0 a$, πa, σa, $[+]ab$, $[\cdot]ab$ are N-terms, where $\sigma = \lambda xyz.y(xyz)$ – the successor combinator, $[+] = \lambda xy.x\sigma y$ – the sum combinator, $[\cdot] = \lambda xy.x(y\sigma)Z_0$ – the product combinator, $\pi = \lambda x.xw(KZ_0)Z_1$ – the predecessor combinator $[Z_{n+1} = \sigma Z_n$ $(n \geq 0)$, $w = \lambda x.D(\sigma(xZ_0))(xZ_0)$, $D = \lambda xyz.z(Ky)x]$, $\pi_0 = \lambda x.xw(KZ_0)Z_0$.

I.4.

If a and b are N-terms, then the ob Qab is taken to be an N′-formula; if a and b are N′-formulas, then the obs Pab, &ab, $\vee$ab, $\neg$a, $\Pi\lambda$xa and $\exists\lambda$xa are also taken to be N′-formulas. If a is an N′-formula, then rk(a) is the number of occurrences of logical operators P, &, $\vee$, $\neg$, Π and $\exists$ in a (cf. the definition of rk(a) in Kuzichev (1977a, 1978a)). If a is an N′-formula and b $\leftrightarrow$ a (the ob b is convertible into ob a), then b is taken to be an N-formula, rk(b) = rk(a).

I.5.

If a is an N′-formula, then ob λxa is called an f-term.

I.6.

A derivation $\mathfrak{A}$ of a sequent E from sequents $E_1, \ldots, E_n$ in the A-system, $n \geq 0$, will be called an F-derivation if the following conditions hold:

(1) in the case of an application in $\mathfrak{A}$ of rules 16 and 17 (see

postulates of A-extension) the ob c is an N-term;

(2) in the case of an application in $\mathfrak{A}$ of rule 12 the obs ca or cb are Q-obs (Q-ob is an ob convertible into an N-term or an ob of the form Qfg; cf. (3) in the definition of N-derivation);

(3) rule 13 is not applied in $\mathfrak{A}$;

(4) in the case of an application in $\mathfrak{A}$ of rule 14 the ob c is an f-term;

(5) in the case of an application in $\mathfrak{A}$ of rule 2 the obs a and b are Q-obs;

(6) every ob belonging in $\mathfrak{A}$ to a deductive sequent (that is to a sequent of the form $\Gamma \Rightarrow \Delta$) is an N-ob or a Q-ob (N-obs are constructed analogously to N-formulas from obs of the form Na and Qab, where a and b are N-terms).

I.7.

Let $\mathfrak{A}$ be a derivation. We define inductively a dependence relation between obs in $\mathfrak{A}$ indicating for each ob in $\mathfrak{A}$ the list of obs in $\mathfrak{A}$ on which it depends:

(1) when postulates 1 or 2 are used, these lists are empty for each ob of the postulates;

(2) when rules 3-22, 24-27 are applied, the lists for each ob of the sequences Γ, Θ, Λ, Δ of the conclusion are identical, respectively, with lists for corresponding obs in the premises;

(3) when rules 4 and 5 are applied, the list for the ob a is empty;

(4) when rules 6, 7, 23 and 28 are applied, the list for the ob a (in the conclusion of rules 6 and 7) and also the lists for obs which belong to the sequences Γ and Θ (in the conclusion of rules 23 and 28) are obtained by joining the lists for corresponding obs in the premises;

(5) when rules 8 and 9 are applied, the lists of obs on which depend obs b and a in the conclusion are identical with the lists of obs on which depend obs a and b in the premises;

(6) when rules 10 and 11 are applied, the list of obs on which depends ob b in the conclusion coincides with the list of obs on which depends a in the left premise;

(7) when rules 12-28 are applied, the principal ob depends on those obs in the premises from which it is obtained and also on all the obs on which these last depend;

(8) if in a branch H of derivation $\mathfrak{A}$ a rule ∇_1 (where ∇_1 is one of rules 12, 14, 16 or 17) precedes a rule ∇ (where ∇ is one of rules 13, 15 or 18) which is applied relative to a variable having free occurrences in the ob c relative to which the rule ∇_1 is applied, then the principal ob of the rule ∇ depends on the principal ob g of the rule ∇_1 and on the obs on which g depends.

I.8.

Let $\mathfrak{B}$ be a derivation of the sequent $E = \Gamma \Rightarrow \Theta$. When rule ∇ (∇ is one of rules 15-28) is applied in $\mathfrak{B}$, it is called a $\mathfrak{B}$-logical rule (and we distinguish rules 15-18 as $\mathfrak{B}$-predicational and rules 19-28 as $\mathfrak{B}$-propositional), if in the part of derivation $\mathfrak{B}$ leading from the conclusion of rule ∇ to the end sequent E there is no application of rule 14 having principal ob which depends on the principal ob of the rule ∇.

I.9.

A derivation $\mathfrak{B}$ of a sequent $G = \Gamma \Rightarrow \Theta$ will be called prenex derivation, if $\mathfrak{B}$ has a subderivation $\mathfrak{C}$ of a sequent E, not having applications of $\mathfrak{B}$-predicational rules, and such that the part of the derivation $\mathfrak{B}$, leading from E to sequent G, consists only of applications of rules 4-11, 15-18. This sequent E will be called midsequent of the derivation $\mathfrak{B}$.

I.10.

Operators P, &, $\vee$, $\neg$, Π and $\exists$, which were applied in the definition of construction of N′-ob f (see definition 6, Kuzichev (1977a)), will be called logical operators of N′-ob f. If a is an N′-ob and $b \leftrightarrow a$, then the logical operators of a will be called logical operators of N-ob b. If in the construction of N′-ob g according to the definition the introduction of logical operators P, &, $\vee$ and $\neg$ precedes the introduction of logical quantifiers Π and $\exists$, then g will be called a prenex N′-ob. If a is a prenex N′-ob and $b \leftrightarrow a$, then b will be called a prenex N-ob. If in a sequent $E = \Gamma \Rightarrow \Theta$ each ob is a prenex N-ob, then E will be called a prenex sequent.

I.11.

Let $\mathfrak{B}$ be a derivation of sequent $\Gamma \Rightarrow \Theta$. In $\mathfrak{B}$ there can be applications of $\mathfrak{B}$-predicational rules; let us take one of them and find number of applications of rules 12, 14, 19-28 in the branch leading from the conclusion of this rule to the last sequent of derivation $\mathfrak{B}$. The sum of all these numbers for all $\mathfrak{B}$-predicational rules will be called the order of $\mathfrak{B}$ and denoted by $\psi[\mathfrak{B}]$. If in $\mathfrak{B}$ there are no applications of $\mathfrak{B}$-predicational rules, then $\psi[\mathfrak{B}] = 0$.

I.12.

Let $\mathfrak{B}$ be a derivation of sequent $\Gamma \Rightarrow \Theta$; let a be an ob in $\mathfrak{B}$; let $\alpha_1, \ldots, \alpha_r$ be all the obs from the list of obs on which a depends, which are convertible into $\Pi\lambda x.Nx$ and are obtained by an application of rule 14, $r \geqslant 0$. The list $\alpha_1, \ldots, \alpha_r$ will be called the N-list of ob a from derivation $\mathfrak{B}$, and we shall say that the ob a in $\mathfrak{B}$ has N-list $\alpha_1, \ldots, \alpha_r$.

Let β be the sequence $\alpha_1, \ldots, \alpha_n, a$. The obs from β are divided into levels: it is considered that

(1) an ob c from β belongs to the first level, if the list of

obs, on which ob c in $\mathfrak{B}$ depends, has no obs belonging to β;
(2) an ob g from β belongs to the level k + 1, if the list of obs, on which ob g in $\mathfrak{B}$ depends, has obs, belonging to the level $k(k > 0)$, and each ob of this list, appearing in β, belongs to the level $m, m \leq k$. The number of all these levels is denoted by $\gamma[\mathfrak{B}, a]$.

If $a_1, \ldots, a_s$ are all obs in $\mathfrak{B}$, convertible into obs of the form Ne and obtained by an application of rule 14, $s > 0$, then $\gamma[\mathfrak{B}] = \max\{\gamma[\mathfrak{B}, a_1], \ldots, \gamma[\mathfrak{B}, a_s]\}$; otherwise $\gamma[\mathfrak{B}] = 0$.

The number of all occurrences of deductive sequents in the derivation $\mathfrak{B}$ is denoted by $\phi[\mathfrak{B}]$.

LEMMA 1. If a is an N-formula and $b \leftrightarrow a$, then one can construct an F-derivation of the sequent $a \Rightarrow b$.

The lemma is proved by induction with respect to rk(a), making use of the Church-Rosser theorem.

Construction of F-derivations of sequents corresponding to axioms of arithmetic: it is not difficult to verify that the N-derivations $\mathfrak{A}^{14}, \ldots, \mathfrak{A}^{21}$ constructed in Kuzichev (1977b) are F-derivations, and that the N-derivation $\mathfrak{A}^{13}$ is an F-derivation for every N'-formula A (moreover, the sequents $U \Rightarrow Ta$ and $(\lambda z.zx)a \Rightarrow A$ are derived with the aid of suitable conversion by virtue of lemma 1, and the fact that $a = \lambda xA$ is an f-term). Axiom scheme $A \Rightarrow A$ (where A is a formula of arithmetic) corresponds to the last sequent $A \Rightarrow A$ of F-derivation, constructed on the basis of conversion $A \leftrightarrow A$ and of lemma 1.

LEMMA 2. If b is an N-term and $\mathfrak{A}$ is an F-derivation of the sequent $E = \Gamma \Rightarrow \Delta$, then $\mathfrak{B} = [b/x]\,\mathfrak{A}$ is an F-derivation of the sequent $[b/x]E$ such that $\phi[\mathfrak{B}] = \phi[\mathfrak{A}]$, $\gamma[\mathfrak{B}] = \gamma[\mathfrak{A}]$ and $\psi[\mathfrak{B}] = \psi[\mathfrak{A}]$.

The lemma is proved by induction with respect to $\phi[\mathfrak{A}]$ in accordance with definition 14 of Kuzichev (1977a).

THEOREM 1 (on cut for F-derivations). If $\mathfrak{A}$ and $\mathfrak{B}$ are F-derivations of sequents $\Delta \Rightarrow \Lambda, a$ and $a, \Gamma \Rightarrow \Theta$ then one can construct an F-derivation of the sequent $\Delta, \Gamma \Rightarrow \Lambda, \Theta$.

The proof is analogous to the proof of theorem 2 of Kuzichev (1977a, 1978b). Compare the theorem on cut in first-order predicate logic and the corresponding propositions for systems with canonical restrictions of Curry.

THEOREM 2 (on consistency relative to negation for F-derivations). For every ob a, there exist no F-derivations of sequents $\Rightarrow a$ and $\Rightarrow \neg a$.

The theorem is proved by contradiction, applying rule 22, theorem 1 and theorem on absolute consistency of the A-system.

THEOREM 3 (on embedding of formal arithmetic). If the sequent $\Gamma \Rightarrow \Theta$ is derivable in the calculus LA, then one can construct an F-derivation of the sequent $\Pi\lambda x.Nx, \Gamma \Rightarrow \Theta$.

The theorem is proved analogously to the theorems 1 and 4 of Kuzichev (1977b, 1978b), applying theorem 1 and F-derivations of sequents corresponding to axioms of arithmetic; we note that in the case of an application in LA of the cut rule we can use instead of theorem 1 the rule 14 which, evidently, is stronger than the cut rule.

It is well known that a formula f of arithmetic is derivable in calculus HA iff the sequent $\Rightarrow$ f is derivable in calculus LA. Therefore, in view of theorem 3, for every formula f derivable in HA there exists an F-derivation of the sequent $\Pi\lambda x.Nx \Rightarrow f$. Let us remark that the assumption of the inconsistency of HA implies the derivability in LA of the 'empty' sequent $\Rightarrow$. By theorem 3, from the derivability of $\Rightarrow$ in LA we obtain the F-derivability of $\Pi\lambda x.Nx \Rightarrow$. Thus, the problem of the consistency of HA reduces to a proof of nonderivability in the class $\mathfrak{M}$ of all F-derivations of the sequent $\Pi\lambda x.Nx \Rightarrow$.

In the case of an application in F-derivations of rules 4 and 5 we suppose henceforth that rk(a) = 0, where a is the ob indicated in the conclusion of rules; for the rule 5 we demand furthermore that the ob a is not convertible into an ob of the form Ne. In lemma 3 below we suppose the given F-derivation $\mathfrak{A}$ to be transformed (with conservation of parameter $\gamma[\mathfrak{A}]$) so that by an application in $\mathfrak{A}$ of rules 15 and 18 we have for each ob f of each deductive sequent H belonging to the derivation $\mathfrak{R}$ of the premise of the corresponding rule, which is convertible into the principal ob g of the rule, $\gamma[\mathfrak{R},f] = \gamma[\mathfrak{C},g] > \gamma[\mathfrak{C},d]$, where $\mathfrak{C}$ is a derivation of the conclusion of the considered rule, d is an ob in $\mathfrak{C}$ convertible into g, and in the case of rule 15 the ob f belongs to the succedent, in the case of rule 18 f belongs to the antecedent of H. Moreover, if in the branch of the derivation $\mathfrak{A}$ the rule Ω from the list 12-28 precedes the considered rule (15 or 18) which is applied relative to the variable having free occurrences in ob b (on which the principal ob c of rule Ω depends), then $\gamma[\mathfrak{C},g] > \gamma[\mathfrak{A},c]$. It is not difficult to verify that these suppositions do not lead to loss of generality.

THEOREM 4 (on midsequent for F-derivations). Let $\mathfrak{A}$ be an F-derivation of a prenex sequent $G = \Gamma \Rightarrow \Theta$. Then one can construct a prenex F-derivation $\mathfrak{B}$ of sequent G such that $\gamma[\mathfrak{B}] \leqslant \gamma[\mathfrak{A}]$ and if $\mathfrak{C}$ is a subderivation of the midsequent of derivation $\mathfrak{B}$, then $\gamma[\mathfrak{C}] < \gamma[\mathfrak{B}]$. (Compare the theorem on midsequent in first-order predicate logic; see, for example, theorem 50 of §79 Kleene (1952).)

Proof. The theorem is proved by induction with respect to $\alpha = \psi[\mathfrak{A}]$ as in Kuzichev (1978e) Theorem 1. For $\alpha = 0$ we set $\mathfrak{B} = \mathfrak{A}$.

Assume $\alpha > 0$. Then $\mathfrak{A}$ has an $\mathfrak{A}$-predicational rule ∇ after which in the branch there is at least one application of rule ∇_1 (where ∇_1 is one of the list 12, 14, 19-28) whose principal ob does not depend on the principal ob of rule ∇. Using rules 4-11 and (in the case when ∇ is 15 or 18) lemma 2, we change the order of applications of rules ∇ and ∇_1 obtaining F-derivation $\mathfrak{R}$ of sequent G such that $\psi[\mathfrak{R}] < \psi[\mathfrak{A}]$ and $\gamma[\mathfrak{R}] \leqslant \gamma[\mathfrak{A}]$. Applying the inductive hypothesis, we find from $\mathfrak{R}$ the desired F-derivation $\mathfrak{B}$ of sequent G.

LEMMA 3 (on deletion of $NZ_k, k \geqslant 0$). Let $\mathfrak{A}$ be an F-derivation of a sequent $E = \Gamma \Rightarrow \Theta$. Then one can construct an F-derivation $\mathfrak{B}$ of the sequent $\Gamma_\nu \Rightarrow \Theta_\nu$ such that

(1) sequences Γ_ν and Θ_ν are obtained from Γ and Θ respectively by deleting all obs convertible into NZ_k;
(2) $\gamma[\mathfrak{B}] \leqslant \gamma[\mathfrak{A}]$;
(3) if $\mathfrak{A}$ has an application of rule 14 with the principal of c, convertible into NZ_k, and if for every ob b from $\mathfrak{A}$ which is convertible into an ob of the form Ne and is obtained with the aid of rule 14 ($b \not\leftrightarrow c$), we have $\gamma[\mathfrak{A},c] > \gamma[\mathfrak{A},b]$, or if $\mathfrak{A}$ has no obs convertible into obs of the form Ne, non-convertible into c and obtained by an application of rule 14, then $\gamma[\mathfrak{B}] < \gamma[\mathfrak{A}]$;
(4) for every ob d of derivation $\mathfrak{B}$, $\gamma[\mathfrak{B},d] = \gamma[\mathfrak{A},d^*]$, where d^* is an ob of $\mathfrak{A}$ on the basis of which d is obtained in $\mathfrak{B}$;
(5) in the case of an application in $\mathfrak{B}$ of rule 14 the principal ob of rule 14 is not convertible into NZ_k.

Proof. The lemma is proved by induction on $\alpha = \phi[\mathfrak{A}]$ as in the proof of Lemma 2 of Kuzichev (1978f). This is essentially the proof of Lemma 8 of Kuzichev (1978d).

LEMMA 4. There exists no F-derivation of the sequent $T = \Pi\lambda x.Nx \Rightarrow$.

The proof of Lemma 4 is the same as that of Lemma 3 of Kuzichev (1978f).

THEOREM 5 (on the consistency of formal arithmetic). There does not exist a formula f of arithmetic such that $\neg f$ and f are derivable in the calculus HA of formal arithmetic.

Proof. The theorem is proved by contradiction. Assume HA inconsistent. Then the 'empty' sequent $\Rightarrow$ is derivable in LA, and, by theorem 3, there exists an F-derivation of the sequent $\Gamma\lambda x.Nx \Rightarrow$, and that contradicts Lemma 4.

II. EMBEDDING THEOREM

In the present part we propose such restrictions on the application of rules of the A-system that the corresponding class of derivations will be a conservative extension of formal arithmetic.

An F-derivation $\mathfrak{A}$ will be called an L-derivation, if $\mathfrak{A}$ has only such Q-obs which are convertible into N-terms or into obs of the form Qfg where f and g are N-terms.

Remarks.

1. The results of part I are preserved, if everywhere in the statements we replace F-derivations by L-derivations.
2. The results of this paper are preserved, if ob a is

considered to be a Q-ob iff a is convertible into an ob of the form Qfg where f and g are N-terms, or if the clause (6) of the definition of F-derivation (L-derivation, definition 6) is reformulated as following: every ob belonging in $\mathfrak{A}$ to a deductive sequent either is an N-formula or is convertible into $\Pi\lambda x.Nx$ or into an ob of the form Nh, where h is N-term. In this connection, taking into account clause (2) of definition 6, we let correspond to the axiom scheme 15 of calculus HA the following derivation: $\Rightarrow$ Qaa; $\Rightarrow$ Dc(Qaa)(σa); Q(σa)$Z_0 \Rightarrow$ Q(σa)Z_0; c $\Rightarrow$; Dc(Qaa)$Z_0 \Rightarrow$; Q(σa)$Z_0 \Rightarrow$; $\Rightarrow \neg$(Q(σa)Z_0), where a is an N-term, c = &($\neg$(Q(σa)Z_0))(Q(σa)Z_0), and we use postulates 1, 10, 2, 22, 24, 9, 25, 7, 11, 12, 21 of A-extension. Since the rule 3 is applied in Kuzichev (1977a,b; 1978a,b) only for the construction of the derivation $\mathfrak{A}^{15}$, one can exclude rule 3 from the definition of A-system.

3. In the remaining part of present work clauses (4) and (5) of the definitions of L-derivation are not applied.

THEOREM 6 (embedding in a sharper form). A formula f is derivable in the calculus HA of formal arithmetic iff the sequent $\Pi\lambda x.Nx \Rightarrow$ f has an L-derivation.

The proof of the 'if' part is based upon a result analogous to theorem 3. The converse statement was proved by A.A. Kuzichev in Spring 1979; it is based upon the next lemma*.

LEMMA 5. If an ob a is in normal form and has not a form λyd, and obs b and c are such that [a/x]b red c, then there is an ob f such that b red f and c can be obtained from [a/x]f by renaming of bound variables.

*See A.A. Kuzichev, A.S. Kuzichev 'On the embedding of formal arithmetic into combinatorially complete systems' (to appear).

III. A VARIANT OF FORMALIZATION OF LOGICAL AND MATHEMATICAL THEORIES

As was noted above, the absolutely consistent A-system is internally inconsistent: there exists an ob a such that the sequents $\Rightarrow$ a and $\Rightarrow \neg$a are provable. It is natural to seek in the class of all derivations for such systems, subsystems in which for every ob a, there exist no derivations of the sequents $\Rightarrow$ a and $\Rightarrow \neg$a. In Kuzichev (1976, 1978c) to solve this problem for systems with different elementary operators we have investigated the classes of obs having normal forms and have defined the concept of theory in these systems. As an example of a theory we can take the class of derivations which coincides (modulo convertibility) with Gentzen's system of first-order predicate logic. A generalization of the notion of theory, connected with the operators Q and Ξ (see Kuzichev 1979), permits us to construct consistent theories in which formal arithmetic and some set theoretical calculi are expressed. The introduced classes of N-, A-, F- and L-derivations are examples of theories with the operators Q and Ξ.

We note that in the proposed systems the properties of connectives of predicate logic in connection with the principle of combinatorial completeness permit us to construct derivations $\mathfrak{A}$ and $\mathfrak{B}$ of sequents $\Rightarrow$ a and $\Rightarrow \neg$a for some ob a without recourse to obs which do not have normal forms. [For example, see in the Addition to Kuzichev (1974a), a = $\exists\lambda z.\Pi\lambda y.\Xi(zy)(\neg(yy))$.] It is not difficult to verify that the derivation $\mathfrak{A}$ belongs to one theory, and the derivation $\mathfrak{B}$ to the other [for $\mathfrak{A}$ and $\mathfrak{B}$ from the Addition to Kuzichev (1974a), the class of λ-terms (obs, correponding to the terms of first-order predicate calculus) of the first theory contains the ob T = $\lambda x.\neg(xx)$, but T does not occur in the corresponding class of the second theory].

Here we give a further generalization of this notion of theory. As an example of a theory we construct a class of derivations in which calculus CL of combinatory logic can be embedded.

III.1. Definitions and notations.

We shall say that classes $\mathfrak{M}$, $\mathfrak{N}$ and X of obs form a signature $<\mathfrak{M},\mathfrak{N},X>$, if

(1) X is a class of variables, and for every list of variables $x_1, \ldots, x_n$ from X, there exists a variable y from X, different from $x_1, \ldots, x_n$;

(2) in the case of a construction [b/y]a, where y is variable from X, new variables are selected only from X;

(3) X is a subclass of $\mathfrak{N}$;

(4) for all obs d and f from $\mathfrak{N}$ and every variable y from X, the ob [d/y]f belongs to $\mathfrak{N}$;

(5) $\mathfrak{M}$ is a nonempty class of obs;

(6) for all obs d from $\mathfrak{N}$ and g from $\mathfrak{M}$ and for every variable y from X, the ob [d/y]g belongs to $\mathfrak{M}$.

In the signature $M = <\mathfrak{M},\mathfrak{N},X>$ variables from X will be called object or individual variables (in M), obs from $\mathfrak{N}$ will be called terms (in M), and obs from $\mathfrak{M}$ will be called elementary formulas (in M).

We define inductively formulas (in M) and rk(A), where A is a formula (in M):

(1) if A is an elementary formula, then A is a formula (in M), rk(A) = 0;

(2) if A and B are formulas (in M), rk(A) = i, rk(B) = j, x is an individual variable (in M), and G is an ob of the form γAB, $\delta\lambda xA$ or $\neg A$ (where γ is P, & or v, δ is Π or $\exists$) [and G is not convertible into obs from $\mathfrak{M}$,], then G is a formula (in M), and rk(γAB) = i+j+1, rk($\delta\lambda$xA) = i+1, rk($\neg$A) = i+1.

If A is a formula (in M) and b $\leftrightarrow$ A, then the ob b will be called an M-formula, and rk(b) = rk(A). If a is not an M-formula, then rk(a) = 0. If A is a formula (in M), then ob λx.A will be called an f-term (in M).

III.2.

Let $M = \langle \mathfrak{M}, \mathfrak{N}, X\rangle$ be a signature. We shall define in the A-system a class $T(M)$ of derivations which will be called a theory (in M).

A derivation $\mathfrak{A}$ belongs to $T(M)$ iff

(1) in the case of an application in $\mathfrak{A}$ of rules 16 and 17 the ob c is a term (in M);
(2) when the rule 12 is applied in $\mathfrak{A}$, rk(ca) = 0 or rk(cb) = 0;
(3) the rule 13 is not applied in $\mathfrak{A}$;
(4) in the case of an application in $\mathfrak{A}$ of the rule 14 the ob c is an f-term (in M);
(5) when the rule 2 is applied in $\mathfrak{A}$, rk(a) = 0;
(6) every ob, belonging in $\mathfrak{A}$ to a deductive sequent, is an M-formula [in the case of an application in $\mathfrak{A}$ of logical rules 15-28, rk(d) > 0, where d is the principal ob of the corresponding rule].

By analogy to proofs of theorems 1 and 2, we prove theorems on admissibility of the cut rule in a theory $T(M)$ and on consistency of $T(M)$ relative to negation.

Consider examples showing the possibility of such an approach. Let H be the class of all variables of the A-system.

1. We define classes $\mathfrak{M}_1$ and $\mathfrak{N}_1$: variables, obs Z_0 and K belong to $\mathfrak{N}_1$; if a and b are obs from $\mathfrak{N}_1$, then the obs $\pi_0 a$, πa, σa, $[+]ab$, $[\cdot]ab$ belong to $\mathfrak{N}_1$, and the obs Na, Qab belong to $\mathfrak{M}_1$.

By the construction, $T_1 = T(\langle \mathfrak{M}_1, \mathfrak{N}_1, H\rangle)$ is an extension of the class of all F-derivations. However, this extension is non-conservative: for example, the sequent $\Pi\lambda x.Nx \Rightarrow$ is provable in T_1.

Indeed, we set $b = QKK, \bot = QZ_0K$. Then we find successively:

$$\cfrac{\cfrac{\cfrac{\Rightarrow b}{\Rightarrow Z_0(\neg b)b} \quad \cfrac{\cfrac{\Rightarrow b}{\neg b \Rightarrow}}{K(\neg b)b \Rightarrow}}{QZ_0K \Rightarrow}}{\Rightarrow \neg(QZ_0K)} . \qquad (*)$$

Since $\sigma x Z_0 b(P\bot)\bot \leftrightarrow Z_0(xZ_0b)(P\bot)\bot \leftrightarrow P\bot\bot$ and $KZ_0b(P\bot)\bot \leftrightarrow Z_0(P\bot)\bot \leftrightarrow \bot$, we obtain

$$\cfrac{\cfrac{\cfrac{\cfrac{\Rightarrow \sigma x Z_0 b(P\bot)\bot \quad KZ_0 b(P\bot)\bot \Rightarrow}{Q(\sigma x)K \Rightarrow}}{\Rightarrow \neg(Q(\sigma x)K)}}{\neg(QxK) \Rightarrow \neg(Q(\sigma x)K)}}{\Rightarrow \Pi\lambda x.P(\neg(QxK))(\neg(Q(\sigma x)K))} . \qquad (**)$$

Then (*) and (**) generate

$$\cfrac{\cfrac{\cfrac{\Rightarrow \&(\neg(QZ_0K))(\Pi\lambda x.P(\neg(QxK))(\neg(Q(\sigma x)K)))}{\Rightarrow (\lambda y.\&(yZ_0)(\Pi\lambda x.P(yx)(y(\sigma x))))\lambda t.\neg(QtK)} \quad \cfrac{\cfrac{\Rightarrow QKK}{\neg(QKK) \Rightarrow}}{(\lambda z.zK)\lambda t.\neg(QtK) \Rightarrow}}{NK \Rightarrow}}{\Pi\lambda x.Nx \Rightarrow} .$$

Compare the lemma 4 on nonderivability of the last sequent in the class of all F-derivations.

2. For a construction theory T_2, containing the calculus CL of combinatory logic, we introduce classes $\mathfrak{M}_2$ and $\mathfrak{N}_2$ as follows: obs I, K, S and variables belong to $\mathfrak{N}_2$; if a and b are obs from $\mathfrak{N}_2$, then the ob ab belongs to $\mathfrak{N}_2$, and the ob Qab belongs to $\mathfrak{M}_2$.

By induction with respect to construction of derivations in CL, we prove that if a formula A is derivable in CL, then the sequent $\Rightarrow$ A is derivable in the theory $T_2 = T(<\mathfrak{M}_2, \mathfrak{N}_2, H>)$.

For example, the sequent $\Rightarrow$ Q(Kab)a is obtained in T_2 for all obs a and b on the basis of the axiom $\Rightarrow$ Qaa and the conversion Kab $\leftrightarrow$ a and applying rule 10. The derivation of sequent $\Rightarrow \neg$(QSK) is as follows: SIQK($\neg$b)K $\leftrightarrow$ b and KIQK($\neg$b)K $\leftrightarrow \neg$b, where b = QKK, consequently, we have

$$\dfrac{\dfrac{\dfrac{\Rightarrow b}{\Rightarrow SIQK(\neg b)K} \quad \dfrac{\neg b \Rightarrow}{KIQK(\neg b)K \Rightarrow}}{QSK \Rightarrow}}{\Rightarrow \neg(QSK)}\ .$$

The consistency of CL follows from the consistency of T_2 relative to negation.

Remarks.

1. Suppose we exclude the expressions put in square brackets from the clause (6) of the definition of $T(M)$ and from the clause (2) of the definition of formulas (in M), but add the following clause (7) to the definition of the signature $<\mathfrak{M}, \mathfrak{N}, X>$:

> (7) obs from $\mathfrak{M}$ are not convertible into obs of form γAB, $\delta\lambda$xA and $\neg$A (where γ is P, & or v, δ is Π or $\exists$, x is a variable from X, A and B are obs).

Then the results of the paper are preserved, though the class of theories (in M) decreases.

2. Suppose that in the clause (2) of the definition of formulas (in M) we replace the phrase 'is not convertible into obs from' by the phrase: 'does not belong to' (or exclude the expression put in square brackets, though in this case we must take into account rk(g) for every ob g of the postulates) and suppose we replace the definition of M-formula by the following:

> every formula (in M) will be called M-formula; if a is a formula (in M) and b $\leftrightarrow$ a, where the ob b is not a formula (in M), then b will be called M-formula, rk(b) = rk(a).

Then the class of the theories (in M) widens, but the results of the paper are preserved.

REFERENCES

Church, A. (1941). 'The Calculi of Lambda-Conversion', Princeton University Press, Princeton.

Curry, H.B., Feys, R. and Craig, W. (1958). 'Combinatory Logic', Vol. I, North-Holland Co., Amsterdam.

Curry, H.B., Hindley, J.R. and Seldin, J.P. (1972). 'Combinatory Logic', Vol. II, North-Holland Co., Amsterdam.

Hindley, J.R., Lercher, B. and Seldin, J.P. (1972). 'Introduction to Combinatory Logic', Cambridge University Press, Cambridge.

Kleene, S.C. (1952). 'Introduction to Metamathematics', Van Nostrand, New York.

Kuzichev, A.S. (1973a). A system of λ-conversion with a deductive operator of formal implication, *Dokl. Akad. Nauk SSSR* 212 (6), 1290-1292. English translation, *Soviet Math. Doklady* 14 (5), 1584-1587.

Kuzichev, A.S. (1973b). Consistent extensions of pure combinatory logic, *Vestnik Moskov. Univ. Ser. I Mat. Meh.* 28 (3), 76-81. English translation, *Moscow Univ. Math. Bull.* 28 (3), 66-72.

Kuzichev, A.S. (1974a). Deductive operators of combinatory logic, *Vestnik Moskov. Univ. Ser. I Mat. Meh.* 29 (3), 13-21. English translation, *Moscow Univ. Math. Bull.* 29 (3), 8-14.

Kuzichev, A.S. (1974b). On the expressive potentialities of deductive systems of λ-conversion and combinatory logic, *Vestnik Moskov. Univ. Ser. I Mat. Meh.* 29 (6), 19-26. English translation, *Moscow Univ. Math. Bull.* 29 (6), 58-64.

Kuzichev, A.S. (1976). Combinatorially complete systems with operators Ξ, F, Q, Π, ∃, P, ¬, &, ∨, ≡, *Vestnik Moscov. Univ. Ser. I. Mat. Meh.* 31 (6), 53-59. English translation, *Moscow Univ. Math. Bull.* 31 (6), 117-122.

Kuzichev, A.S. (1977a). A system of λ-conversion with logical operators and an equality operator, *Dokl. Akad. Nauk SSSR* 236 (4), 796-799. English translation, *Soviet Math. Doklady* 18 (5), 1268-1272.

Kuzichev, A.S. (1977b). Formal arithmetic in the Ⱥ-system of λ-conversion, *Dokl. Akad. Nauk SSSR* 236 (5), 1072-1075. English translation, *Soviet Math. Doklady* 18 (5), 1327-1330.

Kuzichev, A.S. (1977c). Cut theorem for 𝔑-theories in combinatorially complete systems, *Vestnik Moskov. Univ. Ser. I Mat. Meh.* 32 (1), 62-67. English translation, *Moscow Univ. Math. Bull.* 32 (1), 53-57.

Kuzichev, A.S. (1978a). Formal arithmetic in combinatorially complete systems I, *Vestnik Moskov. Univ. Ser. I Mat. Meh.* 33 (2), 13-21. English translation, *Moscow Univ. Math. Bull.* 33 (2), 9-15.

Kuzichev, A.S. (1978b). Formal arithmetic ir combinatorially complete systems II, *Vestnik Moskov. Univ. Ser. I Mat. Meh.* 33 (3), 60-66. English translation, *Moscow Univ. Math. Bull.* 33 (3), 55-60.

Kuzichev, A.S. (1978c). The Curry paradox in combinatorially complete systems, *The History and Methodology of Natural Sciences (Moscow University)* 20, 81-98.

Kuzichev, A.S. (1978d). A theorem on the consistency of formal arithmetic, *Dokl. Akad. Nauk SSSR* 238 (2), 269-272. English translation, *Soviet Math. Doklady* 19 (1), 30-33.

Kuzichev, A.S. (1978e). The theorem on the midsequent in the A-system of λ-conversion, *Dokl. Akad. Nauk SSSR* 243 (1), 19-21. English translation, *Soviet Math. Doklady* 19 (6), 1305-1308.

Kuzichev, A.S. (1978f). On the consistency of formal arithmetic, *Dokl. Akad. Nauk SSSR* 243 (5), 1123-1126. English translation, *Soviet Math. Doklady* 19 (6), 1486-1490.

Kuzichev, A.S. (1979). Classes of objects, having normal forms, in one system of λ-conversion with logical operators, *Dokl. Akad. Nauk SSSR* 249 (1), 41-45.

Kuzichev, A.S. (1979b). Type-free extensions of the calculi of λ-conversion and of pure combinatory logic, VI Intern. Congress for Logic, Methodology and Philosophy of Science (Hannover, August 1979), abstracts, 113-117.

Kuzichev, A.S. (1979c). Formal arithmetic in one system of λ-conversion with logical operators, *Zeitschr. f. math. Logik* 25 (4), 299-314.

Schönfinkel, M. (1924). Über die Bausteine der mathematischen Logik, *Math. Annalen* 92, 305-316.

Stenlund, S. (1972). 'Combinators, λ-Terms and Proof Theory', Reidel, Dordrecht.

COMBINATORY AND LAMBDA SYNTAX

OPTIMAL REDUCTIONS IN THE LAMBDA-CALCULUS

Jean-Jacques Lévy
INRIA-LABORIA
Domaine de Voluceau
78150 Le Chesnay, France

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

The standardisation theorem [Cu] implies that leftmost-outermost reductions reach normal forms whenever they exist. Thus these reductions insure termination and can be called correct in the sense of [Vu,Do,Le1]. However, they may not be optimal, i.e. reach the normal form in a minimum number of steps because of duplications of redexes. At first glance, it seems difficult to design optimal reduction strategies. First, innermost redexes seem better than the outermost ones because they avoid copying redexes. Secondly, one does not want to reduce useless redexes, i.e. redexes not reduced in the leftmost-outermost reduction, but they cannot be found effectively in the λ-calculus. Worse, even in the λI-calculus where all redexes are useful for the normal form, innermost reductions may not be optimal. Take for instance $(\lambda x.xI)(\lambda y.(\lambda z.zzzz)(yt))$ with $I=\lambda x.x$. However, some optimal strategies have been defined in [Vu] for recursive programs schemes. Independently, an efficient, though not optimal, strategy was designed by [Wa] for the λ-calculus. Both methods use the same simple principle : at each reduction step, contract the leftmost-outermost redex and avoid duplications of redexes by adding some sharing mechanism. This is done by leaving the usual universe of λ-expressions and by considering shared λ-expressions. For instance, let $M=\Delta((\lambda x.xy)I)$ with $\Delta=\lambda x.xx$ and $I=\lambda x.x$. The graph normal reduction method in [Wa] works for M as follows :

(i) $M \to (\circ \;\; \circ)\;\; (\lambda x.xy)I \to (\circ \;\; \circ)\;\; Iy \to (\circ \;\; \circ)\;\; y$

Thus both copies of $(\lambda x.xy)I$ and Iy are simultaneously contracted. The corresponding reduction in the usual λ-calculus framework is :

(ii) $M \to ((\lambda x.xy)I)((\lambda x.xy)I) \to (Iy)(Iy) \to yy.$

But shared reductions are not so easy in the λ-calculus because of functional arguments (which cannot happen in recursive program schemes considered in [Vu]). Take for instance $M=(\lambda x.(xz)(xt))(\lambda y.Iy)$. Then :

(iii) $M \to (\circ\; z)(\circ\; t)\;\; (\lambda y.Iy) \to ???$

The leftmost outermost redex $(\lambda y.Iy)z$ has to be reduced without duplicating the Iy redex. The evaluator of [Wa] cannot handle this example and makes copies of $(\lambda y.Iy)$ in this case, which makes it non-optimal. Example (iii) shows that, in order to get an optimal λ-evaluator, one must share, not only subexpressions, but pairs of subexpressions and substitutions for free variables, i.e. closures in the programming languages terminology. To our knowledge, such optimal evaluators have not yet been designed for the λ-calculus, and Wadsworth's method is still the most efficient one. Unfortunately, we shall not give here a solution to that problem. But we will characterize exactly the redexes whose contraction needs to be shared. For doing this, we shall study the duplication of redexes in the usual λ-calculus setting.

Take again example (i). In the corresponding reduction (ii), at each step the leftmost-outermost redex is at least contracted. But in the second and third steps, some additional redexes are reduced. In the second step, the two $((\lambda x.xy)I)$ contracted redexes are copies (or residuals) of one single redex in M. In step three, the two Iy redexes are not residuals of one redex in reduction (ii), but they are residuals of a single redex if one permutes the two first steps, which gives reduction

(iv) $M \to \Delta(Iy) \to (Iy)(Iy) \to yy$.

This key observation leads to the organisation of our paper. First, we recall definitions and properties of permutations of reductions. The full treatment of them has been done in [Be1] for recursive programs schemes, but is quite similar in the λ-calculus. Secondly, we formalize duplications of redexes as residuals modulo permutations of reductions. We show that its symmetric and transitive closure, called the family relation, is decidable. This implies that one can effectively find maximum sets of redexes which are duplications of a single redex. Thus complete reductions, i.e. reductions which contract at each step such maximum sets, are effective reduction strategies and can be proved finally optimal if they contract, at each step, at least leftmost-outermost redexes.

Readers interested in a more complete treatment are sent to [Le2]. A similar paper for the easier formalism of recursive programs schemes is [Be1], where results of [Vu] are shown along the lines of this paper. Also, optimal reductions have been studied in [O'D] (although we disagree with some of his results [Be2]) and in [St] for combinatory logic (which is an easier case than the λ-calculus).

2. PRELIMINARIES

If V is an infinite set of variables, the set of λ-expressions built on V is the minimum set containing V and closed by abstraction and application, i.e. (λx.M) and (MN) are λ-expressions when M,N are already λ-expressions and x is a variable. Parentheses are suppressed as much as possible. Around applications, parentheses may be omitted by association to the left. There is the usual notion of free and bound variables. We do not take care of names of bound variables and thus forget all the meaningless problems of the so-called α-rule. Therefore, equality of λ-expressions will be equality modulo α-interconvertibility and, for instance, we do not hesitate to write $\lambda x.xy=\lambda t.ty$.

Here, only the β-rule is considered. Let say that M can be immediately reduced to N, written $M\to N$, iff M and N only differ by a subexpression which is of the form $(\lambda x.P)Q$ in M and $P[x\backslash Q]$ in N (where $P[x\backslash Q]$ is the result of substituting Q to all occurrences of the free variable x in P). A subexpression of the form $(\lambda x.P)Q$ is called a redex. The transitive closure of $\to$ is written $\overset{*}{\to}$ and, thus, $M\overset{*}{\to}N$ means that M can be reduced to N. An expression without redexes is a normal form and, if $M\overset{*}{\to}N$ with N in normal form, then M has a normal form.

Several reductions are possible between two expressions. If one wants to be more specific about one reduction, names can be given to them. We use letters ρ,σ,τ. And one has to specify the initial expression and the successive redex-occurrences contracted at each step. In order to be very precise, one needs some addressing mechanism of redexes in λ-expressions. Here we avoid it by confusing redexes and their occurrences and by always assuming that this distinction is clear from the context. Thus we write $\rho:M\xrightarrow{R}N$ to say that ρ is the immediate reduction consisting in contracting the redex (occurrence) R in M. More generally, a reduction ρ can be specified by writing :

(v) $\rho:M\xrightarrow{R_1}M_1\xrightarrow{R_2}M_2\ \dots\xrightarrow{R_n}M_n$.

We use also the notation $\rho:M\overset{*}{\to}N$ for meaning that ρ is a reduction from M to N. By $\rho\sigma$, we mean the obvious composition of ρ and σ. The empty reduction starting at M will be written, 0_M or simply 0, when M is clear from the context. Similarly, we can forget the initial expression of an immediate reduction and write just the redex (occurrence) R instead of $M \xrightarrow{R} N$. Thus (v) is equivalent to $\rho : M \overset{*}{\to} N$ and $\rho = R_1 R_2 \ldots R_n$. Letters R,S,T will be reserved to redex-occurrences.

Now, suppose that R is a redex in M and $\rho : M \overset{*}{\to} N$. Then R can be copied, modified, eliminated or contracted during ρ. The set R/ρ of redexes corresponding to R is called the set of <u>residuals</u> of R by ρ. First, one has :

$$R/0 = \{R\}$$

$$R/\rho\sigma = \{T \mid T \in S/\sigma, S \in R/\rho\}$$

If $\rho : M \xrightarrow{S} N$, then $R/\rho = R/S$ is defined by one tedious inspection of the relative positions of R and S. If R is not contained in S, then $R/S = \{R'\}$ where R' is the redex of N which is at the same place as R in M. If R coincides with S, then $R/S = \emptyset$. If R is strictly in $S = (\lambda x.A)B$, there are two possibilities. First R is in A, and $R/S = \{R'\}$ is the redex which corresponds to R in the contractum $A[x \backslash B]$ of S in N. (Remark then that $R' = R[x \backslash B]$). Otherwise R is in B and $R/S = \{R_1, R_2, \ldots R_n\}$ where n is the number of occurrences of the free variable x in A and every R_i corresponds to R in the $i^{\underline{th}}$ instance of B in the contractum $A[x \backslash B]$ of S in N.

Finally, it will be necessary to consider another kind of reductions, <u>parallel reductions</u>. Let $\mathcal{F}$ be a set of (maybe nested) redexes in M. The immediate parallel reduction $M \xrightarrow{\mathcal{F}} N$ can be defined by use of the following definitions and theorem. A reduction $\rho = R_1 R_2 \ldots R_n \ldots$ is <u>relative</u> to $\mathcal{F}$ iff $\forall n \geq 1 \quad R_n \in \mathcal{F}/(R_1 R_2 \ldots R_{n-1})$. Moreover ρ is a <u>development</u> of $\mathcal{F}$ iff ρ is relative to $\mathcal{F}$ and $\mathcal{F}/\rho = \emptyset$. Then

THEOREM 2.1. (Finite developments theorem) : [Cu]. Let $\mathcal{F}$ be a set of redexes in an expression M. Then

1) there is no infinite reduction relative to $\mathcal{F}$,

2) all developments end at a same expression,

3) for all redex R in M, if ρ and σ are two developments of F, then $R/\rho=R/\sigma$.

Thus, the order in which redexes of F are contracted is not relevant and parallel reductions can be defined now without ambiguity as reductions $M \xrightarrow{F_1} M_1 \xrightarrow{F_2} M_2 \ldots \xrightarrow{F_n} M_n$ contracting some set of redexes at each step. Non parallel reductions are a particular case of parallel ones since, at each step, a singleton set of redexes is contracted. We generalize all the previous notations to parallel reductions without difficulty.[1] In the rest of the paper, only parallel reductions are considered. Therefore we simply call them reductions. But we do not hesitate to write $\rho=R_1R_2\ldots R_n$ for $\rho=\{R_1\}\{R_2\}\ldots\{R_n\}$. Remark finally that $\emptyset\neq 0$, i.e. one immediate contraction of an empty set of redexes is not an empty (parallel) reduction.

3. PERMUTATIONS OF REDUCTIONS

If F and C are two sets of redexes in M, let $F\sqcup C=F(C/F)$. Then one useful corollary of the finite developments theorem is

LEMMA 3.1. (Lemma of parallel moves) [Cu] : Let F and C be two sets of redexes in M. Then

1) $F\sqcup C$ and $C\sqcup F$ end at the same expression,

2) $H/(F\sqcup C)=H/(C\sqcup F)$ for any set H of redexes in M.

This can be summarized in figure 1. Now, permutations of reductions may be defined.

Definition 3.2. The <u>equivalence</u> of reductions <u>by permutations</u> is the least congruence with respect to composition satisfying the lemma of parallel moves and elimination of empty steps.

More explicitly, this relation $\equiv$ is the least equivalence relation satisfying :

1) $F \sqcup C \equiv C \sqcup F$ when F and C are two sets of redexes in a same expression,
2) $\emptyset \equiv 0$,
3) $\rho\sigma\tau \equiv \rho\sigma'\tau$ if $\sigma \equiv \sigma'$.

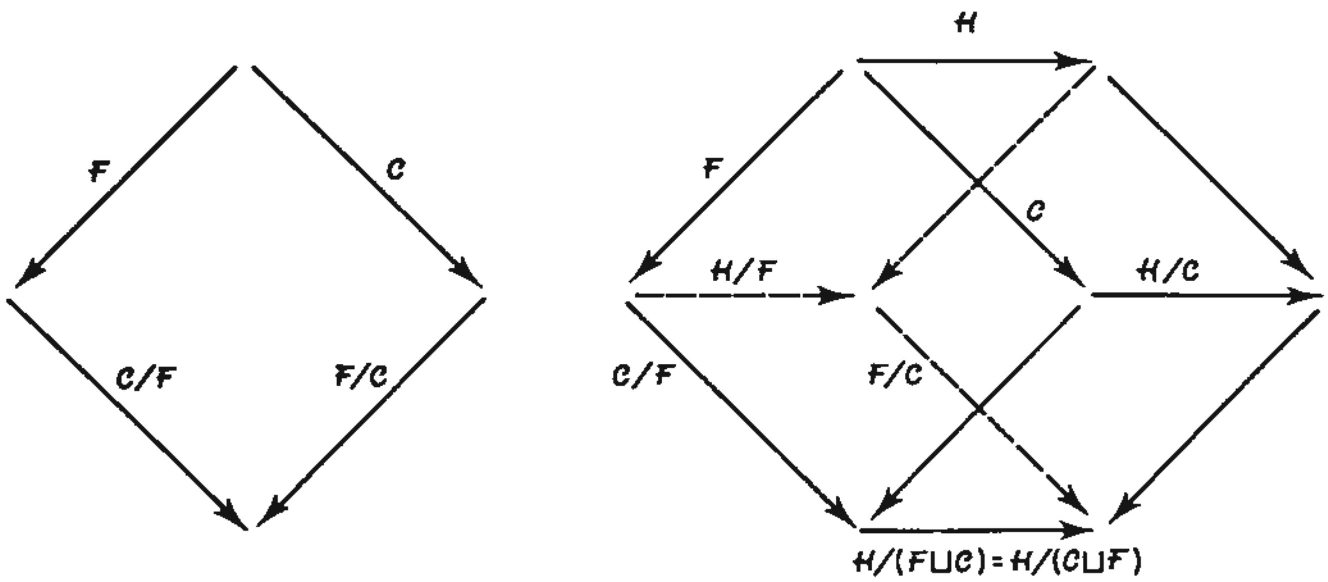


Figure 1

Similarly, an embedding relation can be defined by stating : $\rho \sqsubseteq \sigma$ iff $\exists\tau.\ \rho\tau \equiv \sigma$. Fortunately, both embedding and permutation equivalence relations can be proved effective by extending the residual definition to reductions. (Remark that $\rho \equiv \sigma$ is not clearly decidable since the length of ρ and σ may differ because of the elimination of empty steps).

Suppose ρ and σ are two reductions which start at a same expression. Then the reduction residual σ/ρ of σ by ρ is a reduction starting at the end of ρ which is defined inductively on the sum of length of ρ and σ by :

$$0/\rho = 0$$
$$(\sigma F)/\rho = (\sigma/\rho)(F/(\rho/\sigma))$$

From this definition, some easy algebraic properties of residuals of reduction can be shown. For instance :

$(\sigma\tau)/\rho=(\sigma/\rho)(\tau/(\rho/\sigma))$

$\rho/(\sigma\tau)=(\rho/\sigma)/\tau$

$\rho/0=\rho$

A simpler way of considering residuals of reductions is to iteratively apply the square diagram of figure 1. Thus one gets figure 2.

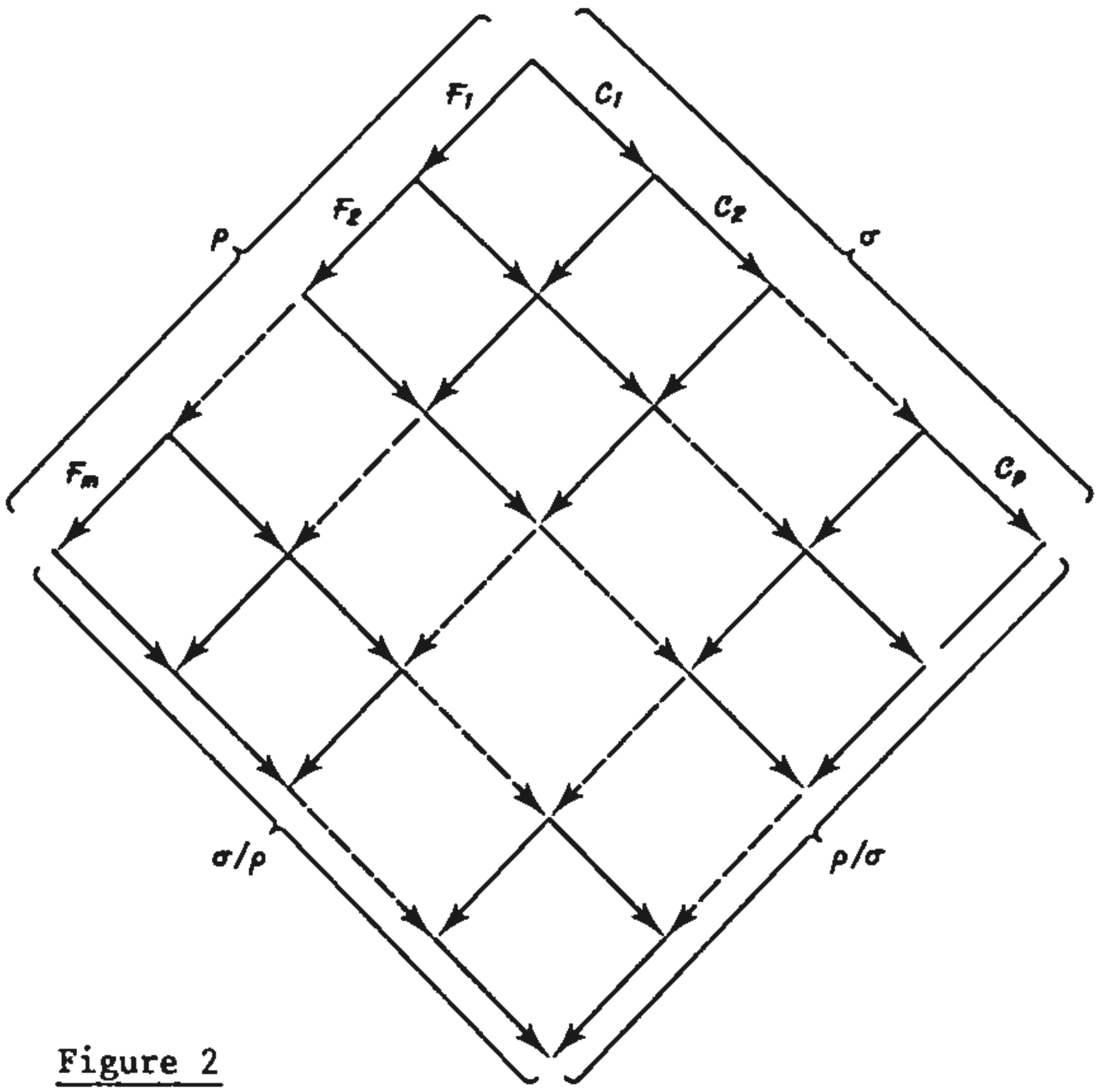


Figure 2

Now, we extend obviously the parallel moves lemma. If ρ and σ start at the same expression, let $\rho\sqcup\sigma=\rho(\sigma/\rho)$. Then :

LEMMA 3.2. Let ρ and σ be two reductions starting at M. Then

1) $\rho\sqcup\sigma$ and $\sigma\sqcup\rho$ end at the same expression,
2) $\tau/(\rho\sqcup\sigma)=\tau/(\sigma\sqcup\rho)$ for any other reduction τ starting also at M.

Proof : obvious iteration of lemma 3.1. □

Let $\emptyset^n_M$ be the reduction consisting of n empty steps starting at M, and $\emptyset^n=\emptyset\emptyset\ldots\emptyset$ (n times) when M is assumed from the context. Furthermore, let $|\rho|$ be the length of ρ (i.e. its number of steps). Now, one can state the easy decision procedures for relations $\equiv$ and $\sqsubseteq$.

LEMMA 3.3. Let ρ and σ be two reductions starting at M. Then

1) $\rho\equiv\sigma$ iff $\rho/\sigma=\emptyset^m$ and $\sigma/\rho=\emptyset^n$ with $m=|\rho|,n=|\sigma|$,

2) $\rho\sqsubseteq\sigma$ iff $\rho/\sigma=\emptyset^m$ with $m=|\rho|$.

Proof : by using properties of residuals of reductions and the previous lemma. Remark first that, when ρ,σ,τ are coinitial, one has $(\rho\sqcup\sigma)/\tau=(\rho/\tau)\sqcup(\sigma/\tau)$. Thus one can prove by induction on the definition of $\equiv$ that $\rho\equiv\sigma$ implies $\tau/\rho=\tau/\sigma$ for all τ. Thus $\rho\equiv\sigma$ implies $\rho/\sigma=\rho/\rho$ and $\sigma/\rho=\sigma/\sigma$. Therefore $\rho/\sigma=\emptyset^m$, $\sigma/\rho=\emptyset^n$ where $m=|\rho|$, $n=|\sigma|$.

Now, if $\rho\sqsubseteq\sigma$, there is τ such that $\rho\tau\equiv\sigma$. Hence $\rho\tau/\sigma\equiv\emptyset^p$ with $p=|\rho\tau|$, which implies $\rho/\sigma=\emptyset^n$ for some n. As $|\rho/\sigma|=|\rho|$, one has $n=|\rho|$.

Conversely, suppose $\rho/\sigma=\emptyset^m$ and $\sigma/\rho=\emptyset^n$ for some m,n. Then $\rho\equiv\rho\sqcup\sigma$ and $\sigma\equiv\sigma\sqcup\rho$ by elimination of empty steps. But $\rho\sqcup\sigma\equiv\sigma\sqcup\rho$ for any coinitial ρ and σ. And $\rho\equiv\sigma$. Similarly, when $\rho/\sigma=\emptyset^m$, let $\tau=\sigma/\rho$. Then $\rho\tau=\rho\sqcup\sigma\equiv\sigma\sqcup\rho\equiv\sigma$. □

Two equivalent reductions are reductions with the same initial and final expressions, but the converse may not be true. Take for instance $M=I(Ix)$ with $I=\lambda x.x$. There are two redexes R and S in M, but reductions $M\xrightarrow{R}Ix$ and $M\xrightarrow{S}Ix$ are not equivalent since $R/S\neq\emptyset$ and $S/R\neq\emptyset$. Similarly, if $\Delta=\lambda x.xx$ and R is the only redex in $\Delta\Delta$, one gets $0\not\equiv R\not\equiv RR\not\equiv RRR\ldots$ (However $0\sqsubseteq R\sqsubseteq RR\sqsubseteq RRR\ldots$). In figure 3, some more complicated example is treated.

Some easy properties of $\equiv$ and $\sqsubseteq$ can be proved by algebraic manipulations. We give a list of them (forgetting the appropriate conditions on initial and final expressions of reductions).

$\rho\sqcup\sigma\equiv\sigma\sqcup\rho$,

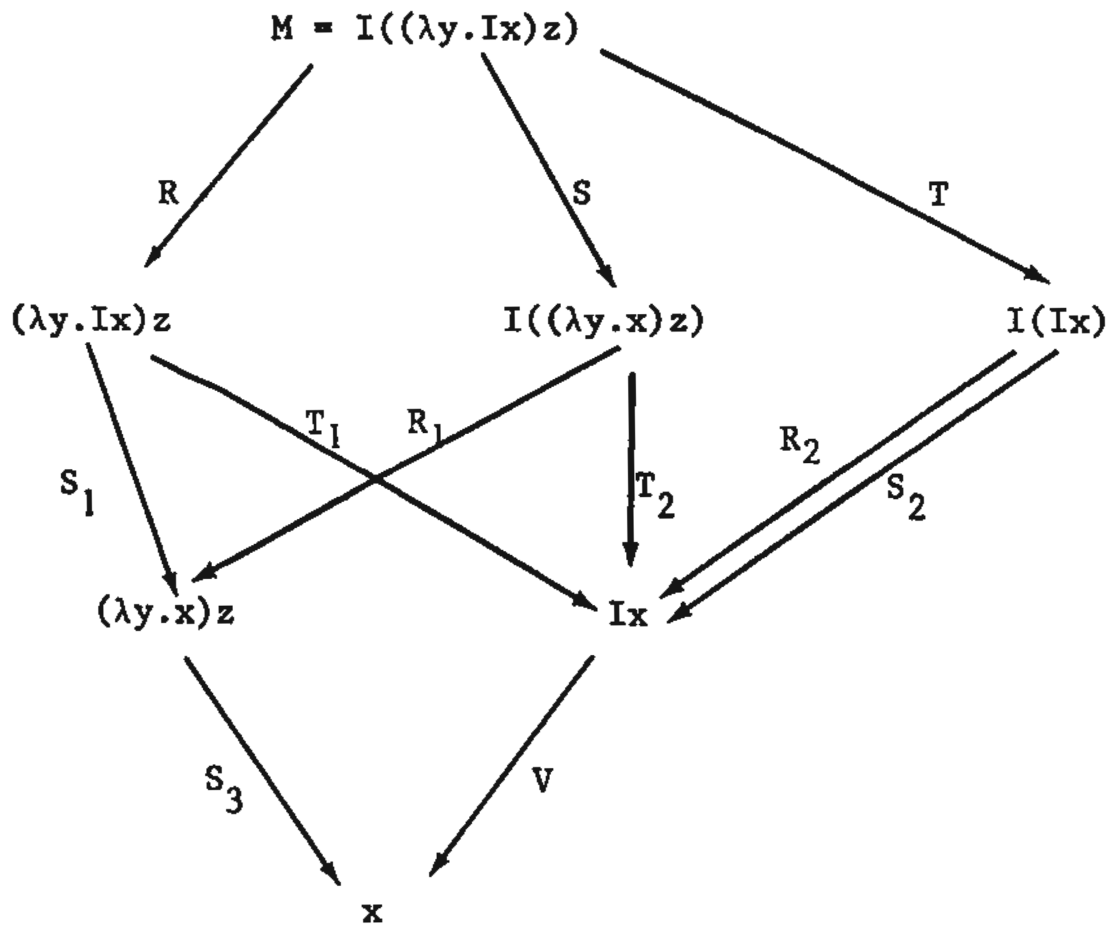
$\rho\equiv\sigma$ iff $\forall\tau.\ \tau/\rho=\tau/\sigma$,

$\rho\sigma\equiv\rho\tau$ iff $\sigma\equiv\tau$,

$\rho\equiv\sigma$ implies $\rho/\tau\equiv\sigma/\tau$,

$\rho\sqsubseteq\rho$,

$\rho\sqsubseteq\sigma\sqsubseteq\tau$ implies $\rho\sqsubseteq\tau$,



$$R_2 \not\equiv S_2$$

$$R\sqcup S = RS_1 \equiv SR_1 = S\sqcup R$$

$$R\sqcup T = RT_1 \equiv TR_2 = T\sqcup R$$

$$S\sqcup T = ST_2 \equiv TS_2 = T\sqcup S$$

$$RT_1 \not\equiv ST_2$$

<u>Figure 3</u>

$\rho\sqsubseteq\sigma\sqsubseteq\rho$ iff $\rho\equiv\sigma$,

$\rho\sqsubseteq\sigma$ implies $\rho/\tau\sqsubseteq\sigma/\tau$,

$\rho\sigma\sqsubseteq\rho\tau$ iff $\sigma\sqsubseteq\tau$,

$\rho\sqsubseteq\rho\sqcup\sigma$,

$\sigma\sqsubseteq\rho\sqcup\sigma$,

$\rho\sqsubseteq\tau$ and $\sigma\sqsubseteq\tau$ implies $\rho\sqcup\sigma\equiv\sigma\sqcup\rho\sqsubseteq\tau$.

Thus, $\sqsubseteq$ is a preorder with $\equiv$ as associated equivalence. Also $\equiv$ is a congruence for / and hence for $\sqcup$. Moreover $\sqsubseteq$ and $\equiv$ are left-simplifiable. Finally, $\sqsubseteq$ induces some semi sup-lattice structure on coinitial reductions quotiented by $\equiv$. Thus, we find in a very elementary way the computation lattice exhibited in recursive programs schemes by [Vu]. However, his proof was more complicated and holds only with certain restrictions, which permit to identify $\rho\sqsubseteq\sigma$ with $N\overset{*}{\rightarrow}P$ whenever $\rho:M\overset{*}{\rightarrow}N$ and $\sigma:M\overset{*}{\rightarrow}P$. All our study can be rephrased in category theory terminology [Pℓ]. We have two remarks.

First, it is not true that $\sqsubseteq$ induces some lattice property in the λ-calculus. Two coinitial reductions may not have some greatest lower bound. Take for instance $M=(\lambda x.K_a(xY))K_b$ with $K_a=\lambda x.a$, $K_b=\lambda x.b$, $Y=(\lambda x.f(xx))(\lambda x.f(xx))$. Then

$$\rho:M\rightarrow K_a(K_bY)\rightarrow K_ab$$

$$\sigma:M\rightarrow(\lambda x.a)K_b$$

have no glb.

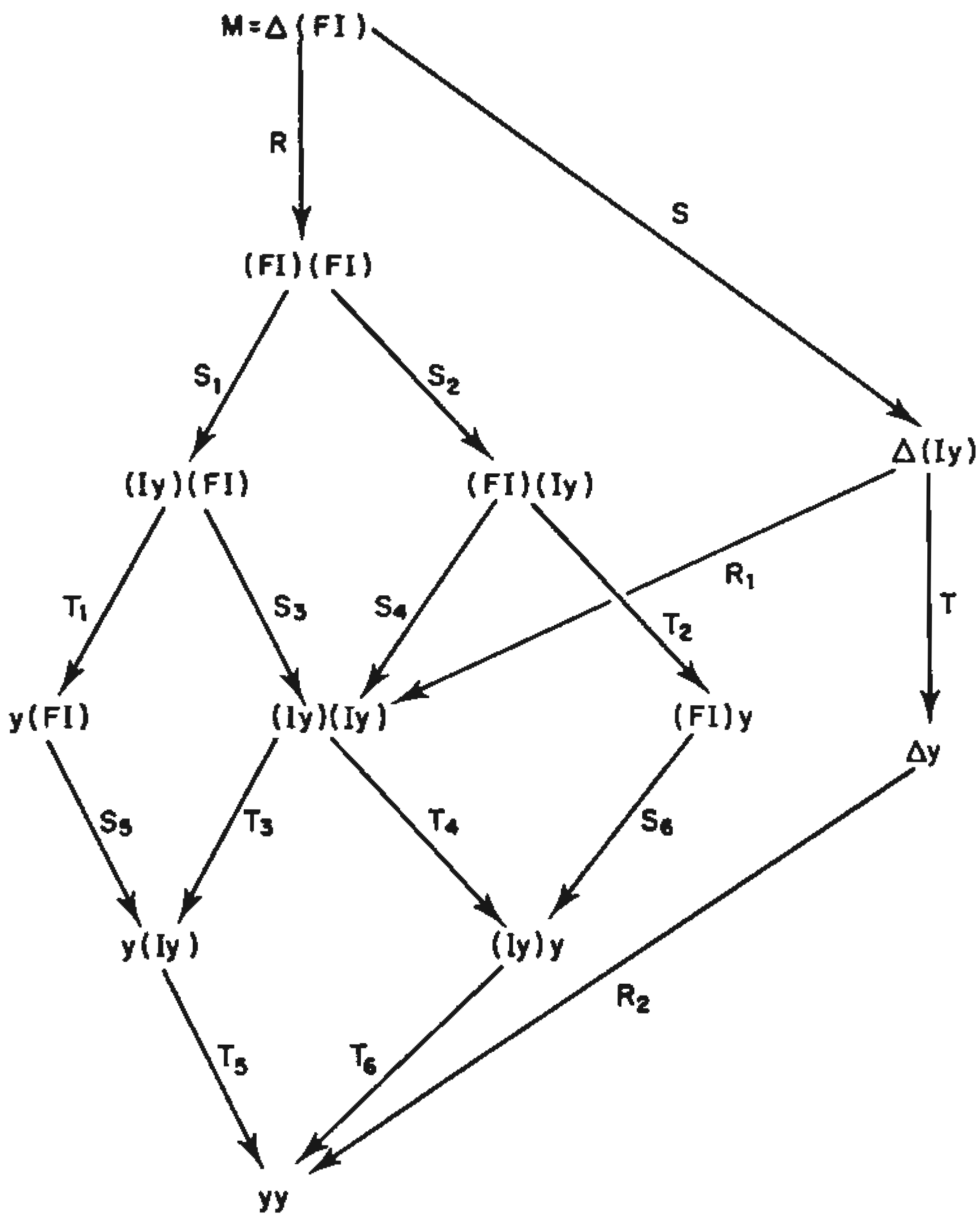
Secondly, the standardisation theorem in [Cu] says that any reduction may be reordered in a standard way. That is : to any reduction corresponds some outside-in and left-to-right reduction. This theorem can be improved. So, let $\rho=R_1R_2\ldots R_n$ be a <u>standard reduction</u> iff, for all i and j such that $1\le i<j\le n$, $R_j\in R_i'/R_iR_{i+1}\ldots R_{j-1}$ implies that R_i' is internal to R_i or disjoint to the right of R_i. Then, the new standardisation theorem says that, for every reduction ρ, there is a <u>unique</u> standard reduction σ such that $\rho\equiv\sigma$. The existence part of the proof follows from the usual proof of the standardisation theorem which consists in permuting reduction ρ. The unicity comes from remarking that, if $S\rho$ is a standard reduction and R is outside S or to the left of S, then $R/S\rho \neq \emptyset$. Thus standard reductions play the role of canonical representative in permutation-equivalence classes. (The situation is a bit analogous to the uniqueness of left derivations for a given parse tree in context-free formal languages.)

4. DUPLICATIONS OF REDEXES - REDEX FAMILIES :

In the introduction, the interest of looking at duplications of redexes modulo permutations was already mentioned. The example (i) is more completely described in figure 4. There seem to be three kind of redexes in this example. For R and S, it is very easy because all the R_i and S_i's are residuals of R and S. But, for the T case, the only feasible connexion between the T_i's can be exhibited by closing residuals downwards. For instance, the only way of connecting T_1 and T_2 is to say that $T_3 \in T_1/S_3$, $T_3 \in T/R_1$, $T_4 \in T/R_1$ and $T_4 \in T_2/S_4$. Thus, only using residuals, it seems possible to connect even redexes which are not in the initial expression. However, we need to be careful.

Remark first that T_1 and T_2 are not connected if the initial expression is (FI)(FI). Thus the wanted relation needs to be relativized to the initial expression. This can be achieved by considering redexes and their history, i.e. the reductions which give rise to them. In that order, we allow to also read reduction ρR as redex(occurrence) R with history ρ.

Secondly, we must not forget permutations when closing down the residual relation. Otherwise, we could connect R and S in the example of figure 3, which should not be related since, for instance, the Wadsworth's evaluator never contracts them simultaneously. Similarly, if $\Delta = \lambda x.xx$ and $R = \Delta\Delta$, then R with the empty history 0 must not be connected to R with history R.



$\Delta=\lambda x.xx$, $F=\lambda x.xy$, $I=\lambda x.x$

Figure 4

This leads us to the following definitions.

Definition 4.1. Redex S with history σ is a copy of redex R with history ρ, written $\rho R\leq\sigma S$, iff there is a reduction τ such that $\rho\tau\equiv\sigma$ and $S\in R/\tau$. Similarly, two redexes R and S with histories ρ and σ are in a same family, written $\rho R\simeq\sigma S$, iff $\rho R\leq\sigma S$ or $\sigma S\leq\rho R$ or there is some τT such that $\rho R\simeq\tau T\simeq\sigma S$.

Thus, in the example of figure 4, one gets $RS_1T_1 \simeq RS_2T_2$ since :

$$RS_1T_1 \leq RS_1S_3T_3 \geq ST \leq RS_2S_4T_4 \geq RS_2T_2 \; .$$

Similarly, one can check that $R \not\simeq S$ in figure 3. Now, we study properties of copies and families. Some first easy set of propositions can again be proved by some pure algebraic manipulations.

LEMMA 4.2. Let $\rho \equiv \rho'$ and $\sigma \equiv \sigma'$. Then

1) $\rho R \leq \sigma S$ iff $\rho' R \leq \sigma' S$,
2) $\rho R \simeq \sigma S$ iff $\rho' R \simeq \sigma' S$.

Proof : obvious since $\equiv$ is a congruence for composition. □

LEMMA 4.3. $\rho R \leq \sigma S$ iff $\rho \sqsubseteq \sigma$ and $S \in R/(\sigma/\rho)$. Thus $\leq$ is easily decidable.

Proof : As $\rho R \leq \sigma S$, there is τ such that $\rho\tau \equiv \sigma$ and $S \in R/\tau$. By definition $\rho \sqsubseteq \sigma$. Thus $\rho \sqcup \sigma \equiv \sigma$ and $\rho(\sigma/\rho) \equiv \rho\tau$. By left-cancellation $\sigma/\rho \equiv \tau$. Therefore $R/(\sigma/\rho) = R/\tau$ and $S \in R/(\sigma/\rho)$. Conversely, if $\rho \sqsubseteq \sigma$ and $S \in R/(\sigma/\rho)$, one takes $\tau = (\sigma/\rho)$. □

LEMMA 4.4. $\leq$ is a preorder. Namely,

1) $\rho R \leq \sigma S \leq \tau T$ implies $\rho R \leq \tau T$,
2) $\rho R \leq \sigma S \leq \rho R$ iff $\rho \equiv \sigma$ and $R = S$.

Proof : Since $\rho R \leq \sigma S \leq \tau T$, there are ρ' and σ' such that $\rho\rho' \equiv \sigma$, $\sigma\sigma' \equiv \tau$, $S \in R/\rho'$ and $T \in S/\sigma'$. Thus $\rho\rho'\sigma' \equiv \tau$ and $T \in R/(\rho'\sigma')$. Therefore $\rho R \leq \tau T$. Now suppose $\rho R \leq \sigma S \leq \rho R$. Then $\rho \sqsubseteq \sigma \sqsubseteq \rho$ and $S \in R/(\sigma/\rho)$ by previous lemma. Thus $\rho \equiv \sigma$ and $\sigma/\rho = \emptyset^n$ for $n = |\sigma|$. But $R/\emptyset^n = \{R\}$ and $S = R$. The converse is obvious. □

LEMMA 4.5. $\leq$ satisfies some interpolation and unicity properties:

1) If $\rho \sqsubseteq \sigma \sqsubseteq \tau$ and $\rho R \leq \tau$, there is some redex S such that $\rho R \leq \sigma S \leq \tau T$,

2) If $\rho R_1 \le \sigma S$ and $\rho R_2 \le \sigma S$, then $R_1 = R_2$ (same occurrences).

Proof : Let $\rho \sqsubseteq \sigma \sqsubseteq \tau$ and $\rho R \le \tau T$. Then, by definition, there are reductions ρ', σ', τ' such that $\rho\rho' \equiv \sigma$, $\sigma\sigma' \equiv \tau$, $\rho\tau' \equiv \tau$ and $T \in R/\tau'$. Thus $\rho\tau' \equiv \rho\rho'\sigma'$ and $\tau' \equiv \rho'\sigma'$ by left-cancellation. Thus $R/\tau' = R/(\rho'\sigma')$ and $T \in S/\sigma'$ for some S such that $S \in R/\rho'$. Therefore $\rho R \le \sigma S \le \tau T$.

Now, we remark first that the residual definition implies that there is at most one redex R such that $S \in R/\rho$ for given S and ρ. Thus, if $\rho R_1 \le \sigma S$ and $\rho R_2 \le \sigma S$, we have $S \in R_1/(\sigma/\rho)$ and $S \in R_2/(\sigma/\rho)$. Therefore $R_1 = R_2$. □

We turn now to the hard part of this paper, which is to show that the family relation is decidable. The trouble comes from the necessity of looking now inside λ-expressions and from not being able to go on with algebraic manipulations. But first of all there is an easy case when one considers families of redexes with an empty history.

LEMMA 4.6. $R \simeq \rho S$ iff $S \in R/\rho$.

Proof : First, if $S \in R/\rho$, then $R \le \rho S$ and $R \simeq \rho S$. Conversely, we use an induction on the recursive definition of $\simeq$. Thus, we assume that there is some τT such that $R \simeq \tau T$, $T \in R/\tau$ and either one of the following two cases. First $\tau T \le \rho S$. Then $R \le \rho S$ since $R \le \tau T$. And $S \in R/\rho$, since $\rho = \rho/0$. Otherwise $\rho S \le \tau T$. Then $\rho \sqsubseteq \tau$. Since $0 \sqsubseteq \rho \sqsubseteq \tau$, we get by interpolation $R \le \rho S' \le \tau T$ for some redex S'. But $S = S'$ by unicity. Therefore $R \le \rho S$ and $S \in R/\rho$. □

Roughly speaking, the decision procedure for the family relation is as follows : $\rho R \simeq \sigma S$ iff R and S are "created" in the same way along ρ and σ, when ρ and σ are standard reductions. The problem is to formalise creations of redexes. This is not easy in the λ-calculus. One way is to define a labelled λ-calculus (see [Le2]) following the idea of [Vu] for recursive programs schemes. Another way, considered here and in [Bel], introduces an extraction operation on reductions and is to characterize redexes which do not create R in ρR.

Let a context C[ ] be a λ-expression with some missing subexpression and C[M] be the λ-expression obtained by filling the hole by the expression M. Similarly, we may have contexts C[, , ...] with several (disjoint) holes. Let two reductions ρ and σ starting at M be disjoint iff they are internal to two disjoint subexpressions of M. This means that M=C[N,P] and $\rho:C[N,P] \overset{*}{\to} C[N',P]$, $\sigma:C[N,P] \overset{*}{\to} C[N,P']$.

Let the function part of redex R=(λx.M)N be the left subexpression (λx.M), and the argument part of R the right subexpression N.

Suppose now x is a free variable in M. Let M^x[, , ...] be the context corresponding to M without all free occurrences of x. Assume that the reduction $R\rho$ is such that ρ is internal to the $i^{\underline{th}}$ instance of the argument N of R=(λx.M)N in its contractum M[x\N]. Thus $R\rho$ is as in figure 5 (d). Let then the reduction $\rho//R$, read ρ parallelised by R, be the reduction defined inductively by :

$$0//R=0$$

$$(S\rho)//R=(S'/R)((\rho/F)//(R/S')) \text{ where } S\in S'/R \text{ and } F=S'/(RS).$$

Remark that, since $S\rho$ is in the $i^{\underline{th}}$ instance of the argument of R in its contractum, one has S' in the argument part of R, F disjoint from ρ, $R/S'=\{R_1\}$ and ρ/F in the $i^{\underline{th}}$ instance of the argument of R_1 in its contractum. (See again figure 5 (d)).

Now, we eliminate unnecessary steps of ρ for redex R with history ρ.

Definition 4.7. The extraction relation $\triangleright$ is the union of the four following relations :

1) $\rho RS \triangleright_1 \rho S'$ if $S \in S'/R$,

2) $\rho(R \sqcup \sigma) \triangleright_2 \rho\sigma$ if $|\sigma| \geq 1$ and R, σ are two disjoint reductions,

3) $\rho(R \sqcup \sigma) \triangleright_3 \rho\sigma$ if $|\sigma| \geq 1$ and σ is a reduction internal to the function part of R,

4) $\rho R\sigma \triangleright_4^i \rho\sigma'$ if $|\sigma| \geq 1$, σ is internal to the $i^{\underline{th}}$ instance of the argument of R in its contractum and $\sigma'/R = \sigma//R$.

This definition is summarized in figure 5. Remark that, in the last case, the reduction σ' is in the argument part of R. For instance, in the example of figure 4, one has :

$$RS_1S_3 \triangleright RS_2 \triangleright S,$$
$$S\ T\ R_2 \triangleright SR_1 \triangleright R,$$
$$RS_1T_1 \triangleright ST.$$

Let $\blacktriangleright$ denote the transitive closure of $\triangleright$, i.e. $\rho \blacktriangleright \sigma$ iff there is a (possibly empty) chain of extractions leading from ρ to σ. The key property is the following lemma.

LEMMA 4.8. $\blacktriangleright$ has the Church-Rosser property, i.e. if $\rho \triangleright \sigma$ and $\rho \triangleright \tau$, then $\sigma \triangleright \nu$ and $\tau \triangleright \nu$ for some reduction ν.

The proof is tedious, as usual for Church-Rosser properties, because of its number of cases, and is sketched in the appendix. It relies mainly on the following remark : if R and S are two distinct redexes in an expression M, and T is a redex such that $T \in T_1/(R/S)$ and $T \in T_2/(S/R)$, then there is some T' such that $T \in T'/(R \sqcup S) = T'/(S \sqcup R)$. Now comes the decision procedure for the family relation.

THEOREM 4.9. Let ρ and σ be two standard reductions. Then $\rho R \simeq \sigma S$ iff $\rho R \blacktriangleright \tau T \blacktriangleleft \sigma S$ for some τT.

$$\begin{tikzcd}[column sep=large]
 & {} & & & \\
{} \arrow[r, mapsto, "*", "\rho"'] & {} \arrow[u, dashed, "S'"] \arrow[r, "R"] & {} \arrow[r, "S"] & {} & S \in S'/R
\end{tikzcd}$$

Figure 5 (a)

$$\begin{tikzcd}[column sep=large]
{} \arrow[r, mapsto, "*", "\rho"'] & C[M,N] \arrow[r, dashed, "*", "\sigma"'] \arrow[d, "R"'] & C[M,N_n] \arrow[d, dashed, "R/\sigma"] \\
 & C[M',N] \arrow[r, "*", "\sigma/R"'] & C[M',N_n]
\end{tikzcd}$$

R in M and σ in N

Figure 5 (b)

$$\begin{tikzcd}[column sep=large]
{} \arrow[r, mapsto, "*", "\rho"'] & C[(\lambda x.M)N] \arrow[r, dashed, "*", "\sigma"'] \arrow[d, "R"] & C[(\lambda x.M_n)N] \arrow[d, dashed, "R/\sigma"] \\
 & C[M[x\backslash N]] \arrow[r, "*", "\sigma/R"'] & C[M_n[x\backslash N]]
\end{tikzcd}$$

σ in N

Figure 5 (c)

$$\begin{tikzcd}[column sep=large]
{} \arrow[r, mapsto, "*", "\rho"'] & C[(\lambda x.M)N] \arrow[r, dashed, "*", "\sigma'"'] \arrow[d, "R"] & C[(\lambda x.M)N_n] \arrow[d, dashed, "R/\sigma'"] \\
 & C[M[x\backslash N]] \arrow[r, dashed, "*", "\sigma'/R = \sigma/\!/R"'] \arrow[d, "\sigma"', "*"] & C[M[x\backslash N_n]] \\
 & C[M^x[N,\ldots N_n \ldots N]] \arrow[ur, dashed, "*"] &
\end{tikzcd}$$

σ in one instance of N in M $[x\backslash N]$

Figure 5 (d)

Proof : Remark first that $\rho R \rhd \tau T$ always implies $\rho R \simeq \tau T$. In the first three cases of the definition of $\rhd$, one gets obviously $\tau T \leq \rho R$. In the last case, if $\rho = \rho' R' \rho''$, then $\tau = \rho' \tau'$ and $\rho'' \nu \equiv \tau'/R'$ where $\nu = \tau'/(R'\rho'')$. Furthermore, $R/\nu = \{T'\}$ and $T' \in T/(R'/\tau')$. Thus $\rho R \leq \rho'(R' \sqcup \tau')T'$ and $\rho'(R' \sqcup \tau')T' \geq \tau T$. Therefore, when $\rho R \rhd \tau T$ and $\sigma S \rhd \tau T$, one gets $\rho R \simeq \sigma S$.

Conversely, it is enough to show that $\rho R \leq \sigma S$ implies $\rho R \rhd \tau T \lhd \sigma S$ for some τT, because of the Church-Rosser property of $\rhd$. For suppose ρ and σ are standard reductions and $\rho R \simeq \sigma S$. Then there is a chain of $\rho_i R_i$ such that $\rho_0 R_0 = \rho R$ and $\rho_n R_n = \sigma S$ and, for $1 \leq i \leq n$, either $\rho_{i-1} R_{i-1} \leq \rho_i R_i$ or $\rho_i R_i \leq \rho_{i-1} R_{i-1}$. By lemma 4.2, one can always assume that ρ_i is standard. We shall prove the existence of $\tau_i T_i$ for $1 \leq i \leq n$ such that $\rho_{i-1} R_{i-1} \unrhd \tau_i T_i \unlhd \rho_i R_i$. Then by Church-Rosser of $\rhd$, we may conclude that there is some τT such that $\rho R \unrhd \tau T \unlhd \sigma S$.

So let ρ and σ be standard reductions and $\rho R \leq \sigma S$. We use an induction on $|\sigma|$. If $\rho = 0$, then $\sigma/\rho = \sigma$ and $S \in R/\sigma$. Thus $\sigma S \unrhd \rho R$ and $\rho R \unrhd \rho R$. Let $\rho \neq 0$. Then, as $\rho R \leq \sigma S$ implies $\rho \sqsubseteq \sigma$, one cannot have $\sigma = 0$. Thus $\sigma = S'\sigma'$. Let too $\rho = R'\rho'$. If $R' = S'$, then $\rho' R \leq \sigma' S$, since $\sqsubseteq$ is left-cancellable. By induction, there is $\tau' T$ such that $\rho' R \unrhd \tau' T$ and $\sigma' S \unrhd \tau' T$. Thus, if $\tau = R'\tau'$, we get $\rho R \unrhd \tau T \unlhd \sigma S$. Suppose now $R' \neq S'$, which is in fact the only interesting case. Then, since $\rho \sqsubseteq \sigma$, R' cannot be external to S' or to its left. Otherwise $R'/\sigma \neq \emptyset$ which contradicts $\rho/\sigma = \emptyset^n$. Thus, since ρ is standard, there is a decomposition of $\rho = \rho_f \sqcup \rho_a \sqcup \rho_d$ such that ρ_f, ρ_a, ρ_d are standard reductions respectively internal to the function part of S', internal to the argument part of S' and disjoint from S'. (Remark that the definition of $\sqcup$ makes it associative). Now we have several cases with respect to the relative positions of R and of the residual S'' of S' by ρ.

1) If R is external to S'' or to the left of S''. Then one proves easily by induction on $|\rho|$ that $R \in T/\rho$ for some redex T external to S' or to the left of S'. Thus $\rho R \unrhd T$. But, since $\rho R \leq \sigma S$,

one gets $T \approx \sigma S$ and $S \in T/\sigma$ by lemma 4.5. Therefore $\sigma S \triangleright T$.

2) If R is internal to the function part of S", let then $\{S_f\} = S'/\rho_f$ and $\nu = (\rho_a \sqcup \rho_d)/\rho_f$. Then $S_f/\nu = \{S''\}$ and the reduction ν is internal to the argument of S_f or disjoint from S_f. Thus ν is disjoint from the function part of S_f. Thus there is a redex R_f in the function part of S_f such that $R_f/\nu = \{R\}$. Therefore $\rho R \triangleright \rho_f R_f$ and $\rho R \geq \rho_f R_f$. Since $\rho R \leq \sigma S$, one gets too $\rho_f R_f \leq \sigma S$. By lemma 4.3, $S \in R_f/(\sigma/\rho_f)$. So, if $(\rho_f R_f)/S' = \rho_f' R_f'$, then $\rho_f' R_f' \leq \sigma' S$ and $S' \rho_f' R_f' \triangleright \rho_f R_f$. Now, since ρ_f is in the function part of S' and ρ_f is standard, the reduction $\rho_f' = \rho_f/S'$ is also standard. Thus, by induction, there is $\tau' T'$ such that $\sigma' S \triangleright \tau' T'$ and $\rho_f' R_f' \triangleright \tau' T'$. Therefore $\sigma S \triangleright S' \tau' T'$ and $S' \rho_f' R_f' \triangleright S' \tau' T'$. By Church-Rosser for $\triangleright$, there is some τT such that $S' \tau' T' \triangleright \tau T$ and $\rho_f R_f \triangleright \tau T$. Thus $\sigma S \triangleright \tau T$ and $\rho R \triangleright \tau T$.

3) If R is disjoint from S" and to the right of S", there is again some R_d, disjoint from the residual S_d of S' by ρ_d, such that $\rho R \triangleright \rho_d R_d$ and $\rho R \geq \rho_d R_d$. Then one goes on as previously.

4) If R is in the argument part of S", then there is too some redex R_a in the argument part of the residual S_a of S' by ρ_a such that $\rho R \triangleright \rho_a R_a$ and $\rho R \geq \rho_a R_a$. We do as previously, but one gets trouble, since ρ_a/S' is no longer some standard reduction. However $\rho_a R_a \leq \sigma S$, since $\rho R \leq \sigma S$. Thus, by lemma 4.3, $S \in R_a/(\sigma/\rho_a)$. So, there is some redex $R_a'' \in R_a/S_a$ such that $\rho_a'' R_a'' \leq \sigma' S$, where $\rho_a'' = \rho_a/S'$. But R_a'' is in some instance of the argument of S_a in its contractum. Say the $i^{\underline{th}}$ instance. Then, since ρ_a'' is the union of disjoint reductions, each of them being internal to some instance of the argument of S' in its contractum, if ρ_a' is the part of ρ_a'' inside the $i^{\underline{th}}$ instance, then there is R_a' such that $\rho_a' R_a' \leq \rho_a'' R_a''$ and $S' \rho_a' R_a' \triangleright \rho_a R_a$. Since $\rho_a'' R_a'' \leq \sigma' S$, one has too $\rho_a' R_a' \leq \sigma' S$. But, now ρ_a' is standard, and we can go on by induction. So there is $\tau' T'$ such that $\sigma' S \triangleright \tau' T'$ and $\rho_a' R_a' \triangleright \tau' T'$. Thus $\sigma S \triangleright S' \tau' T'$ and $S' \rho_a' R_a' \triangleright S' \tau' T'$. By Church-Rosser of $\triangleright$, there is τT such that $S' \tau' T' \triangleright \tau T$ and $\rho_a R_a \triangleright \tau T$. Therefore $\sigma S \triangleright \tau T$ and $\rho R \triangleright \tau T$. $\square$

Remark that the previous theorem really gives a decision procedure since, when ρR and σS are given, there are effective ways of finding the standard reductions ρ' and σ' equivalent to ρ and σ by permutations (see [K1]) and furthermore the extraction relation $\trianglerighteq$ always ends at some normal form (because $\rho R \triangleright \tau T$ implies $|\rho|>|\tau|$).

Notice too that, if ρ is a standard reduction and $\rho R \triangleright \sigma S$, σ is also standard. This is obvious by considering the definition of $\triangleright$. Therefore, we can conclude that, in each redex family class, there is only one ρR such that ρ is standard and $|\rho|$ is minimum. Similarly, this ρR is the only one such that ρ is standard and ρR is in $\triangleright$-normal form. This canonical representative of the family class of, say, σS will be written $e^*(\sigma S)$ and named the <u>normal form by extractions</u>. Thus $\sigma_s S \trianglerighteq e^*(\sigma S)$ where σ_s is the standard reduction such that $\sigma_s \equiv \sigma$.

Finally, take the example of figure 4. Then there are three family classes with canonical representatives R,S,ST. In general, if $R \simeq \rho S$, then $R = e^*(\rho S)$ by lemma 4.6. This lemma has a nice generalisation, which will be fundamental in the rest of the paper. We first remark that the extraction relation can be done in some right to left order.

LEMMA 4.10. If $R\rho \triangleright \sigma$ and $\sigma \triangleright \tau$, there is some ν such that $\rho \triangleright \nu$ and $R\nu \triangleright \tau$.

The proof is similar to the proof of 4.6.

LEMMA 4.11. Let $\rho R = e^*(\sigma S)$. Then $\rho \sqsubseteq \sigma$ iff $\rho R \leq \sigma S$.

Proof : The if-part follows from 4.3. Suppose now that τ is the standard reduction such that $\tau \equiv \sigma$. We have $\tau S \trianglerighteq \rho R$ and $\rho \sqsubseteq \tau$, since $\rho R = e^*(\sigma S)$ and $\rho \sqsubseteq \sigma$. Moreover $\tau S = e^*(\tau S)$ is in normal form with respect to extractions.

By 4.2, it is sufficient to show $\rho R \leq \tau S$, since $\tau \equiv \sigma$. We work by induction on $|\tau|$. If $\tau=0$, then $\tau S \triangleright \rho R$ implies $\rho=0$ and $R=S$. Thus $\rho R \leq \tau S$. Let now $\tau=T\tau'$. By using the previous lemma, we have only two cases :

1) $\rho=T\rho'$ and $\rho' R=e^*(\tau' S)$. Then $\rho \sqsubseteq \tau$ implies $\rho' \sqsubseteq \tau'$ by left-cancellation. By induction $\rho' R \leq \tau' S$. And thus $\rho R \leq \tau S$.

2) $\rho' R'=e^*(\tau' S)$ and $T\rho' R' \triangleright \rho R$. Then by the definition of $\triangleright$, one always has $\rho' \sqsubseteq \rho/T$. Since $\rho \sqsubseteq \tau$, then $\rho/T \sqsubseteq \tau/T$. But $\tau/T=\emptyset\tau' \equiv \tau'$. Thus $\rho' \sqsubseteq \rho/T \sqsubseteq \tau'$. Therefore, by induction, $\rho' R' \leq \tau' S$. And, by interpolation, there is some redex S' such that $\rho' R' \leq (\rho/T)S' \leq \tau' S$. Now, we again look at the definition of $\triangleright$. Let $\rho''=\rho/(T\rho')$. Then ρ'' is always disjoint from R' and there is only one residual R'' of R' by ρ''. Furthermore $\rho R \leq (T \sqcup \rho)R''$. But, by 4.3, $S' \in R'/\rho''$. Thus $S'=R''$. And $\rho R \leq (T \sqcup \rho)S' \leq T\tau' S$. That is $\rho R \leq \tau S$. □

5. COMPLETE REDUCTIONS

First, we generalise the finite developments theorem. Let $[\rho R]$ be the family class of ρR, i.e. the equivalence class of ρR with respect to $\simeq$. Let $FAM(\rho)$ be the set of family classes of the redexes contracted in ρ. More exactly, if $\rho=F_1F_2\ldots F_n\ldots$,

$$FAM(\rho)=\{[F_1F_2\ldots F_{n-1}R_n] \mid R_n \in F_n,\ n\geq 1\}$$

Say that ρ is <u>relative</u> to X if $FAM(\rho)\subset X$. Similarly, a reduction ρ relative to X is a <u>development</u> of X if there is no redex R such that $[\rho R]\in X$. Then lemma 4.6 tells us that these definitions are exact extensions of the ones of §2.

THEOREM 5.1. (Generalised finite developments theorem). Let X be a finite set of family classes. Then :

1) there is no infinite reduction relative to X,

2) if ρ and σ are two developments of X, then $\rho\equiv\sigma$. (This implies that ρ and σ end at the same expression and $\tau/\rho=\tau/\sigma$ for for all reductions τ starting at the initial expression of ρ and σ.)

Proof : The finiteness part is proved by using a labelled λ-calculus (see [Le1] with a so-called bounded predicate. In that case, there is a strong normalisation property of this calculus. And it is straightforward to show that relative reductions to some finite set X can be embedded in such a calculus. (One has just to show that $S \in R/\rho$ implies that R and S have the same labels. Thus $\rho R = \sigma S$ also implies that R and S have the same labels). Now the second part of the theorem follows easily from noticing that, when ρ and σ are relative to X, also $\rho \sqcup \sigma$ and $\sigma \sqcup \rho$ are relative to X. □

Developments have nice properties with respect to the family relation. This follows from the two next remarks. First, when $\sigma S \triangleright \rho R$, one gets clearly $FAM(\rho) \subset FAM(\sigma)$. Secondly, also $FAM(\rho) \subset FAM(\sigma)$, when ρ is the standard reduction such that $\rho \equiv \sigma$. (This comes directly from the proof of the standardisation theorem). Thus, if $\rho R = e^*(\sigma S)$, the reduction ρ is relative to $FAM(\sigma)$.

Notice too that, when ρ is relative to X and σ is a development of X, one has $\rho \sqsubseteq \sigma$, since one always can extend ρ to some development $\rho\tau$ of X and $\rho\tau \equiv \sigma$ by the previous theorem. Therefore, one gets the two following lemmas.

LEMMA 5.2. Let ρ be a development of X. Then, for all redexes R with history ρ, one has $e^*(\rho R) \leq \rho R$.

Proof : Let $\sigma S = e^*(\rho R)$. Then σ is relative to $FAM(\rho)$. Therefore σ is relative to X, since $FAM(\rho) \subset X$. Thus $\sigma \sqsubseteq \rho$, since ρ is a development of X. By 4.11, $\sigma S \leq \rho R$. □

LEMMA 5.3. Let ρ be a development of X. Then, for any σS such that $\rho \sqsubseteq \sigma$, one has $[\sigma S] \notin FAM(\rho)$.

Proof : First, since ρ is a development of X, ρ is also a development of FAM(ρ). Thus there is no T such that $[\rho T]\in$FAM(ρ). Now, suppose $\rho \sqsubseteq \sigma$ and $[\sigma S]\in$FAM(ρ). Then $\rho=\rho_1 F \rho_2$ and there is some redex $R\in F$ such that $\rho_1 R\simeq\sigma S$. Let $\rho' R' = e^*(\rho_1 R)$. Then $e^*(\rho_1 R)=e^*(\sigma S)$, and ρ' is relative to FAM(ρ_1). Thus ρ' is relative to FAM(ρ) and $\rho' \sqsubseteq \rho$. By 4.11, $\rho' R' \leq \sigma S$. Therefore $\rho' R' \leq \rho T \leq \sigma S$ by interpolation for some redex T. Thus $\rho T \simeq \rho' R' \simeq \rho_1 R$ and $[\rho T]\in$FAM(ρ). Contradiction. □

Now, we consider complete reductions, which we will show as being particular developments. Let a reduction $F_1 F_2 \ldots F_n \ldots$ be <u>complete</u> iff, for every $n\geq 1$, $F_n \neq \emptyset$ is a maximum set of redexes such that, for all $R\in F_n$ and $S\in F_n$, $F_1 F_2 \ldots F_{n-1} R \simeq F_1 F_2 \ldots F_{n-1} S$.

Thus, at each step of a complete reduction, one non-empty family class is contracted.

LEMMA 5.4. Any complete reduction ρ is a development of FAM(ρ).

Proof : By induction on $|\rho|$. The case $\rho=0$ is obvious. Let $\rho=\sigma F$. By induction σ is a development of FAM(σ). Thus there is no ρR such that $[\rho R]\in$FAM(σ) by 5.3, since $\sigma \sqsubseteq \rho$. Suppose now that there is R and $S\in F$ such that $\rho R\simeq\sigma S$. Then $e^*(\sigma S)\leq\sigma S$ by 5.2. Let $\sigma' S' = e^*(\sigma S)$. Then $\sigma' \sqsubseteq \sigma \sqsubseteq \rho$ by 4.3. By 4.11, since also $\sigma' S' = e^*(\rho R)$, one gets $\sigma' S' \leq \rho R$. By interpolation, there is some T such that $\sigma' S' \leq \sigma T \leq \rho R$. Therefore $\sigma S \simeq \sigma T$ and $T\in F$, since ρ is complete. By 4.3, $R\in T/(\rho/\sigma)$. Therefore $R\in T/F$ and $R\in F/F$, since $T\in F$. Contradiction since $F/F=\emptyset$. □

LEMMA 5.5. Let ρ be a complete reduction. Then $|\rho| = \#$FAM(ρ), where $\#$FAM(ρ) is the number of elements in FAM(ρ).

Proof : Let $\rho=F_1F_2...F_n$. Then $|\rho|=n$. Let $1\le i<j\le n$ and $\rho_i=F_1F_2...F_{i-1}$, $\rho_j=F_1F_2...F_{j-1}$. Then, since ρ_i is complete and $\rho_i\sqsubseteq\rho_j$, one cannot have $\rho_i R\simeq\rho_j S$ for some $R\in F_i$ and $S\in F_j$, by using 5.4 and 5.3. Furthermore $F_i\ne\emptyset$ for every F_i. Thus $n=\#FAM(\rho)$. □

Further properties of complete reductions may be shown (see [Be1,Le2]. For instance, complete reductions make a sub-semi-lattice of the one of reductions. That is $\rho\sqcup\sigma$ is complete (up to some empty steps), when ρ and σ are complete reductions.

6. OPTIMAL REDUCTIONS

We come back to the problem discussed in the introduction. First, we show that call-by-need stategies are terminating. Let some reduction ρ be terminating iff its final expression is in normal form. Now, if $\rho=F_1F_2...F_n...$, let $R(\rho)$ be the set of redexes one of whose residuals is contracted in ρ. More exactly,

$$R(\rho)=\{R\,|\,R/F_1F_2...F_{i-1}\cap F_i\ne\emptyset,\ i\ge 1\}$$

Let some redex R in an expression M be needed iff, for all terminating reductions ρ starting at M, one has $R\in R(\rho)$. Let a reduction $\rho=F_1F_2...F_n...$ be a call-by-need reduction, iff there is at least one needed redex in every F_n, for $n\ge 1$.

We easily show that there is at least a needed redex in any expression M. Take R as being the leftmost-outermost redex of M. Let $\rho:M\xrightarrow{*}N$ be a reduction such that $R\notin R(\rho)$. Then $R/\rho=\{S\}$ and S is the leftmost-outermost redex in N. Thus R is needed, since, for all terminating ρ issued from M, one has $R\in R(\rho)$, since one must have $R/\rho=\emptyset$.

THEOREM 6.1. Let M have a normal form. Then any call-by-need reduction starting at M is eventually terminating.

Proof : By the standardisation theorem [Cu], since M has a normal form, the leftmost-outermost reduction eventually reaches the normal form. Let $d(M)$ be the length of the terminating leftmost-outermost reduction issued from M. Let

$$\sigma : M \xrightarrow{F_1} M_1 \xrightarrow{F_2} M_2 \ \ldots \xrightarrow{F_n} M_n \xrightarrow{F_{n+1}} \ldots$$

be a call-by-need reduction. We want to show that

$$d(M) > d(M_1) > d(M_2) > \ldots d(M_n) > \ldots$$

Thus, we will have $d(M_p)=o$ for some $p\geq 1$. That is M_p in normal form.

So let $\rho = R_1 R_2 \ldots R_n$ be the terminating leftmost-outermost reduction starting at M. Then $\rho / F_1 = C_1 C_2 \ldots C_n$ with, for every i, either $C_i = \emptyset$ or $C_i = \{S_i\}$ where S_i is leftmost-outermost (since residuals of leftmost-outermost redexs remain leftmost-outermost). Thus ρ/F_1 is the terminating leftmost-outermost reduction issued from M_1 (up to some empty steps). Now, as σ is call-by-need reduction, there is a needed redex $T \in F_1$. Therefore $T \in R(\rho)$. That is $R_i \in T/R_1 R_2 \ldots R_{i-1}$ for some i. Thus $C_i = \emptyset$ and $d(M) > d(M_1)$. Similarly $d(M_1) > d(M_2)$, etc... □

Now, we define some cost measures for reductions. As said in the introduction, some λ-evaluator using sharing allows to contract copies of a single redex in one unit of time. So let us say that F with history ρ is a <u>set of copies of a single redex</u> iff there is one redex S with history σ such that $\sigma S \leq \rho R$ for every $R \in F$. Natural reductions to consider are now the following c-complete reductions. Say that the reduction $\rho = F_1 F_2 \ldots F_n \ldots$ is <u>c-complete</u> iff, for all $n \geq 1$, the non-empty set F_n is a maximum set of copies of a single redex. Now, we shall assume that the cost of such a c-complete reduction satifies the equation : $\mathrm{cost}(\rho) = |\rho|$.

Furthermore, since all redexes in a set of copies of a single redex are in a same redex family and since we do not contract in one unit of time redexes which are not copies of one redex, we may assume for any reduction ρ: $\mathrm{cost}(\rho) \geq \# \mathrm{FAM}(\rho)$.

We first show that these two constraints one the cost measure are compatible and that c-complete reductions are effective, because they correspond exactly to complete reductions. (Remark that c-complete reductions are the right version of non-copying reductions in [O'D].)

LEMMA 6.2. A reduction is c-complete iff it is a complete reduction.

Proof : Suppose ρ is complete. Let R be redex with history ρ. Let F be the set of redexes S such that $\rho R \simeq \rho S$ and F' a maximum set such that $R \in F'$ and there is some σT such that $\sigma T \leq \rho S$ for all $S \in F'$.We want to show $F = F'$. First $F' \subset F$ since, for all $S \in F'$ and $S' \in F'$, we have $\sigma T \leq \rho S$ and $\sigma T \leq \rho S'$. Therefore $\rho S \simeq \rho S'$. Now by 5.4 and 5.2, since ρ is a complete reduction, ρ is a development of FAM(ρ). Therefore $e^*(\rho S) \leq \rho S$ for every $S \in F$. But, for all $S \in F$ and $S' \in F$, since $\rho S \simeq \rho S'$, we have $e^*(\rho S) = e^*(\rho S')$. Therefore F is a set of copies of a single redex. Since $R \in F$, we thus get $F \subset F'$, since F' is maximum.

Now, we prove easily by induction on $|\rho|$ that a reduction ρ is complete iff it is c-complete. The case $\rho=0$ is obvious. Let now $\rho = \sigma F$. By induction σ is complete iff σ is c-complete. Now, if ρ is complete, by the first part of the proof, F is a maximum set of copies of a single redex. Therefore ρ is c-complete. Suppose now ρ c-complete. then σ is complete by induction and F is one family class, by again the first part of this proof. Thus ρ is complete. □

Therefore, we can speak only of complete reductions. And lemma 5.5 tells us that the discussion on the cost measure is consistent, since $|\rho| = \#\text{FAM}(\rho)$ for any complete reduction ρ. Notice too that, if $\rho = R_1 R_2 \ldots R_n \ldots$ is some non parallel reduction, we have too $\#\text{FAM}(\rho) \leq |\rho|$. Now, we prove the optimality theorem.

THEOREM 6.3. Any complete and call-by-need reduction reaches the normal form in an optimal cost.

Proof : Let ρ be a call-by-need and complete reduction. Let σ be a terminating reduction starting at the same expression as ρ. We first prove $FAM(\rho)\subset FAM(\sigma)$. The case $\rho=0$ is trivial. Let $\rho=\rho'F$. Then $FAM(\rho)\subset FAM(\rho')\cup\{[\rho'S]\}$ for any $S\in F$. By induction $FAM(\rho')\subset FAM(\sigma)$. But σ/ρ' is also terminating, since the final expression of σ is in normal form. Since there is some needed redex R in F, we know that $R\cap R(\sigma/\rho')\neq\emptyset$. Therefore $\sigma/\rho'=\sigma_1'F_2'\sigma_3'$ with $R/\sigma_1'\in F_2'$. But $\sigma=\sigma_1F_2\sigma_3$ with $\sigma_1'=\sigma_1/\rho'$, $F_2'=F_2/(\rho'/\sigma_1)$. Thus there is $R_2\in F_2$ and $R_2'\in F_2'$ satisfying :

$$\sigma_1R_2\leq(\sigma_1\sqcup\rho')R_2'\geq\rho'R.$$

Therefore $[\rho'R]\in FAM(\sigma)$ and $FAM(\rho)\subset FAM(\sigma)$. Thus, we may conclude that any terminating call-by-need reduction ρ reaches the normal form in $|\rho|$ steps such that :

$$|\rho|=\#FAM(\rho)\quad\text{(by 5.5)}$$

and

$$|\rho|=\mathrm{cost}(\rho)\leq\#FAM(\sigma)\leq\mathrm{cost}(\sigma)$$

for any other terminating reduction σ. □

COROLLARY 6.4. The leftmost-outermost complete reduction reaches the normal form in an optimal cost.

Other call-by-need strategies were studied in [Le2] and correspond to the safe computation rules in [Vu]. The converse of theorem 6.3 may also be proved, i.e., among the complete reductions, only the call-by-need ones are optimal.

7. CONCLUSION

It remains to design some λ-evaluator implementing our complete and call-by-need reductions. The trouble, as stressed in [Wa], comes from the bound variables. Our results were expressed in words of term rewriting systems and, thus, seem somewhat general. Therefore, some general theory of rewriting systems, including the λ-calculus case, would be welcomed. For instance in [Hu], the problem of defining call-by-need reductions is considered. However, the most important question is to find shared evaluators for rewriting systems with bound variables.

8. ACKNOWLEDGEMENTS

To Gérard Berry who contributed to the ideas of this paper. Thanks too to Gérard Huet who simplified some of the notations.

9. APPENDIX

Proof of lemma 4.8 : Cases will be number m.n where $\rho \triangleright_m \rho'$ and $\sigma \triangleright_n \sigma'$.

1) $RS \triangleright_1 S'$ with $S \in S'/R$.

1.2) $RS \triangleright_2 S''$ with $S''/R = \{S\}$ and S'' disjoint from R. Then $S'' = S'$.

1.3, 1.4) Similar cases.

Now, we notice that, in case $m \geq 2$ and $|\rho| \geq 1$, $|\sigma| \geq 1$, one has $R\rho\sigma \triangleright_m \rho'\sigma'$ iff $R\rho \triangleright_m \rho'$, σ is disjoint from $\rho'' = \rho'/(R\rho)$ and $(R/\rho)(\sigma/\rho'') \triangleright_m \sigma'$. Remark that, when m = 2 or m = 3, then $\rho = \rho'/R$ and $\rho'' = \emptyset^k$, $\sigma/\rho'' = \sigma$. Furthermore, we also remark that, when $\rho \triangleright \sigma$ and τ is disjoint from ρ, then $\rho/\tau \triangleright \sigma/\tau$ and σ is disjoint from τ. Therefore, it is enough to show that, when $RS\sigma \triangleright S'\sigma'$ and $RS\sigma \triangleright R\tau$, then there is τ' such that $S'\sigma' \triangleright \tau'$ and $R\tau \triangleright \tau'$.

2) $RS\sigma \triangleright_2 S'\sigma'$ with $S'\sigma'$ disjoint from R and $S\sigma = (S'\sigma')/R$. Therefore, the initial expression is of the form $C[R,M]$ and reductions $RS\sigma$ and $S'\sigma'$ are of the form :

$$\begin{tikzcd} C[R,M] \arrow[r, "S'"] \arrow[d, "R"] & C[R,M_1] \arrow[r, "*", "\sigma'"'] & C[R,N] \\ C[\overline{R},M] \arrow[r, "S"] & C[\overline{R},M_1] \arrow[r, "*", "\sigma"'] & C[\overline{R},N] \end{tikzcd}$$

Then, when $S\sigma \triangleright_n \tau$, since $S\sigma$ is internal to M, τ is also internal to M. Therefore one checks easily that $R\tau \triangleright \tau'$ and $S'\sigma' \triangleright \tau'$.

3) $RS\sigma \triangleright_3 S'\sigma'$ with $S'\sigma'$ in the function part of R and $S\sigma = (S'\sigma')/R$. Therefore the initial expression is of the form $C[(\lambda x.M)N]$ where $R = (\lambda x.M)N$. Reductions $RS\sigma$ and $S'\sigma'$ are of the form :

$$\begin{array}{ccccc}
C[(\lambda x.M)N] & \xrightarrow{S'} & C[(\lambda x.M_1)N] & \xrightarrow[\sigma']{*} & C[(\lambda x.M_n)N] \\
\Big\downarrow R & & & & \\
C[M[x\backslash N]] & \xrightarrow{S} & C[M_1[x\backslash N]] & \xrightarrow[\sigma]{*} & C[M_n[x\backslash N]]
\end{array}$$

We treat this case more algebraically.

3.1) $\sigma = T$ and $ST\triangleright_1 V$ because $T\in V/S$. But $\sigma' = T'$ and $T\in T'/(R/S')$. Since $\{S\} = S'/R$, one has $T\in V/(S'/R)$. Since $R \neq S'$, there is some V' such that $T\in V'/(R\sqcup S') = V'/(S'\sqcup R)$. Therefore $V\in V'/R$ and $T'\in V'/S'$. Hence $S'T'\triangleright_1 V'$. Furthermore, since $S'T'$ is in the function part of R and since $T'\in V'/S'$, we also have V' in the function part of R. Therefore $RV\triangleright_2 V'$.

3.2) $S\sigma\triangleright_2\tau$ because τ is disjoint from S and $\tau/S = \sigma$. Again, since $R \neq S$, there is τ' such that $\sigma' = \tau'/S'$ and $\tau = \tau'/R$. Now, one checks easily that τ' is disjoint from S' and in the function part of R. Therefore $S'\sigma'\triangleright_2\tau'$ and $R\tau\triangleright_3\tau'$.

3.3) $S\sigma\triangleright_3\tau$ because τ is in the function part of S and $\tau/S = \sigma$. This case is similar to the previous one.

3.4) $S\sigma\triangleright_4^i\tau$ because τ is in the $i^{\underline{th}}$ instance of the argument of S in its contractum and $\sigma/\!/S = \tau/S$. This case is similar to the previous ones in case $|\sigma| = 1$. Now, when $\sigma = \sigma_1\sigma_2$, we use the decomposition of $S\sigma_1\sigma_2\triangleright_4^i\tau_1\tau_2$ according to the remark preceeding case 2.

4) $RS\sigma\triangleright_4^i S'\sigma'$ with $S\sigma$ in the $i^{\underline{th}}$ instance of the argument of R in its contactiun and $(S'\sigma')/R = (S\sigma)/\!/R$. Then the initial expression of $RS\sigma$ is of the form $C[(\lambda x.M)N]$ and $R = (\lambda x.M)N$. Then $RS\sigma$ and $S'\sigma'$ are such that

$$
\begin{tikzcd}
C[(\lambda x.M)N] \arrow[r, "S'"] \arrow[d, "R"] & C[(\lambda x.M)N_1] \arrow[r, "*", "\sigma'"'] & C[(\lambda x.M)N_n] \\
C[M[x\backslash N]] \arrow[r, dashed] \arrow[d, "S"] & C[M[x\backslash N_1]] \arrow[r, dashed, "*"] & C[M[x\backslash N_n]] \\
C[M^x[N,\ldots N_1\ldots N]] \arrow[d, "*", "\sigma"'] & & \\
C[M^x[N,\ldots N_n\ldots N]] & &
\end{tikzcd}
$$

Therefore, if $S\sigma \rhd \tau$, τ is also in the $i^{\underline{th}}$ instance of N in $M[x\backslash N]$. The reductions $S\sigma$ and $S'\sigma'$ are isomorphic reductions inside M. Thus there is τ' such that $R\tau \rhd_4^i \tau'$ and $S'\sigma' \rhd \tau'$. $\square$

FOOTNOTE

1. For example, $F_1F_2\ldots F_n$ will denote the parallel reduction $M \xrightarrow{*} M_n$ above.

10. REFERENCES

[Be1] Berry G., Lévy J-J (1979). "Minimal and optimal computations of recursive programs", J. Assoc. Comp. Machinery, vol. 26, N° 1.

[Be2] Berry G., Lévy J-J (1979). Letter to the Editor, SIGACT News (July).

[Cu] Curry H.B., Feys R. (1958). "Combinatory Logic", vol. 1, North-Holland.

[Do] Downey P.J, Sethi R. (1976). "Correct computation rules for recursive languages", SIAM Journal on Computing, vol. 5, N° 3.

[Hu] Huet G., Lévy J-J. (1979), "Call-by-need Computations in non-ambiguous linear term rewriting systems", IRIA-LABORIA report N° 359 (August).

[Le 1] Lévy J-J. (1976), "An algebraic interpretation of the λ-calculus and an application of a labelled λ-calculus", Theor. Comp. Sc., vol. 2, N° 1.

[Le2] Lévy J-J. (1978), "Réductions correctes et optimales dans le lambda-calcul", Thèse d'Etat, Université de Paris 7, (Janvier)

[O'D] O'Donnell M.J. (1977), "Computing in Systems described by Equations", Lectures Notes in Comp. Sc. N° 58, Springer-Verlag.

[Pl] Plotkin G.D. (1978), Private Communication.

[St] Staples J. (1972), "Efficient Combinatory Reductions", to appear in Zeit. Math. Logik.

[Vu] Vuillemin J. (1974), "Correct and Optimal Implementation of Recursion in a simple programming language", J. Comp. Sys. Sc., vol. 9, N° 3.

[Wa] Wadsworth C.P. (1971), "Semantics and Pragmatics of the λ-calculus", Ph.D. Thesis, Oxford.

REDUCTION CYCLES IN COMBINATORY LOGIC

JAN WILLEM KLOP

Dirkje Mariastraat 20
Utrecht, Netherlands

Dedicated to H.B. Curry on the occasion of his 80th Birthday

INTRODUCTION

As is well-known, reduction in λ-calculus is closely related to reduction in Combinatory Logic. It is also well-known that there are some points of difference between the two. In these notes we investigate such a difference in behaviour, concerning *cyclic* reductions (or reduction 'loops' $M \to M' \to \ldots \to M$).

In the λ-calculus it is not hard to find terms M whose reduction graph $\mathcal{G}(M)$ (i.e. the set of all reducts of M structured by the reduction relation $\twoheadrightarrow$) is *finite* and *contains a cycle*. The easiest example of such a term is of course $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$. Another example is the term WWW where $W \equiv \lambda xy.xyy$. Its reduction graph is shown in Figure 1.

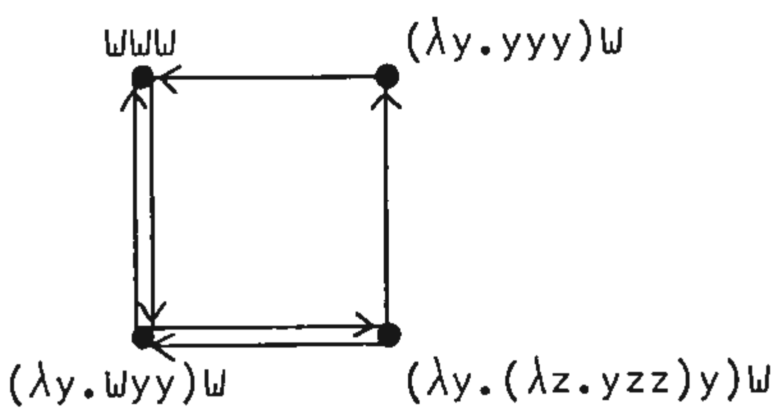


Fig. 1

A third example (which we will reconsider later) is $\omega(WI)$ where $\omega \equiv \lambda x.xx$, W is as above, and $I \equiv \lambda x.x$. (This term appears at p. 109 of Curry and Feys (1958).) See the reduction graph in

Figure 2, where $\omega' \equiv \lambda y.Iyy$. Some lines (reduction steps) in this figure are drawn heavily to indicate that those steps are reversible, i.e. lying on a reduction cycle.

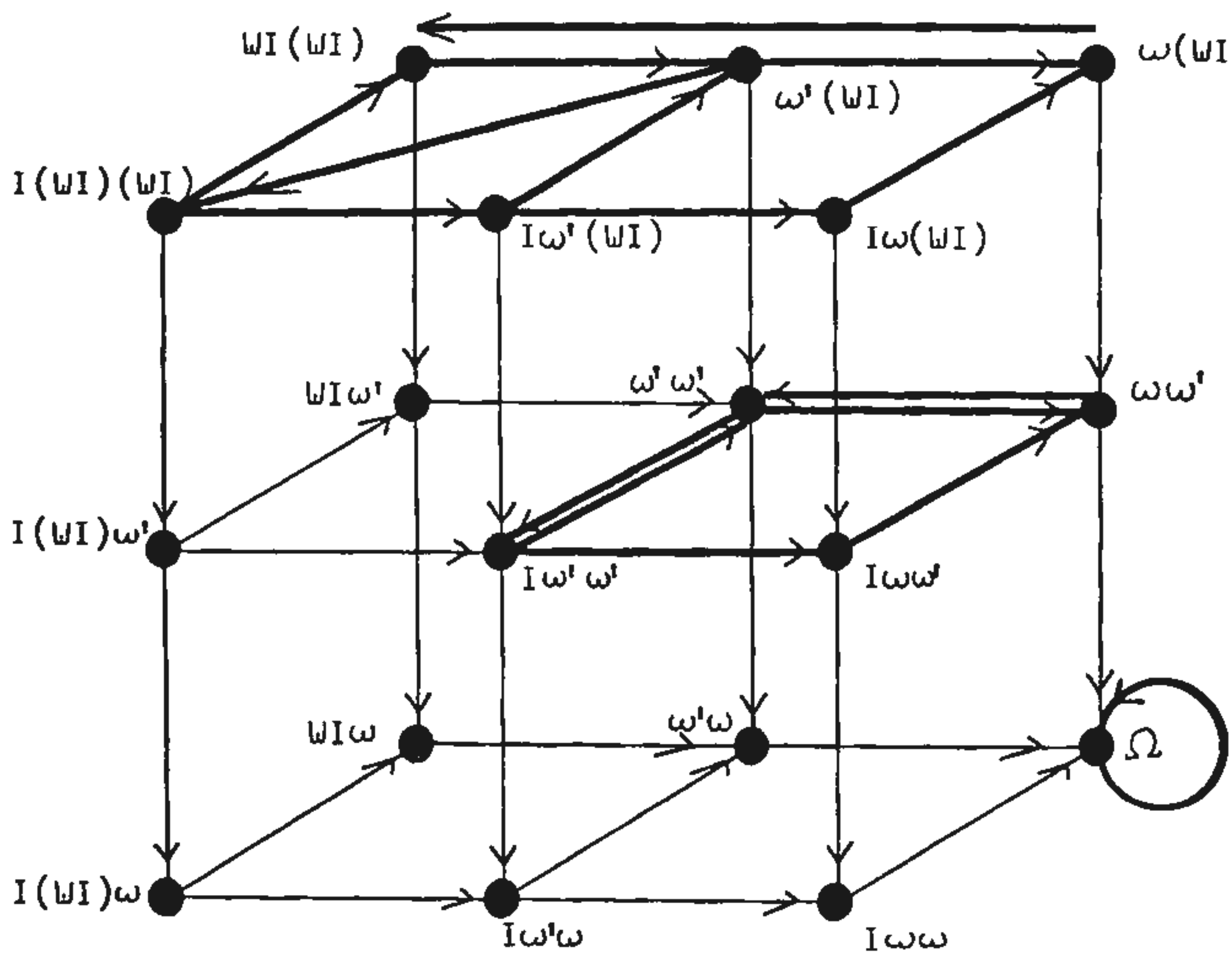


Fig. 2

As a fourth example, consider the term $M_3 \equiv NNNN$ where $N \equiv \lambda abc.cbaa$. Here $G(M_3)$ consists of exclusively one reduction cycle; see Figure 3.

It is easy to find terms M_n s.t. $G(M_n)$ is exactly a cycle of length n, for all $n \geq 1$. Let us call such a cycle a *pure cycle*.

Now the main theorem in this note says that there are no pure cycles in Combinatory Logic with the basic combinators S, K or I, K, S. Moreover, there are no terms M here such that $G(M)$ is finite and contains a cycle. (In fact the theorem says something more.) A corollary is that in CL based on {S,K} or {I,S,K} each

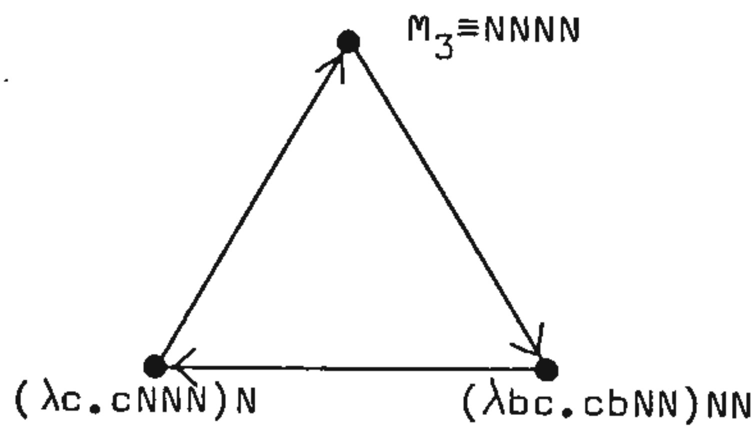


Fig. 3

term having only finitely many reducts, must have a normal form.

As an illustration consider the λ-term $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$, giving a pure cycle of length 1, and its CL-translation SII(SII). Apart from the cycle SII(SII) → I(SII)(I(SII)) → SII(I(SII)) → SII(SII) there are infinitely many reducts SII(SII) → I(SII)(I(SII)) → SII(I(SII)) → I(I(SII))(I(I(SII)) → ... → SII(I(I(I(SII)))) → ... → SII(I^n(SII)) → .... See figure 4 for G(SII(SII)). As will be clearer later, we remark that every cycle in this graph which does not intersect itself, is a *prime* cycle.

However, this result in the main theorem is basis-dependent. It holds for the two most used bases we mentioned; but it does not hold for all other less well-known bases for CL we have considered. We will give some examples to show this.

For the proof of the main theorem we introduce the concept of 'prime cycle', i.e. a cycle which cannot be dissected in smaller cycles. As a side remark we prove that terms lying on a prime cycle are unsolvable (equivalently: have no head normal form) and have moreover order 0 (that is, when substituted in a context they cannot interact with that context.)

Finally, we will state some conjectures and questions.

A word on motivation. Apart from that which we mentioned in the beginning of this introduction, we were also motivated to consider reduction cycles because they turned up in the study of

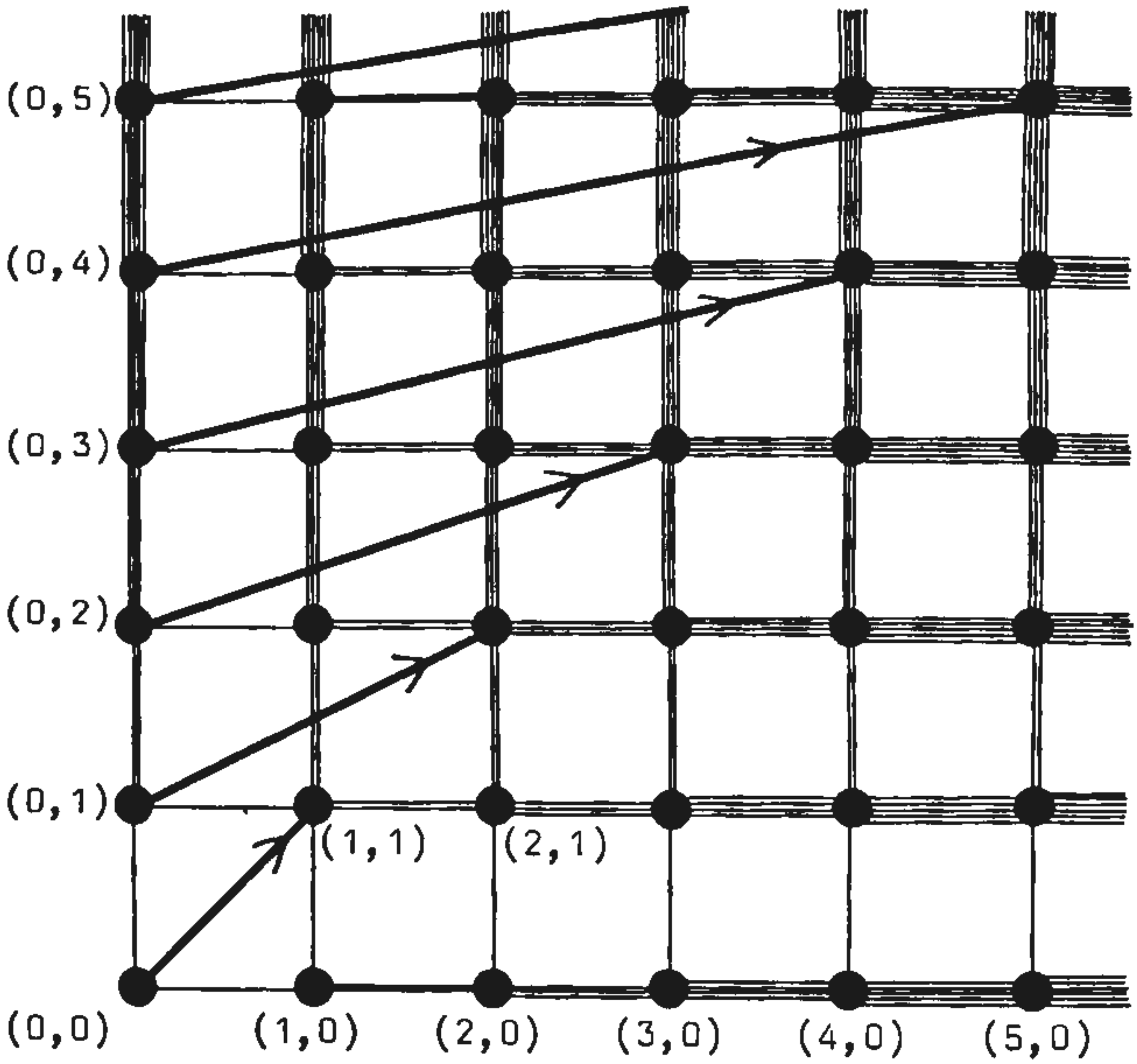


Fig. 4. Here (n,m) is short for $I^n(SII)(I^m(SII))$

Church-Rosser strategies, see Bergstra and Klop (1979), namely as the major stumbling block in the definition of such strategies. We will return to this aspect later, when mentioning a conjecture about cycles.

0. NOTATION

(i) About terms

(1) If $\vec{N}$ is a sequence of terms $N_1,\ldots,N_m$, then

$$M\vec{N} := MN_1N_2 \ldots N_m,$$
$$\overleftarrow{N}M := N_m(\ldots(N_2(N_1M))\ldots).$$

If $N_1 \equiv N_2 \equiv \ldots \equiv N_m$, we write, as is usual, N_1^mM instead of $\overleftarrow{N}M$.

(2) Here $\equiv$ denotes syntactical identity.

(3) $\subseteq$ denotes the subterm relation.

(4) If $N \subseteq M$, then M will be called a *context* of N. A context will be written as C[ ], and can be viewed as a CL-term containing exactly one hole □, viz. S(KS□S) or □KS. Contexts can be defined inductively:

$\begin{cases} 1) \ \square \text{ is a context (the trivial context).} \\ 2) \text{ if M is a CL-term and N is a context, then (NM) and (MN)} \\ \quad \text{are contexts.} \end{cases}$

The result of substituting N for □ in C[ ] is C[N].

(5) $C^n[\ ]$ is defined as follows for a context C[ ]:

$\begin{cases} C^1[\ ] \equiv C[\ ] \\ C^{n+1}[\ ] \equiv C[C^n[\ ]]. \end{cases}$

(ii) About reductions

(1) $\rightarrow$ is one-step reduction ($C[SABC] \rightarrow C[AC(BC)]$ etc.).

(2) $\twoheadrightarrow$ is the transitive reflexive closure of $\rightarrow$.

(3) $\underset{h}{\rightarrow}$ is a head-reduction step, i.e.

$$C[SABC] \underset{h}{\rightarrow} C[AC(BC)] \text{ for } C[\ ] \equiv \square\vec{N}, \text{ etc.}$$

(4) $\mathcal{R}$, $\mathcal{C}$, $\mathcal{P}$ will denote reduction sequences, e.g. $\mathcal{R} = M_0 \rightarrow M_1 \rightarrow \ldots \rightarrow M_n$, resp. cyclic reduction sequences resp. prime cyclic reduction sequences.

(5) * denotes concatenation of (appropriate) reduction sequences.

(6) If C[ ] is a context and $\mathcal{R} = M_0 \rightarrow \ldots \rightarrow M_n$, then $C[\mathcal{R}] = C[M_0] \rightarrow \ldots \rightarrow C[M_n]$.

In this section we will prove the main theorem. Everything in this section takes place in CL based on {(I), K, S}.

1.1 DEFINITION. (i) Suppose $N \subseteq M$ and $M \rightarrow M'$. Then we can trace the subterm N in this reduction step (by keeping track of brackets) in an obvious way; the resulting subterm(s) in M' are called the *descendant(s)* of N in M'. Note that the contractum of a redex R is not a descendant of R.

(ii) Let $C[R] \overset{R}{\rightarrow} C[R']$ be a reduction step in which the redex R is contracted. Let $A \subseteq C[R]$ and $B \subseteq C[R']$. Then B is a *quasi-descendant* of A iff (1) B is a descendant of A, or

(2) $A \equiv R$ and $B \equiv R'$.

(As always, we mean occurrences of subterms.)

1.2 DEFINITION *of 'subreduction'* $R_1 \le R_2$.

Let $R = M_0 \to \ldots \to M_k \to \ldots \to M_l \to \ldots\ M_n$ be a finite reduction sequence and let for some k, l such that $0 \le k < l \le n$ a sequence $L = N_k - \ldots - N_l$ be given such that

(i) $\forall k \le i \le l \quad N_i \subseteq M_i$ and

(ii) $\forall k \le i < l \quad N_{i+1}$ is a quasi-descendant of N_i.

L is not yet a reduction sequence, since it may contain trivial 'steps' (i.e. $N_{i+1} \equiv N_i$ for some i), but it determines a reduction sequence R' by omitting those trivial steps. Now we define such an R' to be a subreduction of R; notation $R' \le R$.

1.3 REMARK (i) $R_1 \le R_0 * R_1$

$R_1 \le R_1 * R_2$

$R_1 \le R_0 * R_1 * R_2$ for all R_0, R_1, R_2 such that the concatenations are defined.

(ii) $R \le C[R]$ for every context $C[\ ]$.

1.4 PROPOSITION. (i) $\le$ is a partial ordering on the set of finite reduction sequences;

(ii) if $<$ is the corresponding strict p.o. (i.e. $R' < R \Leftrightarrow R' \le R \ \&\ R' \ne R$) then

$$R' < R \Rightarrow \|R'\| < \|R\|$$

where $\|R\|$ is the number of symbols of R.

Proof. Obvious. □

1.5 DEFINITION. (i) A reduction *cycle* is a finite reduction $C = M_0 \to \ldots \to M_n \equiv M_0$.

(ii) A *prime cycle* is a reduction cycle C having no proper subcycles, i.e. $\forall R$:

$$R < C \Rightarrow R \text{ is not a cycle.}$$

E.g. if C is a cycle then $C^n = C * C * C * \ldots * C$ (n times) is not a prime cycle for $n \ge 2$. More general: a self-intersecting cycle is not prime.

Furthermore, if C is a cycle and $C[\ ]$ a non-trivial context, then $C[C]$ is not prime.

1.6 EXAMPLES. These examples serve also to refer to in the sequel. See also several examples in Venturini-Zilli (1979).

(i) The following abbreviations will be convenient:

$\Omega := SII(SII)$

$\Omega' := I(SII)(I(SII))$

$\Omega'' := SII(I(SII))$

$C_\Omega :=$ the cycle $\Omega \to \Omega' \to \Omega'' \to \Omega$. It is a prime cycle.

$I_A :=$ SKA. Note that I_A acts like I, since $SKAx \to Kx(Ax) \to x$.

The following example shows how a cycle (C) can contain (w.r.t. <) another cycle (C_Ω) in a less simple way than suggested by remark 1.3 (ii) and (iii).

$$
\begin{aligned}
C :\ & SI_{\Omega}I\ (SI_{\underline{\Omega}}I) \to \\
& SI_{\Omega}I\ (SI_{\underline{\Omega'}}I) \to \\
& I_{\Omega}(SI_{\underline{\Omega'}}I)(I(SI_{\utilde{\Omega'}}I)) \to \\
& I_{\Omega}(SI_{\underline{\Omega''}}I)(I(SI_{\utilde{\Omega'}}I)) \to \\
& K(SI_{\underline{\Omega''}}I)(\Omega'(SI_{\Omega''}I))(I(SI_{\utilde{\Omega'}}I)) \to \\
& SI_{\underline{\Omega''}}I(I(SI_{\utilde{\Omega'}}I)) \to \\
& SI_{\underline{\Omega''}}I(SI_{\utilde{\Omega'}}I) \to \\
& SI_{\underline{\Omega}}I(SI_{\utilde{\Omega'}}I) \to \\
& SI_{\underline{\Omega}}I(SI_{\utilde{\Omega''}}I) \to \\
& SI_{\Omega}I(SI_{\utilde{\Omega}}I).
\end{aligned}
$$

and $C > C_\Omega$ as can be seen by tracing the _ underlined subterms. (Also $C > C_\Omega$ by tracing the ~ underlined subterms.)

(ii) Another cycle C which is also not prime:

$$
\begin{aligned}
C :\ & S\Omega\Omega \to \\
& S\Omega'\Omega \to \\
& S\Omega'\Omega' \to \\
& S\Omega''\Omega' \to \\
& S\Omega''\Omega'' \to \\
& S\Omega\Omega'' \to \\
& S\Omega\Omega.
\end{aligned}
$$

(iii) The following cycle is prime.

Let $W \equiv SS(KI)$

$$
\begin{aligned}
C :\ WWW \equiv\ & SS(KI)WW \to \\
& SW(KIW)W \to \\
& WW(KIWW) \to \\
& WW(IW) \to \\
& WWW.
\end{aligned}
$$

(iv) Another prime cycle:

$$
\begin{aligned}
&(S(KI)\omega)(S(KI)\omega) \to \\
&KI(S(KI)\omega)(\omega(S(KI)\omega) \to \\
&I(\omega(S(KI)\omega)) \to \\
&\omega(S(KI)\omega) \to \\
&I(S(KI)\omega)(I(S(KI)\omega)) \to \\
&(S(KI)\omega)(I(S(KI)\omega)) \to \\
&(S(KI)\omega)(S(KI)\omega).
\end{aligned}
$$

(v) The next reduction cycle (which will illustrate the main lemma below) is obtained as follows. Let $L \equiv \lambda xy.x(yy)x$ and consider the term LLI. This term, suggested by Prof. C. Böhm, has an interesting reduction graph which is infinite and full of cycles (it is a cyclic equivalence class, see below); one of them is:

$$
\begin{aligned}
&LLI \to (\lambda y.L(yy)L)I \to L(II)L \to (\lambda y.II(yy)(II))L \to \\
&II(LL)(II) \to I(LL)(II) \to LL(II) \to LLI.
\end{aligned}
$$

This cycle of λ-terms is translated into CL by means of a well-known translation algorithm. Then we get the cycle of CL-terms

$$LLI \twoheadrightarrow S(BL)(KL)I \xrightarrow[(1)]{} BLI(KLI) \twoheadrightarrow LLI$$

and for the purpose of the illustration we have 'shifted' this cycle such that the displayed step (1) is the first one; after that we have standardized it.

The result is as follows. We use the abbreviations

$L \equiv SAK,$	$\omega \equiv SII,$
$A \equiv S(KS)B,$	$O \equiv K\omega LI,$
$B \equiv SC(K\omega),$	$L' \equiv KLI,$
$C \equiv S(KS)K,$	

and now we have the prime and standard reduction cycle

$$
\begin{aligned}
&S(BL)(KL)I \to \\
&BLI(KLI) \to \\
&CL(K\omega L)I(KLI) \to \\
&KSL(KL)(K\omega L)I(KLI) \to \\
&S(KL)(K\omega L)I(KLI) \to \\
&KLI(K\omega LI)(KLI) \to \\
&L(K\omega LI)(KLI) \to \\
&A(K\omega LI)(K(K\omega LI))(KLI) \to \\
&KSO(BO)(KO)(KLI) \to \\
&S(BO)(KO)(KLI) \to \\
&BO(KLI)(KO(KLI)) \to
\end{aligned}
$$

$CO(K\omega O)L'(KO(KLI)) \to$
$KSO(KO)(K\omega O)L'(KO(KLI)) \to$
$S(KO)(K\omega O)L'(KO(KLI)) \to$
$KOL'(K\omega OL')(KO(KLI)) \to$
$O(K\omega OL')(KO(KLI)) \to$
$I(K\omega OL')(KO(KLI)) \to$
$II(II)(K\omega OL')(KO(KLI)) \to$
$I(II)(K\omega OL')(KO(KLI)) \to$
$II(K\omega OL')(KO(KLI)) \to$
$I(K\omega OL')(KO(KLI)) \to$
$K\omega OL'(KO(KLI)) \to$
$\omega L'(KO(KLI)) \to$
$IL'(IL')(KO(KLI)) \to$
$L'(IL')(KO(KLI)) \to$
$L(IL')(KO(KLI)) \to$
$LL'(KO(KLI)) \to$
$AL'(KL')(KO(KLI)) \to$
$KSL'(BL')(KL')(KO(KLI)) \to$
$S(BL')(KL')(KO(KLI)) \to$
$S(BL)(KL')(KO(KLI)) \to$
$S(BL)(KL)(KO(KLI)) \to$
$S(BL)(KL)O \to$
$S(BL)(KL)(\omega I) \to$
$S(BL)(KL)(II(II)) \to$
$S(BL)(KL)(I(II)) \to$
$S(BL)(KL)(II) \to$
$S(BL)(KL)I$

1.7 PROPOSITION. Let C be a reduction cycle. Then there is a prime cycle P such that $P \leq C$.

Proof. Directly from the definitions and the fact that each descending chain $C > C' > C'' > \ldots$ must end by 1.4(ii). □

In the next proposition we need the concept 'standard'. Therefore:

1.8 INTERMEZZO: *standard reductions*. Standard reductions are defined as usual, see Barendregt (1980), Curry *et al.* (1958), Klop (1978), Lévy (1978). If R is a finite reduction, R_{st} will denote the standard reduction corresponding to R; that is the unique standard reduction equivalent (in the sense of Lévy) to R, see Lévy (1978), Klop (1978).

The following proposition is easily verified:

1.8.1 PROPOSITION. Let R be a standard reduction and $R' \leq R$. Then R' is also standard. □

1.8.2 DEFINITION. (i) Let MN be an applicative term. Then define $(MN)_l \equiv M$ and $(MN)_r \equiv N$.

(ii) We write $M \xrightarrow[l]{} N$ (or $M \xrightarrow[r]{} N$) if $M \to N$ and the head symbol I, S, K of the contracted redex is in $(M)_l$ (resp. $(M)_r$).

(iii) Let $R = M_0 \to \ldots \to M_n$ be a finite reduction and let R_{st} be:

$$M_0 \equiv M_0' \xrightarrow[l]{} M_1' \xrightarrow[l]{} \cdots \xrightarrow[l]{} M_j' \xrightarrow[r]{} \cdots \xrightarrow[r]{} M_m' \equiv M_n$$

(evidently since R_{st} is standard all the $\xrightarrow[l]{}$ steps precede the $\xrightarrow[r]{}$ steps.)

Then M_j' is called the *midterm* of R_{st}. Further: $(R_{st})_l \equiv M_0' \xrightarrow[l]{} \cdots \xrightarrow[l]{} M_j'$ and $(R_{st})_r \equiv M_j' \xrightarrow[r]{} \cdots \xrightarrow[r]{} M_m'$.

1.9 LEMMA. Let $G(M)$ contain a cycle C. Then $G(M)$ contains a cycle $C[P]$ for some context $C[\ ]$ and some standard prime cycle P.

Proof. Consider the standardization C_{st} of the given cycle C. By 1.7 C_{st} contains (w.r.t. $\leq$) some prime cycle P. By 1.8.1 P is moreover a standard reduction cycle. Take such a P and let $M_i \in C_{st} = M_0 \to \ldots \to M_n \equiv M_0$ be the first term through which P is passing (see Figure 5).

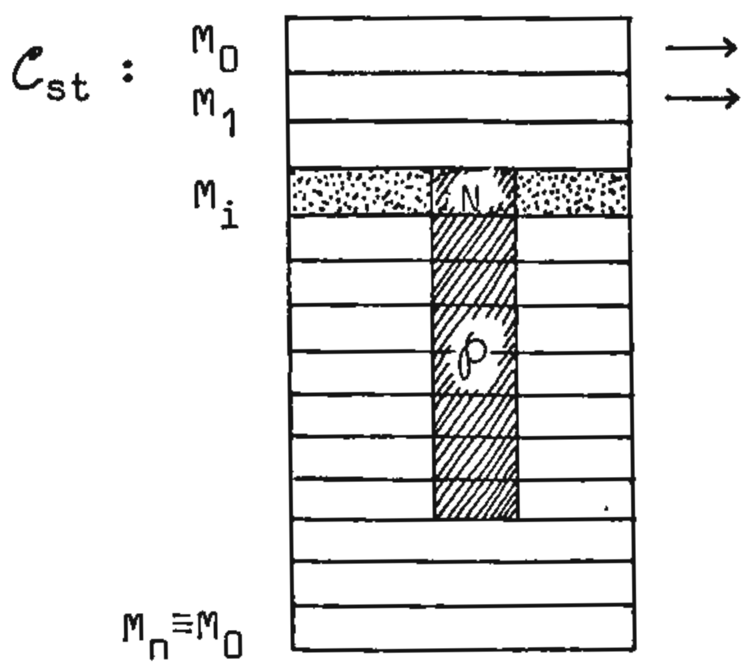


Fig. 5

Then obviously $M_i \equiv C[N]$ for some context $C[\ ]$ (the dotted context in Figure 5) and $N \in P$. Hence $G(M)$ contains also the cycle $C[P]$. $\square$

1.10 PROPOSITION. Let P be a prime cycle. Then P contains a step $SABC \to AC(BC)$ or $KAB \to A$ or $IA \to A$ for some A, B, C.

(Equivalently: one of the terms in P is a redex and P contains a head reduction step.)

Proof. (i) If there is an $M \in P$ s.t. M is a redex (SABC, KAB, or IA) then P must contain a head reduction step and we are through. (For suppose P does not contain a head step and let P contain a term SABC, say. Then P yields reduction cycles $A \twoheadrightarrow A$, $B \twoheadrightarrow B$, $C \twoheadrightarrow C$ of which at least one is not trivial (i.e. has positive length); hence P is not prime, contradiction.)

(ii) If P contains a term SAB, SA, or KA we find again a proper subcycle of P, contradiction.

(iii) So suppose P does not contain a term as in (i), (ii). Then every $M \in P$ is $SABC\vec{Q}$ or $KAB\vec{Q}$ or $IA\vec{Q}$ where $\vec{Q}$ is some *non-empty* sequence of arguments. To be precise:

$$
P \text{ is: } \begin{array}{l}
M_0 \equiv N_0 Q_{00} \cdots Q_{0m_0} \\
\big\downarrow R_0 \\
M_1 \equiv N_1 Q_{10} \cdots Q_{1m_1} \\
\big\downarrow R_1 \\
\vdots \\
\big\downarrow R_{n-1} \\
M_n \equiv M_0 \equiv N_n Q_{n0} \cdots Q_{nm_n} \equiv N_0 Q_{00} \cdots Q_{0m_0}
\end{array}
$$

where the N_i $(i \le n)$ are redexes, R_i $(i < n)$ are the redexes contracted in the successive steps of P, and $m_i > 0$ for all $i \le n$.

Now consider the terms Q_{im_i} $(i \le n)$ appearing at the end. Obviously the Q_{im_i} cannot be in the 'scope' of the contracted redex R_i, i.e. Q_{im_i} cannot be a <u>proper</u> subterm of R_i since the N_i are redexes, although $Q_{im_i} \supseteq R_i$ may be the case. According to this last possibility we can distinguish the two following cases:

(a) $R_i \subseteq Q_{im_i}$ for some i; then $Q_{0m_0} \to \ldots \to Q_{nm_n} \equiv Q_{0m_0}$ is a non-trivial proper subcycle of P, in contradiction to the fact that P is prime; or

(b) not (a); then deleting Q_{im_i} in M_i for all $i \le n$ yields a proper subcycle of P, again a contradiction. □

1.11 MAIN LEMMA. Let $C = SPQR \to PR(QR) \to \ldots \to SPQR$ be a cyclic standard reduction.

Suppose further that none of the steps in C is of the form $KAB \to A$ or $IA \to A$. (See remark (i) below.)

Then there is a non-trivial context $C^*[\;]$ such that

(i) $C^*[R] \twoheadrightarrow R$ and

(ii) $SPQC^{*n}[R] \twoheadrightarrow SPQC^{*n+1}[R]$ for all $n \ge 0$.

Remark. Before giving the proof, we note the following:

(i) Note that K- or I-redex contractions may occur in C as in the lemma, but only if such a redex is a proper subterm. I.e. if say $C[KAB] \to C[A]$ is a step in C, then $C[\;]$ is not the trivial context. (In fact one can show that a K- or I-contraction must occur in a cycle, see 3.2 below.)

(ii) Consider example 1.6(v), showing a cycle C as in the lemma. Here $R \equiv I$, and $C^*[\;] \equiv KO(KL\square)$ ($\equiv \overleftarrow{D}\square$ in the proof below).

Note that R (see the example) plays an essential role; i.e. in (i), (ii) above the R cannot be replaced by a variable x.

Proof of (i). By 1.8.2 (iii) the standard cycle C can be decomposed into the concatenation $C_l * C_r$ as follows (where it is evident that the midterm M_j must be of the form $SPQR_0^*$ since the later $\to_r$ steps cannot work in $(M_j)_1 \equiv SQP$ any more, so this part must already be in the final form).

$$
\begin{array}{l}
C = \underbrace{SPQR \underset{l}{\to} PR(QR) \underset{l}{\to} \ldots \underset{l}{\to} SPQR_0^*}_{C_l} \underbrace{\underset{r}{\to} SPQR_1^* \underset{r}{\to} \ldots \underset{r}{\to} SPQR_p^*}_{C_r} \equiv SPQR \\
\qquad\;\; ||| \qquad\quad ||| \qquad\qquad\quad ||| \qquad\quad ||| \qquad\qquad\quad ||| \\
\qquad\; M_0 \qquad\; M_1 \qquad\qquad\; M_j \qquad\; M_{j+1} \qquad\quad M_m \equiv M_0.
\end{array}
$$

Consider what happens to R in C_l.

<u>Claim</u>:

$$\forall k<j \quad (M_{k+1})_r \equiv (M_k)_r \quad \text{or} \tag{1}$$
$$\equiv D_k(M_k)_r \text{ for some term } D_k \tag{2}$$

The claim can easily be proved by induction on k, using the definition of C_1 and the hypotheses of the lemma, as follows: since there is no $KAB \to A$ step in C, the term $(M_{k+1})_r$ cannot be erased. Such an erasure would yield an $(M_{k+1})_r$ totally different (qua occurrence) from $(M_k)_r$ and everything would be spoiled; for a step $IA \to A$ the same holds, but also such steps do not occur in C. Finally, C_1 contains only $\underset{I}{\to}$ steps, so reductions cannot take place inside $(M_k)_r$. So the only possibility for $(M_k)_r$ to alter shape is in a step of the form $SC_kD_k(M_k)_r \to C_k(M_k)_r(D_k(M_k)_r)$. Thus the claim is proved.

Hence $(M_j)_r \equiv R_0^* \equiv \overleftarrow{D}R$ for some non-empty sequence $\vec{D} = Q, \ldots$. Now it follows immediately that $R_0^* \twoheadrightarrow R$, by definition of C_r. Hence the non-trivial context $C^*[\ ]$ is $\overleftarrow{D}\square$.

Proof of (ii). First we note that $C^{*n}[R] \twoheadrightarrow R$ for all $n \geq 0$. Now amend C_1 as follows:

$$
\begin{array}{llllll}
C_1 = M_0 & \xrightarrow{\quad h \quad} & M_1 & \longrightarrow \ \ldots \ \longrightarrow & M_j & \\
\quad \ \| \! | & & \| \! | & & \| \! | & \\
\ \ SPQR & & PR(QR) & & SPQ(\overleftarrow{D}R) & \\
\quad \ \downarrow & & \downarrow & & \downarrow & \\
R = SPQC^{*n}[R] & \underset{h}{\to} PC^{*n}[R](QC^{*n}[R]) & \twoheadrightarrow PR(QC^{*n}[R]) & \to \ldots \to & SPQ(\overleftarrow{D}C^{*n}[R]) & \\
 & & & & \| \! | & \\
 & & & & SPQC^{*n+1}[R] &
\end{array}
$$

Replace in all the M_k $(k \leq j)$ the subterm R occurring at the end of M_k by $C^{*n}[R]$. Then every step in C_1 is translated to a step in R (see figure above), unless the step in C_1 is of the form $SABC \underset{h}{\to} AC(BC)$, like the first step. In that case some steps have to be interpolated in R, using $C^{*n}[R] \twoheadrightarrow R$. Now R is the reduction we wanted to prove (ii). $\square$

Before formulating the main theorem we need a definition.

1.12 DEFINITION. $M \triangleq N \Leftrightarrow M \twoheadrightarrow N$ and $N \twoheadrightarrow M$. (M and N are 'cyclic equivalent'). Evidently, $\circeq$ is an equivalence relation.

1.13 THEOREM. In CL based on $\{S,K\}$ or $\{S,K,I\}$ the cardinality of a cyclic equivalence class is 1 or $\aleph_0$.

Before giving the proof, we mention the corollaries.

1.14 COROLLARIES. In CL based on $\{S,K\}$ or $\{S,K,I\}$:

(i) If $G(M)$ contains one cycle, it contains infinitely many.

(ii) Finite reduction graphs are acyclic.

(iii) There are no pure cycles.

(iv) If a term has only finitely many reducts, it must have a normal form.

(v) If $G(M)$ contains a cycle, it contains a non-cyclic infinite reduction. □

Proof of 1.13. Let $[M]_{\circeq} = \{N \mid M \circeq N\}$ be a cyclic equivalence class, and suppose that $\mathrm{card}([M]_{\circeq}) > 1$. So there is a cycle $C = M \twoheadrightarrow N \twoheadrightarrow M$ for some $N \not\equiv M$. We may suppose C is standard. By the proof of 1.9 there is a standard prime cycle $P \leq C$, and a context C[ ] and an L such that $C[L] \in C$ and L is the first term of P.

By 1.10 there are two cases.

<u>Case 1</u>. P contains a step $KAB \underset{h}{\rightarrow} A$ or $IA \underset{h}{\rightarrow} A$. Then since this step is part of a cycle, also $A \twoheadrightarrow KAB$ resp. $A \twoheadrightarrow IA$.

So now we have the reduction cycles in Figure 6, which shows that $[M]_{\circeq}$ is infinite.

The case of the $IA \underset{h}{\rightarrow} A$ step is analogous. Then for all n, $C[I^n A] \in [M]_{\circeq}$.

<u>Case 2</u>. P contains neither a step $KAB \underset{h}{\rightarrow} A$ nor a step $IA \underset{h}{\rightarrow} A$. Hence, by 1.10, P contains a step $SPQR \underset{h}{\rightarrow} PR(QR)$. Since P is standard this must be the first step, and so we are in the case of lemma 1.11. Now we have reductions as in Figure 7, and again $[M]_{\circeq}$ is infinite. □

1.15 REMARK. As a concluding remark to this section, we will prove that prime cyclic terms (i.e. terms lying on a prime cycle) are <u>unsolvable</u>, or equivalently, <u>have no head normal</u>

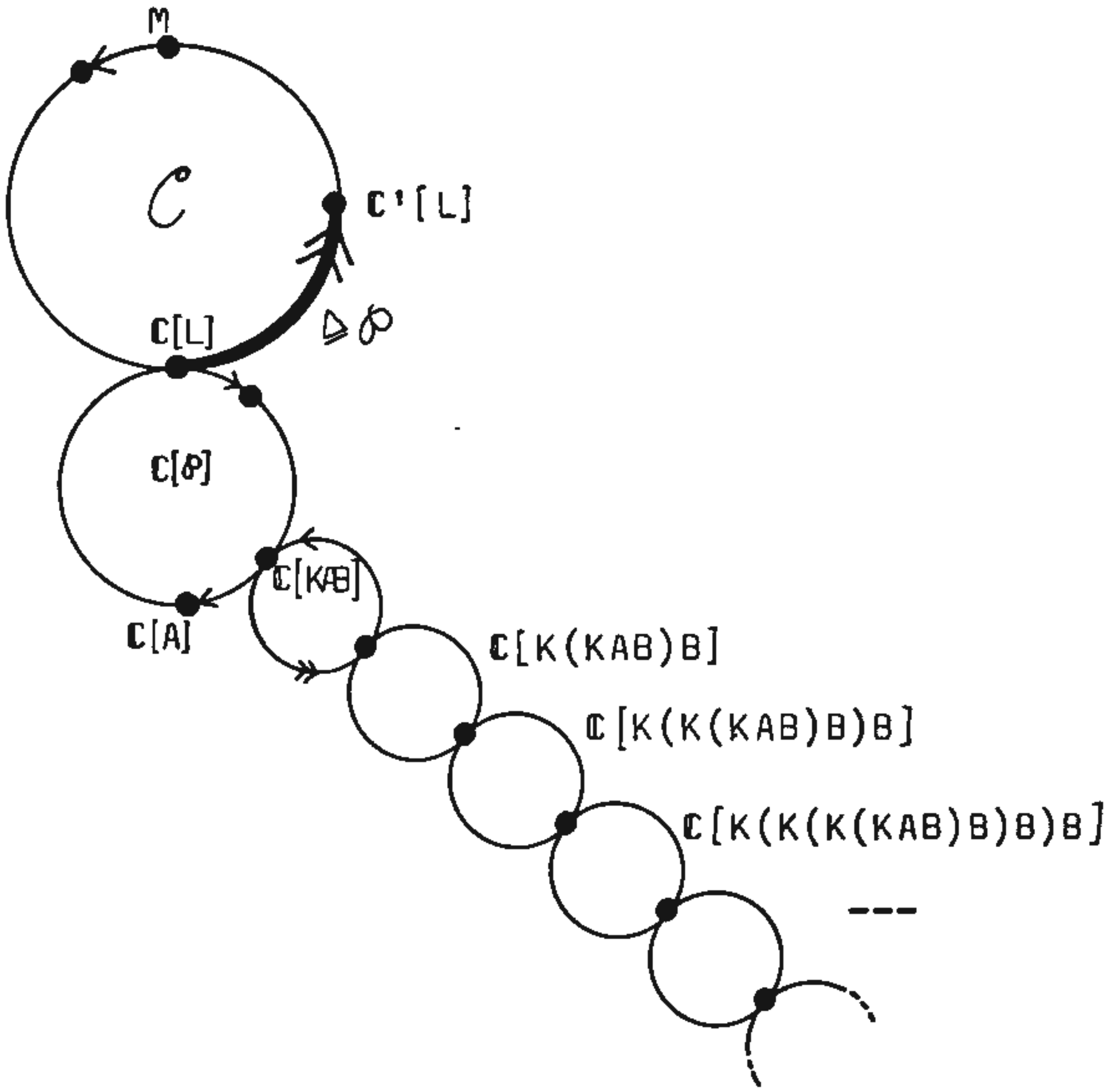


Fig. 6

form, or (since we work in CL) equivalently, have order 0 (so, when substituted in a context, they cannot interact with that context.)

This fact is a corollary of 1.10 and the Generalized head normalization theorem (see Barendregt 1980): 'Let R be an infinite reduction of M in which infinitely many head contraction steps occur. Then M has no head n.f.'

For, let P be a prime cycle. Then P contains by 1.10 a head step. Hence the infinite reduction $R = P{*}P{*}P{*} \ldots$ contains infinitely many head steps. Hence M has no head normal form. □

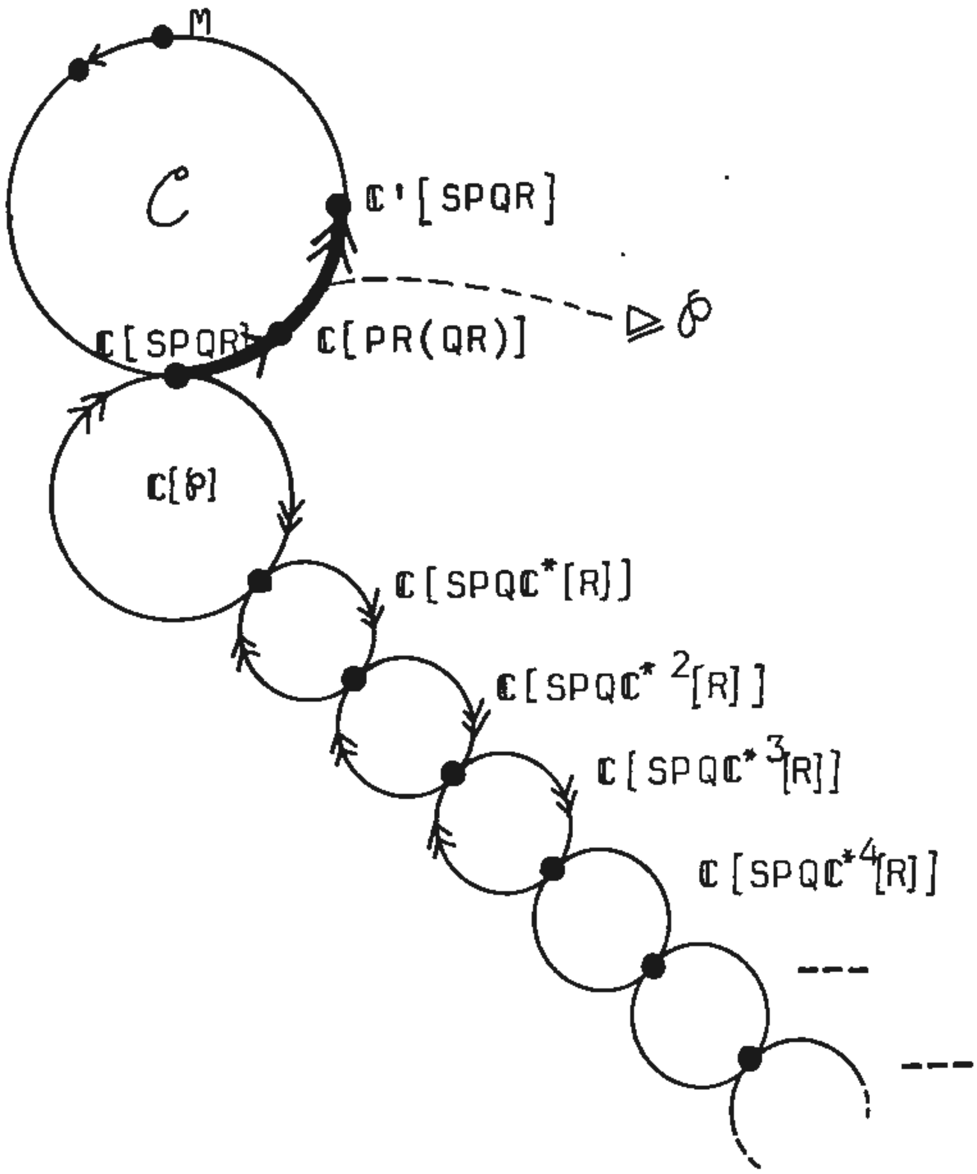


Fig. 7

2. SOME OTHER BASES OF CL.

In this section we consider some bases of CL (which occur in the literature) other than {S,K} and {S,K,I}. It will prove that the preceding theorem and most of its corollaries do not go through for them.

2.1 *The basis* {B,C.W,K} *for* CL.

This basis is mentioned in Curry *et al.* 1958. The primitive combinators have the reduction rules:

$$\begin{aligned} &Babc \to a(bc)\\ &Cabc \to acb\\ &Wab \to abb\\ &Kab \to a. \end{aligned}$$

Now the following is a pure cycle:

2.2 *The basis* {I,S,C,B} *for* CLI *(the* λI*-version of* CL*)*

Here also we have pure cycles: let $\omega^* := S(CI)(CI)$, then

$$
\begin{aligned}
&\omega^*\omega^* \to \\
&CI\omega^*(CI\omega^*) \to \\
&I(CI\omega^*)\omega^* \to \\
&CI\omega^*\omega^* \to \\
&I\omega^*\omega^* \to \\
&\omega^*\omega^* \to
\end{aligned}
$$

2.3 *The basis* {O,A,M,E} *of* CL

See Stenlund (1972), p. 21. The reduction rules are:

$$
\begin{aligned}
&Oxy \to y \\
&Axyzu \to xz(yzu) \\
&Mxyz \to x(yz) \\
&Exy \to yx.
\end{aligned}
$$

(This is Stenlund's notation. M is in fact B above.)

Abbreviation:

$I \equiv OO$

$D \equiv M(AII)E$. (Then $Dxy \twoheadrightarrow yxx$).

Here, although there are probably no <u>pure</u> cycles (?), we have finite reduction graphs containing a cycle: see Figure 8 (all reduction arrows in Figure 8 point in the downward direction).

2.4 *The basis* {I,J} *for* CLI,

where $Jabcd \to ab(adc)$, does not satisfy theorem 1.13 either.

Abbreviate

$U \equiv JI$	$W \equiv FQF$	$V \equiv FEG$
$E \equiv UI$	$Q \equiv FER$	$Z \equiv UC$
$F \equiv JE$	$R \equiv FTZ$	$C \equiv FFF$
$G \equiv UJ$	$T \equiv FEV$	$M \equiv WWW$.

(Now $Wab \twoheadrightarrow abb$.) Then $G(M)$ is finite (it contains several thousands of points) and contains cycles as in Figure 9 (all arrows pointing downward).

The <u>length</u> of the cycles in $G(M)$ is in the order of some dozens.

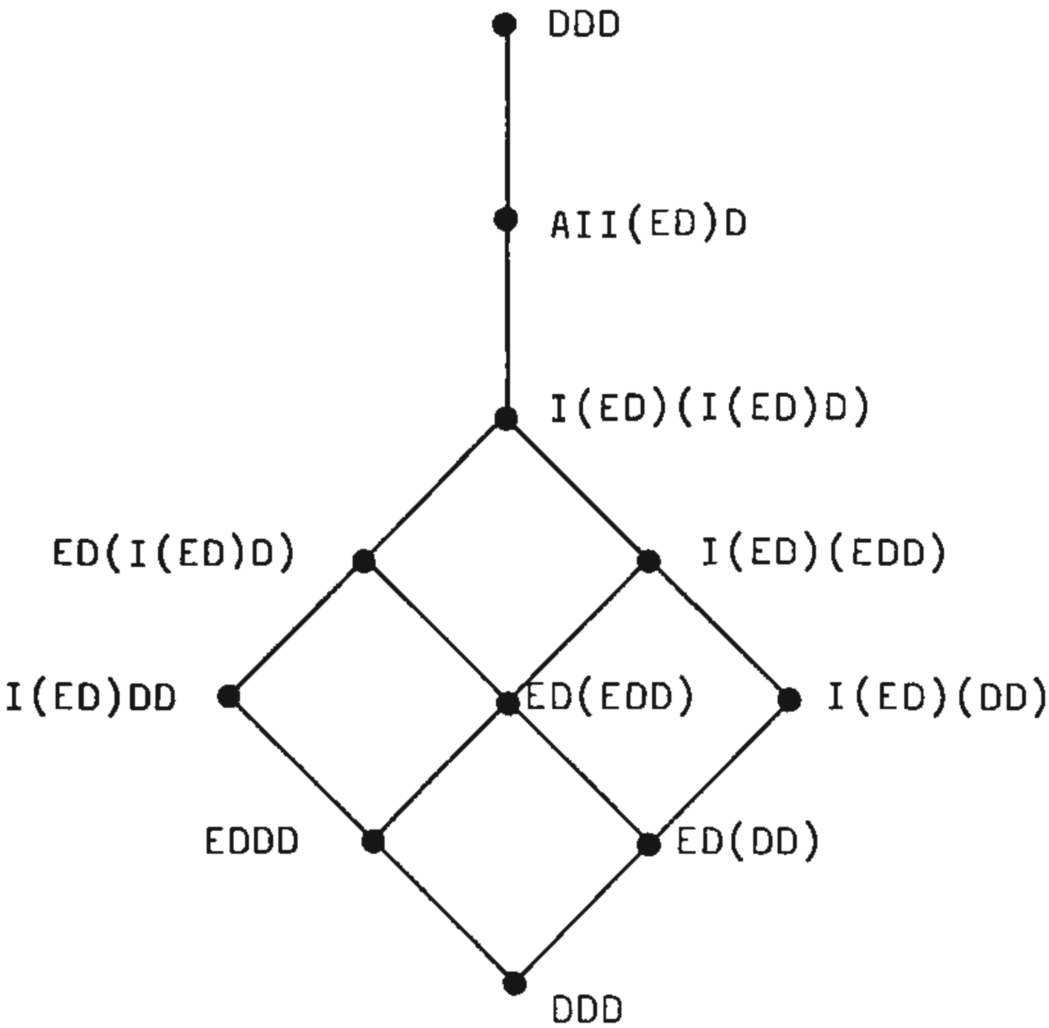


Fig. 8

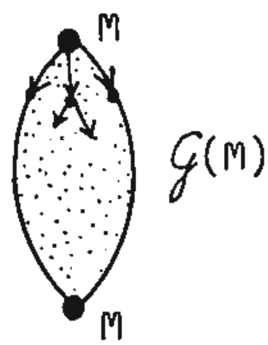


Fig. 9

QUESTION. Is there a simpler, short cycle in CLI with basis {I,J}?

3. SOME CONJECTURES AND QUESTIONS

3.1 CONJECTURE. In CL based on {(I),K,S}, head reduction is acyclic.

Remark: (i) In the case of a cycle C not containing a step KAB $\underset{h}{\to}$ A or IA $\underset{h}{\to}$ A (where $\underset{h}{\to}$ is head contraction), it is easy to prove that C cannot be a head reduction, using a bracket counting argument. (Namely, let r(M) be the number of <u>end</u> parentheses of M; for example, if $M \equiv (((SA)B)(KS))$ then $r(M) = 2$; and consider what happens with r(M) during C.)

(ii) Probably there is no $E \in CL_{\{(I),K,S\}}$ such that by <u>head</u> reduction $Eab \twoheadrightarrow ba$. If there is such an E, the conjecture 3.1 is easily refuted:

$$SEE(SEE) \underset{h}{\rightarrow} E(SEE)(E(SEE))$$

$$\underset{h}{\twoheadrightarrow} E(SEE)(SEE) \underset{h}{\twoheadrightarrow} SEE(SEE).$$

(Cf. the pure cycle in 2.2.)

Also there is probably no D in this system such that $Dx \underset{h}{\twoheadrightarrow} xx$. Otherwise DD refutes the conjecture.

3.2 REMARK. There is no cycle in $CL_{\{(I),K,S\}}$ in which every contracted redex is an S-redex.

Proof: as the proof that terms consisting exclusively of S's cannot cycle, in Bergstra and Klop (1979). □

3.3 QUESTION. As we have seen, there are no pure <u>cycles</u> in $CL_{\{(I),K,S\}}$. Are there 'pure <u>lines</u>' in this system? (See Figure 10.)

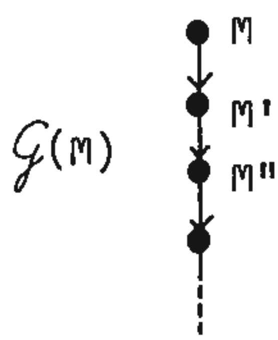


Fig. 10

3.4 QUESTION. Is there in λ-calculus a '<u>pure lemniscate</u>', i.e. a term M such that $G(M)$ is like Figure 11?

Fig. 11

3.5 QUESTION. If M is a λ-term, let M_{CL} denote the CL-term which is the result of applying some translation algorithm on M. As we have seen, $G(M)$ may be finite while $G(M_{CL})$ is infinite.

The reverse is also true; if M is a CL-term, let M_λ denote the result of replacing the basic combinators in M by the 'corresponding' λ-terms (e.g. $S_\lambda \equiv \lambda abc.ac(bc)$, $W_\lambda \equiv \lambda ab.abb$, etc.). Then $G(M)$ may be finite while $G(M_\lambda)$ is infinite. Example:

$$M := S(K\omega')(K\omega').$$

Then $M_\lambda \twoheadrightarrow \lambda x.\omega'_\lambda\omega'_\lambda$ and now it is not hard to find an $\omega' \in$ CL s.t. M is in normal form (hence $G(M)$ is finite) and $\lambda x.\omega'_\lambda\omega'_\lambda$ has infinitely many reducts.

So finiteness of reduction graphs is not an invariant property under translation, in both directions.

Question: is the property 'cycle-containing' of reduction graphs invariant under translation from λ to CL and vice versa? (From CL to λ: yes, trivially.)

3.6 CONJECTURES. These conjectures are meaningful in CL (with arbitrary basis) as well as for λ-calculus.

Let us call a cyclic equivalence class $[M]_{\underline{\underline{o}}}$ a *plane*. Reduction graphs can then be drawn 3-dimensionally like in figure 12; all irreversible reduction steps go downwards, the reversible ones stay in a plane.

Now let $N \in [M]_{\underline{\underline{o}}}$ be a term s.t. there is an irreversible step $N \to L$ for some L (i.e. $L \notin [M]_{\underline{\underline{o}}}$). Then we will say that we can *leave* the plane $[M]_{\underline{\underline{o}}}$ *directly* (in one step).

3.6.1 Conjecture. If a plane can be left somewhere, then it can be left directly at every point.

See Figure 2 in the introduction, where the conjecture is satisfied (we did not draw Figure 2 according to the plane convention above, though).

In case the plane consists of <u>one</u> cycle, it is not hard to prove the conjecture. So, for example, there is no λ- or CL-term M such that $G(M)$ is like Figure 13.

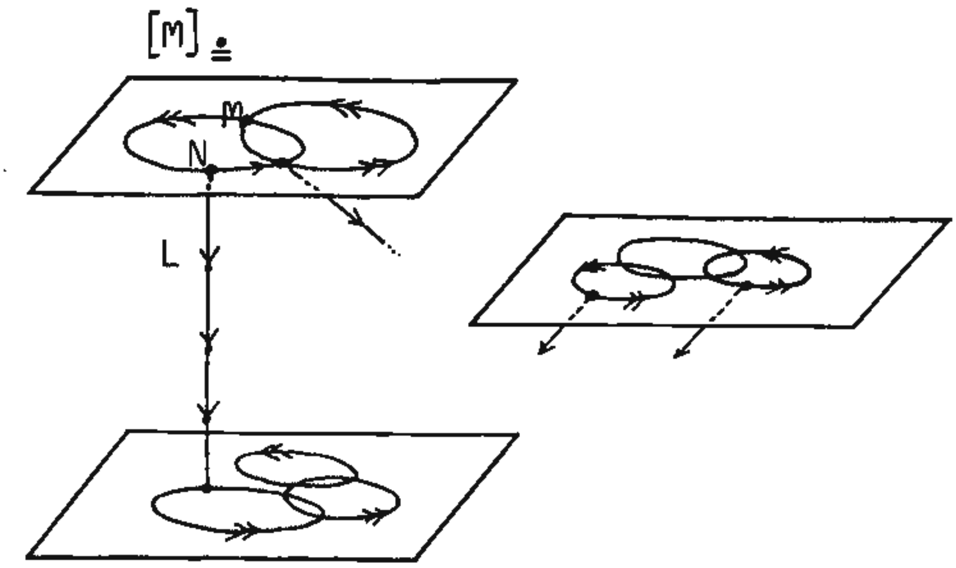


Fig. 12

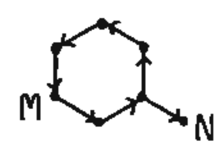


Fig. 13

A positive answer to Conjecture 3.6.1 can easily be shown to be a consequence of a positive answer to

3.6.2 Conjecture. The Church-Rosser theorem holds for planes; i.e. for every cyclic equivalence class $[M]_{\underline{\underline{o}}}$:

$$\forall A,B,C \in [M]_{\underline{\underline{o}}} \exists D \in [M]_{\underline{\underline{o}}} (C \twoheadleftarrow A \twoheadrightarrow B \Rightarrow C \twoheadrightarrow D \twoheadleftarrow B).$$

The motivation for considering these conjectures is that a positive answer would remove a major stumbling block in an attempt to extend the result in Bergstra and Klop (1979) to

'there exists a <u>one step</u> recursive Church-Rosser Strategy.'

Finally, for more information on reduction cycles we mention Jacopini and Venturini Zilli (1978), Böhm and Micali (1978).

REFERENCES

Barendregt, H.P. (1980). 'The Lambda Calculus, its Syntax and Semantics', North-Holland Co., Amsterdam, to appear.

Bergstra, J.A. and Klop, J.W. (1979). Church-Rosser strategies in the lambda calculus, *Theoretical Computer Science*, pp. 27-38.

Curry, H.B. and Feys, R. (1958). 'Combinatory Logic', Vol. I, North-Holland Co., Amsterdam.

Klop, J.W. (1978). Ph.D. Thesis, Math. Instituut, University of Utrecht, Budapestlaan 6, Utrecht. (To be published; first draft 1978.)

Lévy, J.-J. (1978). Réductions correctes et optimales dans le lambda-calcul, Thèse, Université Paris 7.

Stenlund, S. (1972). 'Combinators, λ-terms, and Proof Theory', D. Reidel Publ. Co., Dordrecht, Holland.

Venturini-Zilli, M. (1978). Head recurrent terms in Combinatory Logic: a generalization of the notion of head normal form. Pp. 477-493 of: 'Automata, Languages and Programming' (ed. G. Ausiello and C. Böhm). Springer Lecture Notes in Computer Science, no. 62.

Jacopini, G. and Venturini-Zilli, M. (1978). Equating for Recurrent Terms of λ-calculus and Combinatory Logic. Preprint, I.A.C., Viale del Policlinico 137, Rome.

Böhm, C. and Micali, S. (1978). Minimal forms in λ-calculus computations. Preprint, Istituto Matematico Guido Castelnuovo. Università di Roma, Citta Universitaria, Rome.

SOME UNUSUAL λ-CALCULUS NUMERAL SYSTEMS

Christopher P. Wadsworth

Department of Computer Science, University of Edinburgh, Kings Buildings, Edinburgh EH9 3JZ, Scotland

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

One of the enduring fascinations of combinatory logic is its recurring surprises. Time and again, it seems, "impossible" conjectures turn out to have positive solutions. One thinks, for instance, of the wonders resulting from the paradoxical combinator of Curry & Feys (1958); of the universal generators of Barendregt (1971); or of Plotkin's proof of ω-incompleteness (1974). Sometimes the surprises are technical eccentricities; at other times they form the very heart of the subject, and certainly some amount of λ-calculus trickery is an intrinsic part of the fascination.

One instance of such trickery will be presented here: some unexpected possibilities for numeral systems. It is well-known that the classical correspondence of Kleene (1936) between the partial recursive functions and the λ-definable functions remains valid under considerable variation in the choice of representation for the natural numbers; see, for instance, Chapter 13 of Curry, Hindley, and Seldin (1972). The "unusual" alternatives developed below will entail numerals in which the number of initial bound variables increases indefinitely with the size of the number represented. For such systems it is not immediately clear that a suitable test for zero can be defined. Technically, the difficulty will be resolved by a phenomenon we shall refer to as an η-*collapse*. We shall see, in fact, that the use of η-conversion is essential to the examples.

Section 2 begins with a brief summary of the requirements for λ-calculus numeral systems and presents the basic definitions and results in a form suitable for our development. Most of this material is covered implicitly in Curry *et al* (1972), updated here using the suggestion of Barendregt (1971) that undefined values be represented by unsolvable terms (classically any term without a normal form had been allowed). Detailed familiarity with these references, however, is not needed for the present paper. Our unusual examples are then presented in Section 3, and Section 4 concludes the paper with a few tangential remarks.

2. NUMERAL SYSTEMS AND DEFINABILITY.

First requirements for any representation of the natural numbers are the self-evident

(R1) there should be a term $\bar{0}$ to be taken as zero, and

(R2) there should be a term σ to represent the successor function, so that $\overline{n+1}$ is represented by $\sigma(\bar{n})$.

In the λ-calculus context it is reasonable to suppose also that each numeral $\bar{n}$ has a normal form, since one expects to be able to recognise that the computation of a numeric result has finished, and that different numbers are represented by distinct numerals; that is, to require

(R3) for distinct $m,n \geq 0$, the numerals $\bar{m}$ and $\bar{n}$ should have distinct (β-η-) normal forms.

(R1) - (R3) are not in themselves sufficient to ensure the definability of all recursive functions. The following definition turns out to be both adequate and convenient to our needs:

Definition. A sequence $\bar{0},\bar{1},\bar{2},\ldots,\bar{n},\ldots$ of closed terms in normal form will be called a *numeral system* just when there exist terms σ,π, and δ such that for all $n \geq 0$,

$$\sigma(\bar{n}) = \overline{n+1}\ , \quad \pi(\overline{n+1}) = \bar{n}\ ,$$

and

$$\delta(\bar{n}) = \begin{cases} T\ , & \text{if } n = 0 \\ F\ , & \text{if } n > 0 \end{cases}$$

where T and F are terms representing true and false, respectively:

$$T \equiv K \equiv \lambda x.\lambda y.x\ , \quad \text{and} \quad F \equiv KI = \lambda x.\lambda y.y$$

(Here, and in what follows, the symbol '=' between terms denotes that they are provably equal by α-β-η- conversion, while '≡' denotes syntactic identity.)

The two best known examples are those of Church (1941) and Scott (reported in Curry *et al*, 1972):

Church's system. Take $\bar{n} \equiv c_n$ generated by

$$c_0 \equiv F = \lambda x.\lambda y.\ y$$
$$\sigma_{Church} \equiv \lambda n.\lambda x.\lambda y.\ x(n\,x\,y)$$

so that

$$c_n = \lambda x.\lambda y.\ x^n y$$

Scott's system. Take $\bar{n} \equiv s_n$ generated by

$$s_0 \equiv K = \lambda x.\lambda y.\ x$$
$$\sigma_{Scott} \equiv \lambda n.\lambda x.\lambda y.\ yn$$

so that

$$s_{n+1} = \lambda x.\lambda y.\ y(s_n)\ .$$

For both these systems terms defining π and δ are easy to calculate or may be found in the references.

Before introducing λ-definability, we shall need the concept of solvability and its syntactic characterisation in terms of head normal forms. The relevant properties are summarised in the next two definitions and lemma.

Definition. A closed term M is said to be *solvable* iff there exist terms $X_1, X_2, \ldots, X_m$ $(m \geq 0)$ such that

$$M\ X_1\ X_2\ \ldots\ X_m\ =\ I\ .$$

A term M with free variables $t_1, t_2, \ldots, t_k (k > 0)$ is said to be *solvable* iff its closure $\lambda t_1 t_2 \ldots t_k . M$ is solvable. A term which preserves unsolvability will be called *strict*; that is, M is strict iff MU is unsolvable whenever U is unsolvable.

Definition. A term M is said to have a *head normal form (hnf)* iff there exist variables z and $t_1, t_2, \ldots, t_k$ $(k \geq 0)$ and terms $X_1, X_2, \ldots, X_m$ $(m \geq 0)$ such that

$$M = \lambda t_1 t_2 \ldots t_k . \; z X_1 X_2 \ldots X_m$$

The variable z in such an hnf will be called its *principal variable.*

LEMMA 1. (Barendregt, 1971; Wadsworth, 1976).

(a) If U is unsolvable, then so are $\lambda t.U$ and UX for all variables t and terms X.

(b) All terms having a normal form are solvable.

(c) In the definition of solvability any term having a normal form may be used in place of I.

(d) A term is solvable iff it has an hnf.

(e) If a term has an hnf, then it has one by β-reduction alone.

(f) For any context C[ ] and unsolvable term U, if C[U] has a normal form, then C[M] has the same normal form for all terms M.

Part(f) supports the naturalness of choosing unsolvable terms to represent undefined values (it fails for general terms without normal forms). The next lemma will be technically useful later:

LEMMA 2. In any numeral system with discriminator δ for zero,

(a) δ has an hnf with the first bound variable as principal variable; that is,

$$\delta = \lambda t_1 t_2 \ldots t_k \; . \; t_1 X_1 X_2 \ldots X_m \quad , \quad k \geq 1, m \geq 0.$$

(b) δ is strict.

Proof. For part (a) since, e.g., $\delta(\bar{0}) = T$ is solvable, δ itself must be solvable and hence have an hnf

$$\delta = \lambda t_1 t_2 \ldots t_k \; . \; z \; X_1 X_2 \ldots X_m \quad .$$

So we have to show that $k \geq 1$ and $z \equiv t_1$. If not, then either z is free in δ, or $z \equiv t_i$ with $i \neq 1$. In either case, for any term X,

(1) $\delta(X) = \lambda t_2 \ldots t_k \,.\, z\, X_1' X_2' \ldots X_m'$

where $X_j' \equiv [X/t_1]X_j$, $1 \le j \le m$.

Since δ is a discriminator for zero, in particular $\delta(\bar{0}) = T$ and $\delta(\bar{1}) = F$. But with $X \equiv \bar{0}$ the right hand side of (1) can β-η-reduce to $T = \lambda t_2 t_3 . t_2$ only if $z \equiv t_2$; whereas with $X \equiv \bar{1}$ it can β-η-reduce to $F = \lambda t_2 t_3 . t_3$ only if $z \equiv t_3$ (and only if $m = k - 3 \ge 0$ and $X_j' = t_{j+3}$ for both parts). Hence the supposition $z \not\equiv t_1$ must be false.

Part (b) now follows straightforwardly using the properties of unsolvable terms in lemma 1(a). ⊠

Definition. For each $k \ge 0$, a k-ary partial numeric function f is said to be *λ-definable* with respect to a numeral system iff there exists a term F such that, for all $n_1, n_2, \ldots, n_k \ge 0$,

$F\bar{n}_1\bar{n}_2\ldots\bar{n}_k = \bar{m}$, when $f(n_1,n_2,\ldots,n_k) = m$

$F\bar{n}_1\bar{n}_2\ldots\bar{n}_k$ is unsolvable, when $f(n_1,n_2,\ldots,n_k)$ is undefined.

The following result, due essentially to Kleene (1936), will be assumed without proof. (Modification of the proof for the use of unsolvable terms may be found in Barendregt, 1971).

THEOREM 3. For every $k \ge 0$, a k-ary partial numeric function f is partial recursive iff f is λ-definable with respect to Church numerals.

To extend the theorem to an arbitrary numeral system, notice first that the conditions in the definition are both necessary and sufficient for the existence of terms mapping back and forth between any two numeral systems:

LEMMA 4. A sequence $\bar{0},\bar{1},\bar{2},\ldots$ of terms is a numeral system iff for any other numeral system $\tilde{0},\tilde{1},\tilde{2},\ldots$ there exist strict terms H and H^{-1} such that $H(\bar{n}) = \tilde{n}$ and $H^{-1}(\tilde{n}) = \bar{n}$ for all $n \ge 0$.

Proof. Clearly it suffices to establish the result for any one particular system $\tilde{0},\tilde{1},\tilde{2},\ldots$, for then the general result follows by composing mappings via this one system. Consider

Church's numerals c_n for $\tilde{n}$ and let σ_{Church}, π_{Church}, and δ_{Church} be successor, predecessor, and test-for-zero terms for these numerals.

For the "if"-part, if H and H^{-1} exist, we need only define

$$\sigma \equiv \lambda n.\ H^{-1}\ (\sigma_{Church}(H(n)))$$
$$\pi \equiv \lambda n.\ H^{-1}\ (\pi_{Church}(H(n)))$$
$$\delta \equiv \lambda n.\ \delta_{Church}(H(n))$$

For the "only if" -part, suppose σ, π, and δ exist for $\bar{0},\bar{1},\bar{2},\ldots$ and define

$$H \equiv Y(\lambda h.\lambda n.\ \delta(n)\ (c_o)\ (\sigma_{Church}(h(\pi(n)))))$$
$$H^{-1} \equiv \lambda n.\ n(\sigma)\ (\bar{0})$$

where Y is the paradoxical combinator

$$Y \equiv \lambda f.\ (\lambda x.\ f(xx))(\lambda x.\ f(xx)).$$

The strictness of H and H^{-1} is immediate from the properties of δ and of unsolvable terms in lemmas 1 and 2. The result for H then follows by induction on n, that for H^{-1} from the iterator property

$$c_n\ x\ y\ =\ x^n\ y$$

of Church numerals. ∎

COROLLARY 5. For every $k \geq 0$, a k-ary partial numeric function f is partial recursive iff f is λ-definable with respect to *any* numeral system.

Proof. Immediate by combining the translation functions from the preceding lemma with any term F_{Church} defining f with respect to Church numerals (from theorem 3); that is,

$$F \equiv \lambda n_1 n_2 \ldots n_k\ .\ H^{-1}(F_{Church}(Hn_1)(Hn_2)\ldots(Hn_k))$$

λ-defines f with respect to the system $\bar{0},\bar{1},\bar{2},\ldots$. (The strictness of H and H^{-1} is needed for arguments at which f is undefined.)

∎

3. THE UNUSUAL EXAMPLES

Besides the two examples mentioned in Section 2, many other numeral systems are known, of varying degrees of simplicity or convenience for different purposes. One candidate with a particularly simple successor and predecessor is the sequence

$$z_n \equiv K^n I \quad , \quad n \geq 0 ,$$

for which one can take

$$\sigma \equiv K$$
$$\pi \equiv \lambda n.\ nA \quad , \text{ where } A \text{ may be arbitrary.}$$

Unfortunately, this sequence fails to be a numeral system because it lacks a definable test for zero:

LEMMA 6. There is no term δ such that $\delta(z_o) = T$ and $\delta(z_{n+1}) = F$ for all $n \geq 0$.

Proof. By lemma 2(a), for any δ we have

$$\delta = \lambda t_1 t_2 \ldots t_k.\ t_1 X_1 X_2 \ldots X_m \quad , \quad k \geq 1,\ m \geq 0.$$

Then for any $n \geq m$,

$$\begin{aligned}
\delta(z_n) &\equiv \delta(K^n I) = \delta(K^m(K^{n-m}I)) \\
&= \lambda t_2 \ldots t_k.\ K^m(K^{n-m}I)\ X_1{}''\, X_2{}'' \ldots X_m{}'' \quad (X_i'' = [K^n I/t]X_i) \\
&= \lambda t_2 \ldots t_k.\ K^{n-m}I \\
&\equiv \lambda t_2 \ldots t_k\ t_{k+1} \ldots t_{k+n-m+1}\ .\ t_{k+n-m+1} \\
&\neq \lambda t_2 t_3 . t_3 \equiv F \text{ , for all } n \text{ such that } k+n-m+1 \neq 3.
\end{aligned}$$

Thus, δ cannot satisfy $\delta(z_{n+1}) = F$ for sufficiently large n, contradicting the existence of δ. ☒

An alternative, though longer proof of lemma 6 was also found by Barendregt (private communication).

The sequence z_n would be unusual in providing numerals whose number of initial bound variables increases indefinitely as the number represented increases: writing the terms z_n out explicitly, one would have

$$z_n \equiv \lambda x_1 x_2 \ldots x_{n+1}\ .\ x_{n+1} \quad .$$

This contrasts with previously known systems in which the number of initial bound variables in the numerals remains bounded. (Of course, we disallow trivial variations on known systems obtained by applying unbounded numbers of η-expansions to the numerals.)

Can there exist systems in which the numerals have the form

(2) $\qquad \bar{n} = \lambda x_1 x_2 \ldots x_{k_n} \,.\, E_n$

with k_n increasing indefinitely with n (and the E_n having normal forms)? As with the proposed sequence z_n, other sequences having a successor and predecessor are easy to find; so the question reduces to the discovery of a sequence with a test for zero. The difficulty, as indicated by the proof of lemma 6, is that there is apparently no way of absorbing the unbounded number of initial bound variables by applying numerals to any fixed finite number of arguments.

Indeed, as was observed by Hindley and by Plotkin (private communications), it follows by much the same kind of argument as that in lemma 6 that there can be no solution to this difficulty using only β-conversion:

LEMMA 7. Suppose (2) is a sequence in which k_n increases indefinitely with n. Then there is no term δ such that $\delta(\bar{0})$ β-reduces to T while $\delta(\overline{n+1})$ β-reduces to F for all $n \geq 0$.

Proof. If δ exists, then by lemmas 1(e) and 2(a)

$$\delta =_\beta \lambda t_1 t_2 \ldots t_k \,.\, t_1 X_1 X_2 \ldots X_m \;, \quad k \geq 1,\ m \geq 0.$$

Then for all n such that $k_n \geq m$,

$$\begin{aligned}\delta(\bar{n}) &=_\beta \lambda t_2 \ldots t_k \,.\, (\lambda x_1 x_2 \ldots x_{k_n} \,.\, E_n) X_1 X_2 \ldots X_m \\ &=_\beta \lambda t_2 \ldots t_k \,.\, \lambda x_{m+1} \ldots x_{k_n} \,.\, E_n{}' \\ &\qquad \text{where } E_n{}' \equiv [X_1, X_2, \ldots, X_m / x_1, x_2, \ldots, x_m]\, E_n \,.\end{aligned}$$

Now for large enough k_n the latter can never reduce to T or F by β-conversion alone, since any β-conversion must be internal to $E_n{}'$ and leave the initial bound variables unchanged. ⊠

In the rest of this section, we shall develop two examples, in which in fact k_n increases linearly with n, obtained by making use of η-conversion. For this purpose it is convenient first to change bound variables in (2) and write the numerals we seek in the form

$$(3) \quad \bar{n} \equiv \lambda x x_{k_n} x_{k_n - 1} \dots x_1 \,.\, E_n \quad , \qquad n \geq 0,$$

where we have distinguished the first bound variable as x and re-indexed the others to count down from k_n rather than up from 1. Consider now E_n of the form

$$E_n \equiv x\, A_n\, x_{k_n} x_{k_n - 1} \dots x_1$$

and notice first that if none of $x_{k_n}, \dots, x_1$ occur in A_n, then k_n η-contractions can be applied to $\bar{n}$ to give $\bar{n} = \lambda x.x\, A_n$. (Since we are seeking examples in which k_n increases with n, we shall call such an unbounded number of η-contractions an η-*collapse*.) Of course the numerals would not then have an unbounded number of initial bound variables, so we suppose instead that some or all of $x_{k_n}, \dots, x_1$ occur free in A_n. Then η-reduction of $\bar{n}$ is not possible immediately, but we can arrange that $\bar{n}$ is applied to suitable arguments to cancel out occurrences of $x_{k_n}, \dots, x_1$ in A_n, thus enabling an η-collapse of the resulting term.

We consider here a simple case with $k_n = n$ and all the variables $x_n, x_{n-1}, \dots, x_1$ occurring in A_n, namely

$$A_0 \equiv A$$
$$A_{n+1} \equiv x_{n+1} A_n \equiv x_{n+1}(x_n(\dots(x_1 A)\dots)) \; , \; n \geq 0,$$

where A is some term to be determined later (so that $\bar{0}$ and $\overline{n+1}$ will be discriminated). Now if we look at what happens when the numerals are applied to two arguments B,C, also to be determined later, then for $\overline{n+1}$ with $n \geq 0$ we find

$$\overline{n+1}\; B\, C \equiv (\lambda x x_{n+1} x_n \dots x_1 .\; x(x_{n+1} A_n) x_{n+1} x_n \dots x_1) B\, C$$
$$= \lambda x_n \dots x_1 .\;\; B(C\, A_n)\, C x_n \dots x_1$$

$$= \lambda x_n \ldots x_1 .\ \ B\,C'(KC')x_n \ldots x_1 \quad \text{, choosing } C \equiv KC'$$

(4.1) $= B\,C'(KC')$, by an η-collapse,

assuming, say, B and C are closed terms; for $\bar{0}$ we find

$$\bar{0}\ B\ C \equiv (\lambda x.xA_o)B\,C \quad = \quad B\,A_o\,C$$

(4.2) $= B\,A\,(KC')$, for $C \equiv KC'$ and $A_o \equiv A$.

It is now easy to choose A,B, and C' so that (4.1) and (4.2) reduce to F and T, respectively; e.g. take

$$A \equiv KT \quad , \quad B \equiv I \quad , \quad C' \equiv KF$$

Collecting the various parts together, we have

LEMMA 8. For the sequence with

(5) $\bar{n} \equiv w_n \equiv \lambda x x_n x_{n-1} \ldots x_1 .\ x(x_n(x_{n-1}(\ldots(x_1(KT))\ldots)))x_n x_{n-1} \ldots x_1$

the term

(5.1) $\delta_w \cdot = \lambda n.\ \ n\,I\,(K^2F)$

has the properties required of a test for zero. ⊠

Formulae for σ_w and π_w for the sequence (5) are now easily found. For w_{n+1} , we calculate

$$w_{n+1} \equiv \lambda x x_{n+1} x_n \ldots x_1 \ .\ x(x_{n+1} A_n)x_{n+1} x_n \ldots x_1$$

$$= \lambda x.\lambda y.\lambda x_n \ldots x_1 .\ x(y\,A_n\,)y\ x_n \ldots x_1 \ ,$$

by α-conversion and convention for the abbreviated multiple λ-notation

$$= \lambda x.\lambda y.\lambda x_n \ldots x_1 .\ (\lambda a.x(ya)y)A_n x_n \ldots x_1$$

$$= \lambda x.\lambda y.\ w_n\ (\lambda a.\ x(ya)y)$$

so we can take

(5.2) $\sigma_w = \lambda n.\lambda x.\lambda y.\ n(\lambda a.x(ya)y)$.

Similarly, one finds

(5.3) $\pi_w = \lambda n.\ \lambda x.\ n(\lambda a.\ K(xa))I.$

COROLLARY 9. The sequence (5) gives a numeral system in which the number of initial bound variables in $\bar{n} \equiv w_n$ increases linearly with n. ⊠

Our second example is a system in which every numeral is a permutator; that is, $\bar{n}$ is of the form

$$\bar{n} = \lambda x\, x_{k_n}\; x_{k_n-1} \ldots x_1 \,.\quad E_n$$

with E_n being a permutation of the variables $x, x_{k_n}, x_{k_n-1}, \ldots, x_1$. We shall consider a case with $k_n = n+1$, so we seek E_n as a permutation of $x, x_{n+1}, x_n, \ldots, x_1$. The basic idea remains the same: to absorb the indefinite number of initial bound variables we arrange that an η-collapse is not possible for $\bar{n}$ itself, but becomes possible after application to suitably chosen arguments.

This time, however, we cannot try E_n of the form

$$E_n \quad = \quad F_n\, x_{n+1}\, x_n \ldots x_1$$

for the desire that E_n be a permutation would then require that none of $x_{n+1}, x_n, \ldots, x_1$ occur in F_n, leading to an η-collapse of $\bar{n}$ immediately. Instead, we shall seek E_n as a combination $A_n\, B_n$ whose operand B_n is not a single variable (in particular, not x_1). The simplest such E_n results from taking B_n as a combination of two of the variables, e.g. the first and the last, and taking A_n as the remaining variables (if any) in order, as follows:

(6) $\bar{n} \equiv p_n \equiv \lambda x x_{n+1} x_n \ldots x_1 \,.\; x_{n+1} x_n \ldots x_2 (x\, x_1)$.

Now consider the application of these terms to three arguments A,B, and C. For $\bar{0}$ we find

(7.1) $\bar{0}\; A\, B\, C \equiv (\lambda x x_1 . x x_1) A\, B\, C = A\, B\, C$.

For $\overline{n+1}$ we calculate[1]

$$\begin{aligned}\overline{n+1}\; A\, B\, C &\equiv (\lambda x x_{n+2} x_{n+1} \ldots x_1 \,.\; x_{n+2} x_{n+1} \ldots x_2 (x x_1)) A\, B\, C \\ &= \lambda x_n \ldots x_1 .\;\; B\, C\, x_n \ldots x_2 (A x_1)\end{aligned} \tag{7.2}$$

An η-collapse would now be applicable to (7.2) if we chose $A \equiv I$; but then (7.1) and (7.2) *both* reduce to just $B\,C$, so there can be no test for zero with this choice.

Fortunately there is another approach. Comparing (7.1) and

(7.2) we observe that they have different (undetermined) terms A and B, respectively, essentially at the head of the expressions, so we should be able to separate them. But we must ensure that all non-zero numerals are treated uniformly, which requires us to cope with the problem that, in (7.2), B is followed by an indefinite number of arguments increasing with n. To absorb these arguments we can make use here of the numerals themselves! That is, consider taking B as the numeral one started with applied to two arguments D and E. Then we have

$$\begin{aligned}\bar{0}\ A(\bar{0}\,D\,E)C &= A(D\,E)C\\ \overline{n+1}\ A(\overline{n+1}\ D\,E)C &= \lambda x_n\ldots x_1.\ \overline{n+1}\,D\,E\,C\ x_n\ldots x_2(Ax_1)\\ &= \lambda x_n\ldots x_1.\ E\,C\,x_n\ldots x_2(D(Ax_1))\end{aligned}$$

To arrange on η-collapse in the latter, we need to choose A and D such that $D(Ax_1) = x_1$. The choice $A \equiv D \equiv I$ would achieve this, but we saw earlier that taking $A \equiv I$ will cause the argument to break down. However, we still have $D(Ax_1) = x_1$ if we choose, e.g., $A \equiv K$ and $D \equiv \lambda a.a\,G$, where G may be any term. Then

$$\begin{aligned}\bar{0}\,K(\bar{0}(\lambda a.a\ G)E)C &= K((\lambda a.aG)E)C = K(EG)C\\ &= E\,G\\ \overline{n+1}\ K(\overline{n+1}(\lambda a.aG)E)C &= \lambda x_n\ldots x_1.\ EC\ x_n x_{n-1}\ldots x_2 x_1\\ &= E\,C \quad ,\quad \text{by an } \eta\text{-collapse.}\end{aligned}$$

Again it is now easy to choose C,E, and G such that $E\,G = T$ and $E\,C = F$, e.g. take $C \equiv F$, $E \equiv I$, $G \equiv T$. Collecting the various choices together, we thus have

LEMMA 10. For the sequence (6), the term

$$\delta_p = \lambda n.\ n\,K\,(n(\lambda a.a\,T)\ I)F$$

has the properties required of a test for zero. ⊠

Again formulae for σ_p and π_p for the sequence (6) are easily found. For p_{n+1} , we calculate

$$\begin{aligned}p_{n+1} &\equiv \lambda x x_{n+2}x_{n+1}x_n\ldots x_1.\ x_{n+2}x_{n+1}x_n\ldots x_2(x\,x_1)\\ &= \lambda xyzx_n\ldots x_1.\ yzx_n\ldots x_2(x\,x_1) \quad ,\quad \text{by } \alpha\text{-conversion}\end{aligned}$$

$$= \lambda xyz.\ ((\lambda x_{n+1}.\lambda x_n \ldots x_1.x_{n+1}x_n \ldots x_2(xx_1))(yz))$$
$$= \lambda xyz.\ p_n\ x(yz)$$

so we can take[1]

$$\sigma_p = \lambda n.\lambda x.\lambda y.\lambda z.\ n\,x\,(yz)\ .$$

For the predecessor, one may easily verify that $\pi_p(p_{n+1}) = p_n$ holds for

$$\pi_p = \lambda n.\lambda x.\ n\,x\,I\ .$$

COROLLARY 11. The sequence (6), in which the body of each numeral p_n is just a permutation of its $n+2$ bound variables, is a numeral system.

Barendregt has brought to our attention that Böhm also had found a system in which each numeral $\bar{n}$ is a permutation of $n+2$ bound variables, obtained by reversing the last two; that is, the sequence

(8) $\bar{n} \equiv b_n \equiv \lambda x_1x_2\ldots x_{n+2}.\ x_1x_2\ldots x_nx_{n+2}x_{n+1}$.

The reader may easily verify that the following terms determine the sequence (8) as a numeral system:

$$\sigma_b = \lambda n.\lambda x.\lambda y.\ n(xy)$$
$$\pi_b = \lambda n.\ nI$$
$$\delta_b = \lambda n.\ n(n(KF))(KT)\ .$$

Again notice the double application of n in the term for δ_b, this time to repeat the reversal of the last two variables in the body of the numerals.

4. ADDITIONAL REMARKS

(a) The requirements (R3) that numerals have distinct β-η-normal forms is slightly stronger than is necessary to assume. It suffices in fact to assume distinct β-normal forms, for the definability of all recursive functions then implies they are β-η-distinct:

LEMMA 12. There is no numeral sequence in which for distinct

$m,n \geq 0$, $\bar{m}$ and $\bar{n}$ have distinct β-normal forms but the same β-η-normal form.

Proof. Consider the numeric function

$$f(x) = \begin{cases} 0 & , \text{if } x = m \\ \text{undefined} & , \text{otherwise} \end{cases}$$

Clearly f is partial recursive and λ-defined by some term $\bar{f}$ so that, in particular,

$$\bar{f}(\bar{m}) = \bar{0} \quad , \quad \bar{f}(\bar{n}) \text{ is unsolvable.}$$

If $\bar{m}$ and $\bar{n}$ had the same β-η-normal form, then $\bar{f}(\bar{m})$ and $\bar{f}(\bar{n})$ would be β-η-convertible, which then implies $\bar{0}$ is unsolvable, contradicting our requirement that numerals have normal forms.

(b) On the other hand, although for computational purposes one would naturally choose numerals with normal forms, and we have preferred to include this in the requirement (R3), this is not strictly necessary for the definability of all partial recursive functions. One may relax (R3) to require only that numerals are β-(η-)distinct, whether they have normal forms or not. An example of an adequate numeral system consisting of terms not having a normal form is the following (due to Böhm and Dezani):

$$\begin{aligned} \bar{0} &\equiv Y \\ \sigma &\equiv \lambda nz.znY \\ \pi &\equiv \lambda n.nK \\ \delta &\equiv \lambda n.\ n(K^2T)I \end{aligned}$$

(c) In recursive function theory it is conventional to make the predecessor function total on the natural numbers by declaring it as mapping 0 to 0. One may notice that the results above hold independently of any choice of value for $\pi(\bar{0})$ that might have been made in the definition of numeral systems in Section 2.

(d) Taking the σ-π-δ-criteria as a "standard" definition for an adequate numeral system seems particularly natural for many applications besides those above. Of course, corollary 5 essentially establishes the definition as an "absolute" concept, for any sequence $\bar{0},\bar{1},\bar{2},\ldots$ on which all recursive functions can be defined must clearly have σ,π, and δ definable. This encourages further the abstract approach to combinatory arithmetic begun in Curry *et al* (1972).

ACKNOWLEDGEMENTS

Professor Curry and Drs. Henk Barendregt, Roger Hindley, and Gordon Plotkin all made helpful suggestions on an early draft of this material. I am grateful to them all for their comments, and to Barendregt also for bringing the system of Böhm to my attention. Hindley and Curry also observed a mistake in the original term given for the predecessor function for the permutator system (6).

REFERENCES

Barendregt, H.P. (1971). Some extensional term models of combinatory logics and λ-calculi, Ph.D. thesis, Utrecht, The Netherlands.

Church, A. (1941). The calculi of λ-conversion, *Annals of Mathematics* 6, Princeton, New Jersey.

Curry, H.B. and Feys, R. (1958). "Combinatory Logic : Volume I", North Holland, Amsterdam.

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

Kleene, S.C. (1936). λ-definability and recursiveness, *Duke Math.J.* 2, 340-353.

Plotkin, G.D. (1974). The λ-calculus is ω-incomplete, *J.Symbolic Logic* 39, 313-317.

Wadsworth, C.P. (1976). The relation between computational and denotational properties for Scott's D_∞-models of the λ-calculus, *SIAM J. on Computing* 5, 488-521.

Note 1, added in proof (February 1980).

Equation (7.2) is incorrect for $\bar{1} \equiv p_1$, that is when $n = 0$, since then x_{n+1} is the variable x_1. The discussion given therefore shows that δ_p in lemma 10 distinguishes p_o from p_{n+1} only for $n > 0$. Somewhat fortuitously, however, it may be verified that $\delta_p(p_1)$ does in fact reduce to F as required.

A similar mistake is present in the derivation of a successor term σ_p following lemma 10, with the effect that the terms given both for σ_p and, following equation (8), for σ_b are correct only for positive numerals. But for any sequence $\bar{0}, \bar{1}, \bar{2}, \ldots$ with a test δ for zero, a successor term σ^+ for positive numerals may be extended to all numerals by defining

$$\sigma \equiv \lambda n.\delta(n)(\bar{1})(\sigma^+(n))$$

AN ABSTRACT APPROACH TO (HEREDITARY) FINITE SEQUENCES OF COMBINATORS

Corrado Böhm

Istituto Matematico "G. Castelnuovo", Università di Roma, Italy.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

The aim of this paper is to fill a gap between some choices done by Church (1941), McCarthy (1962), Böhm and Gross (1966), Venturini-Zilli (1965), Kearns (1969) and especially by Curry and al. (1972) and Curry (1975) in order to introduce more or less abstractly the notion of linear and multilevel (or hereditary as in the title) finite sequence of atoms or of combinatory terms.

The essential improvement over the preceding approaches is the release of the requirement on the existence of two objects as in Curry (1972, 1975):

(i) a special atom different from any other of a given class like ⟦end⟧ or #

(ii) a discriminator, like ⟦ltr⟧, between the special atom and the other ones.

As a consequence the class of atoms can be extended to the whole class Λ of combinatory terms.

The class of linear finite sequences, although semi-decidable, will be recursively defined so that each of its members will satisfy a fixed point equation, together with a finiteness condition.

Some additional technical improvements are:

(iii) The mapping *lgh* transforming each linear sequence into the number (numeral) of its components is expressible by means of the basic combinators.

(iv) On Mc Carthy (1962). The associative operator $[\![\mathrm{cat}]\!]$, as defined in Curry (1975) and corresponding to *append* in LISP, is a basic one. Similarly, no recursion is needed to "rotate to the right" a linear sequence.

Our abstract approach is justified by the existence of an infinity of realizations for the basic combinators used for constructing sequences.

2. FINITE LINEAR SEQUENCES

Given arbitrary terms represented by variables x_1, x_2,...etc. we look for a combinatory term representing the sequence

$$X \equiv \langle x_1, x_2, \ldots, x_n\rangle \ .$$

By definition n is here the length of X and we will write

$$\mathrm{lgh}\ X = n$$

There is just one sequence whose lgh is zero and we will denote it by $\langle\rangle$, the empty sequence. Following McCarthy (1962) and Curry and al. (1972) we require the existence of a certain set of operators to make the construction and the manipulation of sequences possible.

In order to construct sequences it is sufficient: to build from an atom a sequence of lgh one by means of an operator ⟦unit⟧, to concatenate two sequences by the operator ⟦cat⟧, which will be associative (its infixed version will be denoted by $\frown$).

In order to manipulate a sequence the extraction of the first component, by the operator ⟦head⟧, is needed, together with the possibility of erasing it from the sequence, by ⟦tail⟧, and of discriminating the case <> or ⟦nil⟧, where there is no component, by ⟦null⟧, from the other ones. Summing up we require:

0) $[\![\mathrm{unit}]\!]\, x = \langle x \rangle$

1) $X \frown (Y \frown Z) = (X \frown Y) \frown Z$

2) $X \frown \langle\rangle = \langle\rangle \frown X = X$

$$\left.\begin{array}{ll} 3) & [\![\mathrm{head}]\!]\, \langle x_1,\dots,x_n\rangle = x_1 \\ 4) & [\![\mathrm{tail}]\!]\, \langle x_1,x_2,\dots,x_n\rangle = \langle x_2,\dots,x_n\rangle \end{array}\right\} \; n > 0$$

5) $[\![\mathrm{null}]\!]\, \langle\ \rangle = 1$

6) $[\![\mathrm{null}]\!]\, \langle x_1,\dots,x_n\rangle = 0 \qquad n > 0$

where

$$X \equiv \langle x_1,\dots,x_n\rangle \qquad Y \equiv \langle y_1,\dots,y_m\rangle \qquad m,n \geqslant 0$$

and

$$X \frown Y = \langle x_1,\dots,x_n,y_1,\dots,y_m\rangle \quad \text{and similarly for } Z\,.$$

Moreover 0 and 1 are the Church numerals for zero and one.

In contrast with Curry (1972, 1975) we are interested in basic operators independent from the length of sequences they are acting on.

We will then first assume the existence of a six-tuple of combinatory terms (⟦cat⟧ , ⟦nil⟧, ⟦unit⟧, ⟦head⟧, ⟦tail⟧, ⟦null⟧)

satisfying a set of axiom schemes with somewhat stronger requirements than the preceding ones:

(1') $[\![\mathrm{cat}]\!]\, x\, ([\![\mathrm{cat}]\!]\, yz) = [\![\mathrm{cat}]\!]\, ([\![\mathrm{cat}]\!]\, xy) z$

(2') $[\![\mathrm{cat}]\!]\, x\, [\![\mathrm{nil}]\!] = [\![\mathrm{cat}]\!]\, [\![\mathrm{nil}]\!]\, x = x$

(3') $[\![\mathrm{head}]\!]\, ([\![\mathrm{cat}]\!]\, ([\![\mathrm{unit}]\!]\, x) y) = x$

(4') $[\![\mathrm{tail}]\!]\, ([\![\mathrm{cat}]\!]\, ([\![\mathrm{unit}]\!] x) y) = y$

(5') $[\![\mathrm{null}]\!]\, [\![\mathrm{nil}]\!] = 1$

(6') $[\![\mathrm{null}]\!]\, ([\![\mathrm{cat}]\!]\, ([\![\mathrm{unit}]\!] x) y) = 0$

where x,y,z are arbitrary variables (or terms). Secondly the set $\mathcal{L}$ of finite linear sequences is the smallest set of terms satisfying the following inductive structural definition:

$$[\![\mathrm{nil}]\!] \in \mathcal{L} \qquad \text{(i)}$$

$$t \in \Lambda \Rightarrow [\![\mathrm{unit}]\!]\, t \in \mathcal{L} \qquad \text{(ii)}$$

$$t \in \Lambda \wedge Y \in \mathcal{L} \Rightarrow [\![\mathrm{cat}]\!]\, ([\![\mathrm{unit}]\!]\, t)\, Y \in \mathcal{L} \qquad \text{(iii)}$$

Then we have the following result:

LEMMA 1

a) $X \in \mathcal{L} \Rightarrow (\exists\, n \geq 0)([\![\mathrm{tail}]\!]^n X = [\![\mathrm{nil}]\!]^1)$

(Finiteness)

b) $X \in \mathcal{L} \Rightarrow X = [\![\mathrm{null}]\!]\, X (K[\![\mathrm{nil}]\!])$

$([\![\mathrm{cat}]\!]\, ([\![\mathrm{unit}]\!]([\![\mathrm{head}]\!]\, X))([\![\mathrm{tail}]\!]\, X))$

(Fixed point equation).

Proof

a) $[\![\mathrm{tail}]\!]^0 [\![\mathrm{nil}]\!] = [\![\mathrm{nil}]\!]$ by definition. $[\![\mathrm{tail}]\!]\, ([\![\mathrm{unit}]\!]\, t) = [\![\mathrm{nil}]\!]$ by 2' and 4'. Induction on the number of applications of iii as a rule, together with 4'.

b) By case analysis: using 2' to 6' in the cases i, ii, etc. the fixed point equation simplifies into $X = [\![\mathrm{nil}]\!]$, $X = [\![\mathrm{unit}]\!]\, t$, etc.

The next definition, suggested by part (a) of the preceding proof:

DEFINITION 2. $X \in \mathcal{L} \Rightarrow \mathrm{lgh}\ X \equiv \mu i\ ([\![\mathrm{tail}]\!]^i X = [\![\mathrm{nil}]\!])$

enables us to state the main theorem for finite sequences:

THEOREM 1

$$X,Y \in \mathcal{L} \Rightarrow X = Y \Leftrightarrow \exists n\, [\mathrm{lgh}\ X = \mathrm{lgh}\ Y = n \wedge (\forall j\ o \leq j < n)([\![\mathrm{head}]\!]([\![\mathrm{tail}]\!]^j X) = [\![\mathrm{head}]\!]([\![\mathrm{tail}]\!]^j Y))] .$$

<u>Proof of</u> $\Rightarrow$: By the (μ) rule (Curry and al. 1972 p. 23) applied to combinators $[\![\mathrm{head}]\!]$ and $[\![\mathrm{tail}]\!]$ and from the fact that lgh is a function.

<u>Proof of</u> $\Leftarrow$: By induction on the value of $n = \mathrm{lgh}\ X = \mathrm{lgh}\ Y$ the case $n = 0$ being satisfied by $X = Y = [\![\mathrm{nil}]\!]$.

With the notation $x_{j+1} \equiv [\![\mathrm{head}]\!]([\![\mathrm{tail}]\!]^j X)$, for $0 \leq j < \mathrm{lgh}\ X$ and $X \in \mathcal{L}$, theorem 1 and the iterated use of definitions i, ii and iii give the following result

$$(7) \qquad X \in \mathcal{L} \Rightarrow \begin{cases} X = [\![\mathrm{unit}]\!]\, x_1 \frown \ldots \frown [\![\mathrm{unit}]\!]\, x_m & \text{if lgh } X > 0 \\ X = [\![\mathrm{nil}]\!] & \text{otherwise} \end{cases}$$

which justifies the notation

$$X = \langle x_1, \ldots, x_n \rangle \quad \text{or} \quad X = \langle\,\rangle$$

given at beginning of this section.

We may exhibit also a λ-term for lgh if we express the integer n by $\hat{n} \equiv \lambda xy.\underbrace{x(x..(xy)..)}_{n}$, following Church, and the successor function by

$$\hat{\sigma} \equiv \lambda xyz.xy(yz):$$

$$\hat{\mathrm{lgh}} \equiv \lambda xy.\ [\![\mathrm{null}]\!]\, x\ \hat{0}(\hat{\sigma}(\hat{\mathrm{lgh}}([\![\mathrm{tail}]\!]\, x))\ y)$$

Finally we may λ-define a combinator <u>rrot</u> acting as follows:

$$\mathrm{rrot}\ \langle x_1, \ldots, x_n \rangle = \langle x_2, \ldots, x_n, x_1 \rangle \quad \text{for } n > 0$$

by the non recursive definition

$$\mathrm{rrot} \equiv \lambda x.\, [\![\mathrm{cat}]\!]\, ([\![\mathrm{tail}]\!]\, x)(\, [\![\mathrm{unit}]\!]\, (\, [\![\mathrm{head}]\!]\, x)).$$

3. REALIZATIONS

Let D be the Church abstract pair

$$D \equiv \lambda xyz.zxy \equiv \lambda xy.[x,y]$$

and o the infixed version of the combinator

$B \equiv \lambda xyz.x(yz)$. Then the following six-tuple

$$([\![\mathrm{cat}]\!] \equiv B, [\![\mathrm{nil}]\!] \equiv I, [\![\mathrm{unit}]\!] \equiv D, [\![\mathrm{head}]\!] \equiv D\alpha K,$$
$$[\![\mathrm{tail}]\!] \equiv \lambda xy.xy\hat{0}, [\![\mathrm{null}]\!] \equiv D\hat{0}\ (K^2\hat{0}))$$

where $I \equiv \lambda x.x$ and α is arbitrary, satisfies the axiom schemes 1'-6'. By this realization (7) becomes

$$X \in \mathcal{L} \Rightarrow \begin{cases} X \equiv \langle x_1,\ldots,x_n\rangle = Dx_1 o \ldots o Dx_n = \lambda t.[x_1,[\ldots[x_n,t]\ldots]] \\ \qquad \text{if } \mathrm{lgh}\ X > 0 \\ X \equiv I \quad \text{if} \quad \mathrm{lgh}\ X = 0 \end{cases}$$

which is similar, but not identical, to case b) 13C2 in Curry and al. (1972) p. 263[2].

Our abstract axiom schemes are justified by the

REMARK 1. There exists the following infinity of realizations or six-tuples:

$$(\phi^n B,\ K^n I,\ C_{[n+2]}I,\ C_{[n+2]}I\alpha_1\ldots\alpha_{n+1}K^{n+1}, \lambda x_1\ldots x_{n+2}.x_1\ldots x_{n+2}(K^n\hat{0}),$$
$$,\ C_{[n+2]}I\alpha_1\ldots\alpha_n\hat{0}(K^{n+2}\hat{0})).$$

The combinators ϕ and $C_{[n]}$ are defined in Curry and al. (1958) 5A2 and 5E2 and $\alpha_1\ldots\alpha_{n+1}$ are arbitrary terms. In order to render this text self-contained some λ-definitions follow:

$\phi \equiv \lambda fabx\,.f(ax)(bx)$

$\phi^n \equiv \lambda fabx_1 \dots x_n . f(a\, x_1 \dots x_n)(b\, x_1 \dots x_n)$

$C_{[n]} \equiv \lambda f x_1 \dots x_{n+1} . f x_{n+1} x_1 \dots x_n$

$C_{[n+2]} I \equiv \lambda x_1 \dots x_{n+3} . x_{n+3} . x_1 \dots x_{n+2}$

The case n=0 is the realization from the start of this section; it is easy to check the result by induction on n.

4. HEREDITARY FINITE SEQUENCES

When inside a linear finite sequence (lfs) each component is again a lfs and this recurs a finite number of times we are faced with an *hereditary finite sequence* (hfs).

Maintaining the requirement of the absolute arbitrariness for the atoms building a hfs, some difficulties arise when we wish to recover constructively the single atoms from a hfs, since an atom can be any term whatsoever, even a hfs...

There are still other ways of considering the hfs's, illustrated by their visualization through ordered rooted tree structures (Fig. 1) or by ordered rooted trees labelled at each node (Fig. 2) or by ordered rooted trees labelled only at their end nodes (leaves) (Fig. 3). We will call $\mathfrak{T}_1$, $\mathfrak{T}_2$ and $\mathfrak{T}_3$ the corresponding families of hfs.

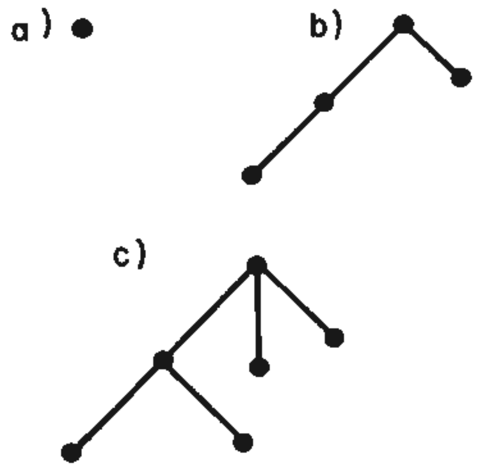


Fig. 1. Examples of ordered rooted tree structures.

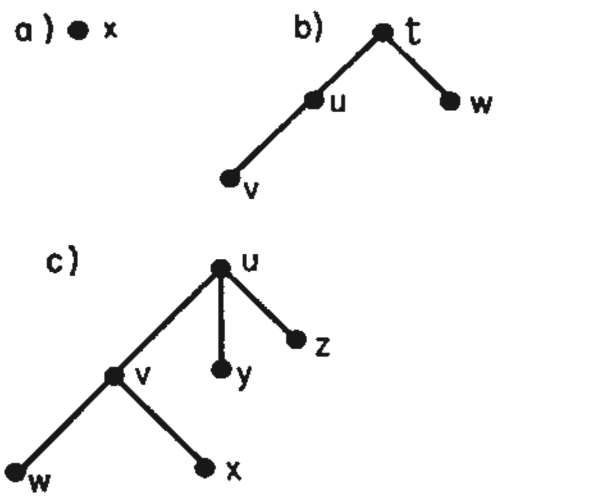


Fig. 2. Examples of labelled ordered rooted trees.

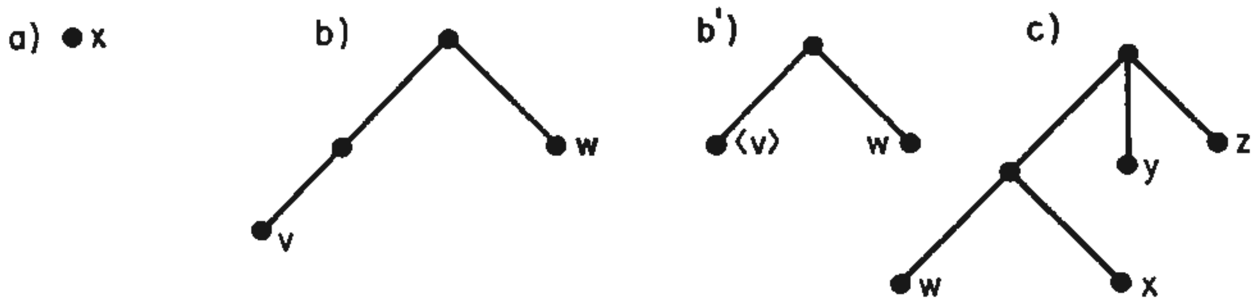


Fig. 3. Examples of ordered rooted trees with labelled leaves.

DEFINITION OF $\mathcal{T}_1$

In this case the only atom admitted is $\llbracket \mathrm{nil} \rrbracket$. The finiteness condition is that the function $\#_1$ counting the number of atoms be always defined. The following is a recursive domain description:

$$T \varepsilon \mathcal{L} \Rightarrow \{T \varepsilon \mathcal{T}_1 \Leftrightarrow (\llbracket \mathrm{null} \rrbracket T \vee (\llbracket \mathrm{head} \rrbracket T \varepsilon \mathcal{T}_1 \wedge \llbracket \mathrm{tail} \rrbracket T \varepsilon \mathcal{T}_1)) \wedge (\exists\, n)\ (\#_1 T = \hat{n})\}$$

where

$$\#_1 \equiv \lambda x. \llbracket \mathrm{null} \rrbracket\, x \hat{0} (\llbracket \mathrm{null} \rrbracket\, (\llbracket \mathrm{tail} \rrbracket\, x) (K(\#_1 (\llbracket \mathrm{head} \rrbracket\, x))) (\hat{+}\ (\#_1 (\llbracket \mathrm{head} \rrbracket\, x))\ (\#_1 (\llbracket \mathrm{tail} \rrbracket\, x))))$$

and

$\hat{+} \equiv \lambda vxyz.vy(xyz)$ (addition for Church numerals).

Following this definition the examples of Fig. 1 are written as strings of properly paired parentheses:

a): <> b): <<<>><>> c): <<<><>><><>> .

REMARK 2. An example of an "infinite" object belonging to $\mathcal{L}$ but not to $\mathcal{J}_1$ is any solution U of the fixed-point equation $U = [\![unit]\!]\,U$. In fact it follows that

$$[\![head]\!]\,U = [\![head]\!]\,([\![unit]\!]\,U) = U$$

and

$$\#_1([\![head]\!]\,U) = \#_1\,U$$

leaving the value of $\#_1$ undefined.

DEFINITION OF $\mathcal{J}_2$

To each lfs occurring inside a hfs an arbitrary combinatory term is assigned as label: the agreement is to postfix the label as last component of every lfs forming the hfs. This method is a slight simplification of that one described by Coppo and Dezani-Ciancaglini (1977).

The recursive definition of the domain $\mathcal{J}_2$ is

$$X \in \mathcal{L}-\{[\![nil]\!]\} \Rightarrow \{X \in \mathcal{J}_2 \Leftrightarrow ([\![null]\!]\,([\![tail]\!]\,X) \vee (\forall j: 0 \leq j < lgh\,X-1)\ ([\![head]\!]\,([\![tail]\!]^j\,X)) \in \mathcal{J}_2) \wedge (\exists\, n)(\#_2\,X = \hat{n})\}$$

where

$$\#_2 \equiv \lambda x.\,[\![null]\!]\,([\![tail]\!]\,x)\hat{0}(\hat{+}(\#_2([\![head]\!]\,x))(\#_2([\![tail]\!]\,x))).$$

A simple class of "infinite" objects belonging to $\mathcal{L}-\{[\![nil]\!]\}$ but not to $\mathcal{J}_2$ is any solution V of the fixed-point equation

$V = [\![cat]\!]\,([\![unit]\!]\,V)\ ([\![unit]\!]\,\alpha)$

where α is an arbitrary combinatory term. The proof runs essentially as in Remark 2.

The examples of Fig. 2 are written as

a): $\langle x\rangle$ b): $\langle\langle\langle v\rangle u\rangle\langle w\rangle t\rangle$ c): $\langle\langle\langle w\rangle\langle x\rangle v\rangle\langle y\rangle\langle z\rangle u\rangle$.

DEFINITION OF $\mathcal{J}_3$.

To label only the leaves of an ordered rooted tree amounts to considering a hfs where atoms may be located anywhere inside a lfs component and not only in the last position as was the case in $\mathcal{J}_2$. The problem here is how to recognize atoms inside a hfs, since no predicate-like combinator ATOM may exist under the hypothesis of arbitrariness of atoms.

Our proposal is to identify, so to say, atoms with unit sequences, forbidding hence the formation in $\mathcal{J}_3$ of unit sequences of non atomic components (as in Fig. 3 b). This seems a not very strong restriction, also from the point of view of Computer Science: for example, in all LISP-like languages unit sequences can be neither defined nor constructed.

The recursive definition of domain $\mathcal{J}_3$ is

$$X \in \mathcal{L} - \{\llbracket \mathrm{nil} \rrbracket\} \Rightarrow \{X \in \mathcal{J}_3 \Leftrightarrow (\llbracket \mathrm{null} \rrbracket\ (\llbracket \mathrm{tail} \rrbracket\ X) \vee (\forall j{:}0 \leq j < \mathrm{lgh}\ X)$$

$$(\llbracket \mathrm{head} \rrbracket\ (\llbracket \mathrm{tail} \rrbracket^j\ X)) \in \mathcal{J}_3) \wedge (\exists\ n)(\#_3\, X = \hat{n}\,)\}$$

where

$$\#_3 \equiv \lambda x.\llbracket \mathrm{null} \rrbracket\ (\llbracket \mathrm{tail} \rrbracket\ x)\hat{0}(\mp(\#_3(\llbracket \mathrm{head} \rrbracket\ x))\ \ (\delta(\llbracket \mathrm{tail} \rrbracket\ x)))$$

$$\delta \equiv \lambda\ x.\llbracket \mathrm{null} \rrbracket\ (\llbracket \mathrm{tail} \rrbracket\ x)\ \ (K(\#_3(\llbracket \mathrm{head} \rrbracket\ x)))\ \ (\#_3 x)\ .$$

The combinator δ allows the correct counting of the number of leaves by $\#_3$ also in the case where forbidden unit sequences are produced occasionally by the counting process itself.

The example of Fig. 3 are written as:

a): $\langle x\rangle$ b'): $\langle\langle\langle v\rangle\rangle\langle w\rangle\rangle$ c): $\langle\langle\langle w\rangle\langle x\rangle\rangle\langle y\rangle\langle z\rangle\rangle$.

REMARK 3 With the exception of b') the tree structure reflects into the parenthesis structure irrespective of the labelling mode. This seems to be a good adequacy property.

FOOTNOTES

1. $K \equiv \lambda xy.x$ and we will assume, for any terms t and u and for any integer i,

$$t^{o}u = u, \quad t^{i+1}u = t(t^{i}u).$$

2. In the preceding and next realizations equality means strong equality i.e. having the rule $X = Y \Rightarrow \lambda xX = \lambda xY$.

ACKNOWLEDGEMENT

I wish to express my appreciation of the guidance and encouragement that prof. Curry has given me over the years.

REFERENCES

Böhm C. and Gross W. (1965), Introduction to the Cuch, in "Theory of Automata" (ed. E. Caianiello), Academic Press, 35-65.

Church A. (1941), "The calculi of lambda conversion", Ann. of Math. Studies 6, Princeton University Press.

Curry H.B. (1975), Representation of Markov Algorithms by combinators, in "The Logical Enterprise" (ed. A.R. Anderson, R.B. Marcus and R.M. Martin), Yale University Press.

Curry H.B. and Feys R. (1958), "Combinatory Logic", vol. I, North Holland Publishing Co.

Curry H.B., Hindley J.R. and Seldin J.P. (1972), "Combinatory Logic", vol. II, North Holland Publishing Co.

Coppo M. and Dezani-Ciancaglini M. (1977), Trees and Λ-terms, in "Les Arbres en Algebre et Programmation (ed. G. Jacobs), Debock, 91-120

Kearns J.T. (1969), Combinatory Logic with discriminators, JSL 34, 561-75.

McCarthy J. (1962), "The LISP 1.5 Programmer Manual", MIT Press.

Venturini Zilli M. (1965), λ-K-Formulae for vector operators, Bulletin of the International Computing Center 4, 157-74.

INFINITE TERMS AND INFINITE REDUCTIONS

Gerd Mitschke

Paosostrasse 105,
Pasing, München, W. Germany

Dedicated to H.B. Curry on the occasion of his 80th Birthday

0. INTRODUCTION, FUNDAMENTAL NOTIONS

Infinite typed λ-terms have been used at least since Tait's 1965 paper by proof theorists to formalize 'functional interpretations' and e.g. in Schwichtenberg (1973) they are used in connections with extensions of the Grzegorczyk-hierarchy. Also it is most likely that all the λ-calculus systems with extra recursion operators can be translated into systems with infinite terms having the same applications. Now a peculiarity of nearly all λ-calculus systems is that their consistency proofs work via the so-called Church-Rosser-theorem, which implies that one can obtain all information about the equality relation from the reducibility relation. In systems with only finite terms the reducibility can nicely and efficiently be described by 'reductions', a reduction being roughly a finite series of replacements in a term of left sides of instances of axioms by their corresponding right sides. This fails for infinite terms because in general a proof-step of the form (ω = set of natural numbers)

$$\frac{M_i \geq N_i \quad \text{for all} \quad i \varepsilon \omega}{\langle M_i | i\varepsilon\omega\rangle \geq \langle N_i | i\varepsilon\omega\rangle}$$

cannot be replaced by a *finite* series of replacements in $\langle M_i | i\varepsilon\omega\rangle$. Nevertheless it seems desirable to replace it by an 'infinite reduction'

$\langle M_0,M_1,\ldots,M_n,\ldots\rangle \to \langle N_0,M_1,\ldots,M_k,\ldots\rangle$
$\to \ldots \to \langle N_0,N_1,\ldots,N_k,M_{k+1},\ldots\rangle \to \ldots$
with $\langle N_i|i\varepsilon\omega\rangle$ as a 'limit'. But how shall one define the 'limit of an infinite reduction'? Surely not every infinite reduction should have a limit, e.g. terms without a normal form are always starting points of infinite reductions which should have no limit.

The difficulty just described is probably the reason why in the existing literature no attempt is made to define infinite reductions. It is the purpose of this paper to fill the gap and to study how much of the results for finite terms can be carried over to infinite terms and we shall see that most of the fundamental results are valid, sometimes in a modified form. The problems just mentioned were brought to my attention by Roger Hindley who told me that he in turn was asked about infinite reductions by Stan Wainer.

For our general considerations we work in a type free system as described e.g. in Maass (1974). In our formalism we use a set Var = $\{v_i|i\varepsilon\alpha\}$ of (indexed) variables, here α is an ordinal to be specified later. A set Num = $\{n|n\varepsilon\omega\}$ of numerals, the symbols λ,(,) and the sequence former $\langle\ |\ \rangle$.

0.1 DEFINITION Terms are inductively defined by the formation rules

T0) All variables and numerals are terms

T1) If M,N are terms, so is (MN)

T2) If M is a term, $x\varepsilon$Var, then λxM is a term

T3) If for all $i\varepsilon\omega$ M_i are terms then the infinite sequence $\langle M_i|i\varepsilon\omega\rangle$ is a term.

As usual we associate parentheses to the left; $\langle M_i\rangle$ sometimes stands for $\langle M_i|i\varepsilon\omega\rangle$. As we shall be concerned only with reducibility we use = to denote syntactical identity. The sets of free and bound variables and the subterms in a term M are defined in the obvious way and denoted FV(M) resp BV(M).

If we assume a continuum of variables, i.e. $\alpha = \omega_1$, we can carry over the definition of substitution from the finite case, namely

0.2 DEFINITION $[N/x]M$ is inductively defined by

(i) $$[N/x]y = \begin{cases} N & \text{if } x = y \\ y & \text{if } x \neq y \end{cases} \quad \text{for } y \in Var \cup Num \text{ and } x \in Var$$

(ii) $[N/x](MM') = ([N/x]M\ [N/x]M')$

(iii) $[N/x]\langle M_i | i\in\omega\rangle = \langle [N/x]M_i | i\in\omega\rangle$

(iv) $$[N/x]\lambda yM = \begin{cases} \lambda yM & \text{if } x = y \\ \lambda y[N/x]M & \text{if } y\notin FV(N) \text{ or } x\notin FV(M) \\ \lambda z[N/x][z/y]M & \text{if } y\in FV(N) \text{ and } x\in FV(M) \end{cases}$$

where z is the variable with smallest index which occurs free neither in M nor in N. Using this definition we naturally need a continuum of variables. If one wants only countably many variables one has to have $[N/x]\lambda yM$ undefined in the last case of (iv). But contrary to the finite case one cannot just change the bound y in λyM and get a $\lambda zM'$ such that $[N/x]\lambda zM'$ is defined, because it might happen that all variables occur free in M. Fortunately in most applications only terms with finitely many free variables are of interest. In the sequel I assume $\alpha = \omega_1$.

The reducibility relation $\geq$ between terms is now defined by the following axioms and rules

(ρ) $M \geq M$

(α) $\lambda xM \geq \lambda y[y/x]M$ if $y\notin FV(M)$

(β) $(\lambda xM)N \geq [N/x]M$

(γ) $(\langle M_i\rangle P)Q \geq \langle (M_iQ)\rangle P$

(π) $\langle M_i\rangle n \geq M_n$ for all $n\in\omega$

(τ) $\dfrac{M\geq N,\ N\geq L}{M\geq L}$

(μ) $\dfrac{M\geq N}{ZM \geq ZN}$ for all Z

(ν) $\dfrac{M\geq N}{MZ \geq NZ}$ for all Z

(κ) $\dfrac{\forall i\in\omega\ M_i\geq N_i}{\langle M_i\rangle \geq \langle N_i\rangle}$

$(\xi)\ \dfrac{M \geq N,\ x \in Var}{\lambda x M \geq \lambda x N}$

For this reducibility relation Maass (1974) proved the Church-Rosser-theorem using a method which generalizes the proof of Tait for the case of finite terms. A different proof will be indicated here using infinite reductions which are to be defined below.

From now on we shall not mention axiom (α) i.e. changes of bound variables. First we define subrelations $\geq_\alpha$ (α an ordinal) of $\geq$ where $M \geq_\alpha N$ will roughly mean that there is a proof tree for $M \geq N$ with depth $\leq \alpha$.

0.3 DEFINITION

$M \geq_0 N$ iff $M = N$

$M \geq_1 N$ iff $M \geq N$ is an axiom

$M \geq_{\alpha+1} N$ iff $\begin{cases} M \geq_\alpha N \text{ or } M \geq N \text{ can be deduced by one of the rules} \\ (\tau),\ (\mu),\ (\nu),\ (\kappa),\ (\xi),\ (\pi) \text{ from premises of the} \\ \text{form } A \geq_\beta B \text{ where } \beta \leq \alpha. \end{cases}$

$M \geq_\lambda N$ iff $\exists \alpha < \lambda\ M \geq_\alpha N$; for limits λ.

Easy to see is

0.4 LEMMA $M \geq N$ iff $M \geq_{\omega_1} N$.

In the following we make use of the *context-notation*:

C[ ] will denote a term with exactly one hole,

C[M] the same term with M in place of the hole.

The exact definition of this is obvious. Terms of the forms $(\lambda x M)N$, $(\langle M_i \rangle P)Q$, $\langle M_i \rangle n$ will be called β-,γ-,π-*redexes* resp. and $[N/x]M$, $\langle M_i Q \rangle P$, M_n their corresponding *contracta*. We write $M \xrightarrow{R} N$ iff R is a redex with contractum R' and $M = C[R]$, $N = C[R']$ for some context C[ ].

We are now able to give the promised precise definition of an infinite reduction. First we associate to every proof for $M \geq N$ a sequence of pairs $(M_i, R_i)_{i \leq \beta}$, β an ordinal, such that R_i is a redex occurrence in M_i and for all $i < \beta$ $M_i \xrightarrow{R_i} M_{i+1}$, $R_\beta = \phi$, $M_0 = M$, $M_\beta = N$. These sequences will be the reductions from M to N and β will be called the length of the reduction. To make things easier we will often write reductions in the form

$$M_0 \xrightarrow{R_0} M_1 \xrightarrow{R_1} M_2 \to \ldots \to M_\alpha \xrightarrow{R_\alpha} M_{\alpha+1} \to \ldots \to M_\beta$$

when this cannot be mistaken.

0.5 Construction of reductions

Let $M \geq N$; by lemma 0.4 there exists an α such that $M \geq_\alpha N$, the reductions will be constructed by induction on α. We write $\sigma : M \to N$ if σ is a reduction from M to N.

$\underline{\alpha = 0}$: $M = N$, $\beta = 0$, $M_0 = M$, $R_0 = \phi$.

$\underline{\alpha = 1}$: $\beta = 1$, $M_0 = M$, $R_0 := M_0$, $M_1 := N$.

$\underline{\alpha \to \alpha+1}$: Here we have to consider several cases.

Case 1. $M \geq_{\alpha+1} N$ is obtained from $M' \geq_\alpha N'$ by application of rule (μ), and let $M = ZM'$, $N = ZN'$. By induction hypothesis we have already constructed a reduction

$$\sigma' : M' = M_0' \xrightarrow{R_0'} M_1 \xrightarrow{R_1} \ldots \to M' = N'$$

from M' to N'. The new reduction from M to N is then

$$\sigma : M = M_0 = ZM_0' \xrightarrow{R_0} ZM_1' \xrightarrow{R_1} \ldots \to ZM_\beta' = N,$$

where R_i in ZM_i' is the same redex occurrence as R_i' in M_i'. We call the new reduction $\sigma = Z\sigma'$ and its length is β.

Case 2. Rules (ν) and (ξ) are treated analogously and the new constructed reductions are called $\sigma = \sigma' Z$ respectively $\sigma = \lambda x \sigma'$.

Case 3. $M \geq_{\alpha+1} N$ is obtained from $M \geq_\alpha L \geq_\alpha N$ by application of rule (τ). Let $\sigma_1 : M \to L$, $\sigma_2 : L \to N$ be reductions of lengths β_1, β_2. Then $\sigma : M \to L$ is constructed by simple juxtaposition of σ_1 and σ_2; $\sigma = \sigma_1 + \sigma_2$ and its length is $\beta_1 + \beta_2$.

Case 4. $M \geq_{\alpha+1} N$ is obtained from $M_k \geq_\alpha N_k$ ($k \varepsilon \omega$) by application of rule (κ). Let for each $k \varepsilon \omega$

$$\sigma_k : M_k = M_{k0} \xrightarrow{R_{k0}} M_{k1} \xrightarrow{R_{k1}} \ldots \to M_{k\beta_k} = N_k$$

be a reduction from M_k to N_k of length β_k. The reduction σ from $M = \langle M_k | k\epsilon\omega \rangle$ to $N = \langle N_k | k\epsilon\omega \rangle$ will have length $\sum_{k\epsilon\omega} \beta_k$ and is constructed in the obvious way, namely

$$
\begin{array}{l}
L_0 = \langle M_0, M_1, \ldots \rangle \\
\big\downarrow \; R_0 \text{ is } R_{00} \text{ in } M_0 \\
L_1 = \langle M_{01}, M_1, \ldots \rangle \\
\big\downarrow \; R_1 \text{ is } R_{01} \text{ in } M_{01} \\
L_2 = \langle M_{02}, M_1, \ldots \rangle \\
\big\downarrow \\
\vdots \\
\big\downarrow \\
L_{\sum_{n=0}^{k} \beta_n + i} = \langle M_{0\beta_0}, M_{1\beta_1}, \ldots, M_{k\beta_k}, M_{k+1\ i}, M_{k+2}, \ldots \rangle \\
\big\downarrow \; R_{\sum_{n=0}^{k} \beta_n + i} \text{ is } R_{k+1\ i} \text{ in } M_{k+1\ i} \text{ for } i<\beta_{k+1} \\
\vdots \\
\big\downarrow \\
L_{\sum_{n\epsilon\omega} \beta_n} = \langle N_0, N_1, \ldots, N_k, \ldots \rangle
\end{array}
$$

We write $\sigma = \langle \sigma_k | k\epsilon\omega \rangle$.

Finally a 'non-terminating' sequence of pairs $(M_i, R_i)_{i<\lambda}$, λ a limit ordinal, is called a reduction iff for all $\alpha<\lambda$ $(M_i, R_i)_{i\leq\alpha}$ is a reduction from M_0 to M_α; again we call λ the length of the reduction. Note the difference between terminating and non-terminating reductions of limit length λ.

The reader should convince himself that the above definition of reductions appropriately formalizes the intuitive notions.

We sometimes write $M \overset{\sigma}{\to} N$ for $\sigma:M\to N$ and $M \underset{\alpha}{\to} N$ if there exists a reduction of length α from M to N; $|\sigma|$ denotes the length of σ. The notations from the above constructions can be generalized to arbitrary reductions, moreover we define $C[\alpha]$ for any context $C[\ ]$ and reduction σ in the obvious way. We are now going to look how long reductions can be.

0.6 DEFINITION For any ordinal $\alpha<\omega_1$ we define $m(\alpha):=\sup\{\beta|\exists M,N(M\geq_\alpha N \text{ and } M \underset{\beta}{\to} N)\}$

0.7 LEMMA. m is continuous and $m(0) = 0$, $m(1) = 1$, $m(n+1) = \omega^n$ $(1\leq n<\omega)$, $m(\alpha) = \omega^\alpha$ $(\alpha\geq\omega)$.

Proof. The continuity of m follows immediately from the definition. The equations $m(0) = 0$, $m(1) = 1$ are obvious. Now let for all $i\epsilon\omega$ $M_i\geq_\alpha N_i$ be such that $\sigma_i:M_i\to N_i$ are reductions of length $\leq m(\alpha)$, and whose lengths have $m(\alpha)$ as supremum. (If $m(\alpha)$ is not a limit ordinal, we can make all $M_i\geq_\alpha N_i$ the same: and if $m(\alpha)$ is a limit ordinal, we use the fact that all countable limit ordinals are cofinal with ω.) We have $\langle M_i\rangle \geq_{\alpha+1} \langle N_i\rangle$ and $\langle\sigma_i\rangle:\langle M_i\rangle \to \langle N_i\rangle$ and obviously $|\langle\sigma_i\rangle| = m(\alpha)\cdot\omega$ and no longer reductions can be produced from reductions with length $\leqslant m(\alpha)$. Thus $m(\alpha+1) = m(\alpha)\cdot\omega$ and this proves the lemma.

0.8 COROLLARY. Whenever $M \geqslant N$ then there is a reduction of length $< \omega_1$ from M to N.

This corollary is the counterpart of the (trivial) fact that in the finite λ-calculus every terminating reduction is finite.

1. TRANSLATION OF RESULTS FROM THE FINITE CASE

It is easy (and left to the reader) to define the *residuals* in N of a redex occurrence in M after a reduction $\sigma:M\to N$. The only difficulty occurs in the situation of a γ- and a π-redex 'close together'

$$(\langle M_i\rangle n)Q \to \langle M_iQ\rangle n,$$

here we assume that $\langle M_iQ\rangle n$ is the residual of $\langle M_i\rangle n$. Note that the residuals of a redex are always redexes of the same kind and that a redex occurrence R in N is residual of at most one redex in M.

The following lemma on the form of terminating limit reductions will be useful

1.1 LEMMA. Let λ be a limit ordinal and $\sigma: M \to N$ a reduction of length λ. Then there is a context $C[\ ]$, reductions $\tau_i: M_i \to N_i\ (i \in \omega)$ with $|\tau_i| \geq 1$ for infinitely many $i \in \omega$, such that σ is of the form $M \xrightarrow{\sigma'} C[\langle M_i | i \in \omega \rangle] \xrightarrow{C[\langle \tau_i \rangle]} C[\langle N_i | i \in \omega \rangle] = N$.

Proof. An easy induction on the construction of σ.

As an application we prove

1.2 LEMMA. Let $\sigma: M \to N$ and N be a finite term, then there is a finite reduction $\sigma': M \to N$.

Proof. Let $\sigma: M \to N$ be of minimal length. By 1.1, σ must have a length $\lambda + n$, $n \in \omega$, $n \geq 1$, λ a limit and without loss of generality N is the first finite term in the reduction. We show by induction on n that $|\sigma|$ is not minimal.

<u>n = 1</u>. Now σ must be of the form

$$\sigma:\ M \xrightarrow{\sigma'} C[\langle L_i \rangle k] \to C[L_k]$$

where $|\sigma| = \lambda$ and $C[\ ]$ is a finite context. By lemma 1.1 there is a context C' and reductions $\tau_i: M_i \to N_i$ with $|\tau_i| \geq 1$ for infinitely many $i \in \omega$, such that

$$M \xrightarrow{\sigma''} C'[\langle M_i \rangle] \xrightarrow{C'[\langle \tau_i \rangle]} C'[\langle N_i \rangle] = C[\langle L_i \rangle k] \to C[L_k].$$

Call $C'[\langle N_i \rangle] = C[\langle L_i \rangle k] = N'$. We have to check the relative positions of $\langle N_i \rangle$ and $\langle L_i \rangle k$ in N'.

Case 1. $\langle N_i \rangle$ and $\langle L_i \rangle k$ are disjoint. This is impossible since then $C[L_k]$ would be infinite.

Case 2. $\langle L_i \rangle k$ is a subterm of $\langle N_i \rangle$ is again impossible.

Case 3. $\langle N_i \rangle$ is a subterm of L_k, impossible.

Case 4. $\langle N_i \rangle$ is a subterm of L_j, $j \neq k$. Now σ is of the form

$$M \xrightarrow{\sigma''} C[\langle L_0, \ldots, C''[\langle N_i \rangle], \ldots, L_k, \ldots \rangle k]$$

$$\xrightarrow{C'\langle \tau_i \rangle} C[L_0, \ldots, C''[\langle M_i \rangle], \ldots, L_k, \ldots \rangle k] \to C[L_k]$$

and certainly

$\sigma''':M \to C[\langle L_0,\ldots,C''[\langle N_i\rangle],\ldots,L_k,\ldots\rangle k] \to C[L_k]$

is a shorter reduction from M to N than σ.

Case 5. $\langle L_i\rangle = \langle N_i\rangle$, clearly σ is not minimal.

$\underline{n \to n+1}$. Essentially the same argument works in the induction step. One has to trace the subterm $\langle N_i\rangle$ in the last n+1 steps of the reduction, look where it is in $C[\langle L_i\rangle k]$ and proceed similarly to the case n = 1.

The next thing to do naturally is to prove the Church-Rosser-theorem. Contrary to Maass (1974) we shall do it by generalizing the methods of Curry-Feys (1958) and Hindley (1969) to the system of infinite terms.

1.3 DEFINITION. A reduction σ starting in M is called a *development* of a set R of redex occurrences in M iff every redex contracted in σ is a residual of a redex in R, a development $\sigma:M\to N$ of R is *complete* iff R has no residual in N.

Notation: $R(M)$:= the set of all redex occurrences in M.

1.4 LEMMA. Every set R of redex occurrences in a term M has a complete development.

Proof. Induction on the formation of M:

1) M is a variable or numeral, $R = \phi$.

2) $M = \lambda xM'$. All elements of R must occur in M', let $\sigma:M'\to N'$ be a complete development of R in M', then $\lambda x\sigma$ is a complete development of R in M.

3) $M = \langle M_i | i\epsilon\omega\rangle$. All elements of R must occur inside the M_i's. Let $\sigma_i:M_i\to N_i$ be complete developments of $R\cap R(M_i)$. Then $\sigma = \langle\sigma_i | i\epsilon\omega\rangle:\langle M_i\rangle \to \langle N_i\rangle$ is a complete development of R.

4) M = XY. We consider the subcases

4a) M is not an element of R, trivial

4b) $M = \langle M_i\rangle n\epsilon R$. Let $\sigma:M_n\to N$ be a complete development of $R\cap R(M_n)$. Then

$\langle M_i\rangle n \overset{M}{\to} M_n \overset{\sigma}{\to} N$ is a complete development of R.

4c) $M = (\lambda xX)Y\epsilon R$. Let $\sigma:X\to X'$, $\tau:Y\to Y'$ be complete developments of $R\cap R(X)$, $R\cap R(Y)$ resp. then

$$(\lambda xX)Y \xrightarrow{(\lambda x\sigma)Y} (\lambda xX')Y \xrightarrow{(\lambda xX')\tau} (\lambda xX')Y' \xrightarrow{M} [Y'/x]X'$$

is a complete development.

4d) $M = (\langle M_i\rangle X)Y \in R$.

Subcase 1. $X = n$ is a numeral and $(\langle M_i\rangle n) \in R$. Let $\sigma: Y \to Y'$ be a complete development of $R \cap R(Y)$, $\sigma_n: M_n \to M'_n$ one of $R \cap R(M_n)$ then $(\langle M_i\rangle n)Y \to M_nY \to M'_nY \to M_n'Y'$ with the obvious reductions is a complete development of R.

Subcase 2. X is not a numeral or $\langle M_i\rangle X \notin R$. Let $\sigma_i: M_i \to M_i'$, $\delta: X \to X'$, $\tau: Y \to Y'$ be complete developments of $R \cap R(M_i)$, $R \cap R(X)$, $R \cap R(Y)$; then

$$(\langle M_i\rangle X)Y \xrightarrow{\langle \sigma_i\rangle XY} \langle M_i'\rangle XY \xrightarrow{\langle M_i'\rangle \sigma Y} \langle M'_i\rangle X'Y \xrightarrow{\langle M_i'\rangle X'\tau} \langle M'_i\rangle X'Y' \to \langle M'_iY'\rangle X'$$

is a complete development of R.

As in the finite case the above proof gives a method how to construct complete developments. The developments constructed have the following property: if R and S are redex occurrences in M and R is a subredex of S, then either (a residual of) S is not contracted earlier in the reduction than (a residual of) R or S is a π-redex. Developments with this property will be called *minimal* (Hindley (1969)). It is not essential that we allow π-redexes to be contracted before their subredexes, it only avoids unnecessary long reductions.

1.5 DEFINITION. (i) Two reductions $\sigma: M \to N$, $\tau: M \to L$ are (weakly) *equivalent* ($\sigma \sim \tau$) iff $N = L$; (ii) $\sigma: M \to N$, $\tau: M \to L$ are *strongly equivalent* in the sense of Hindley ($\sigma \sim_H \tau$) iff $\sigma \sim \tau$ and all redex occurrences in M have the same residuals in N after σ and τ.

1.6 LEMMA. Any two minimal complete developments of a set R of redex occurrences in M are strongly equivalent.

Proof. By induction on the formation of M, following the lines suggested by the proof of 1.4.

In the following I shall make tacit use of the fact that the essential properties of substitution remain valid for the infinitary system. This is due to the relatively simple extension of the operator $[N/x]$ to infinitary terms.

The key lemma for the proof of the Church-Rosser theorem is (cf. Hindley 1969)

1.7 LEMMA. Let $\sigma:M \to L$ be a minimal complete development of a set $\mathcal{R}$ of redex occurrences in M and $\tau:M \to N$ any development of $\mathcal{R}$. Then there is a minimal complete development $\tau':N \to L$ of the residuals of $\mathcal{R}$ after τ. (Figure 1.)

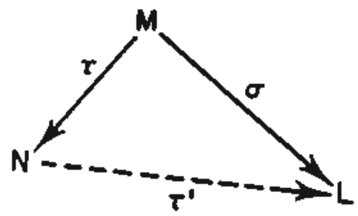


Fig. 1

Proof. By induction on the length of τ

$\underline{|\tau| = 1}$. This case is dealt with by an induction on the formation of M. Now $M \in Var \cup Num$ is trivial and in the induction step the only interesting case is: $M = M'M''$, M is a redex in $\mathcal{R}$, and $M \overset{M}{\to} N$.

Subcase 1. $M = (\lambda xX)Y$. We construct Figure 2,

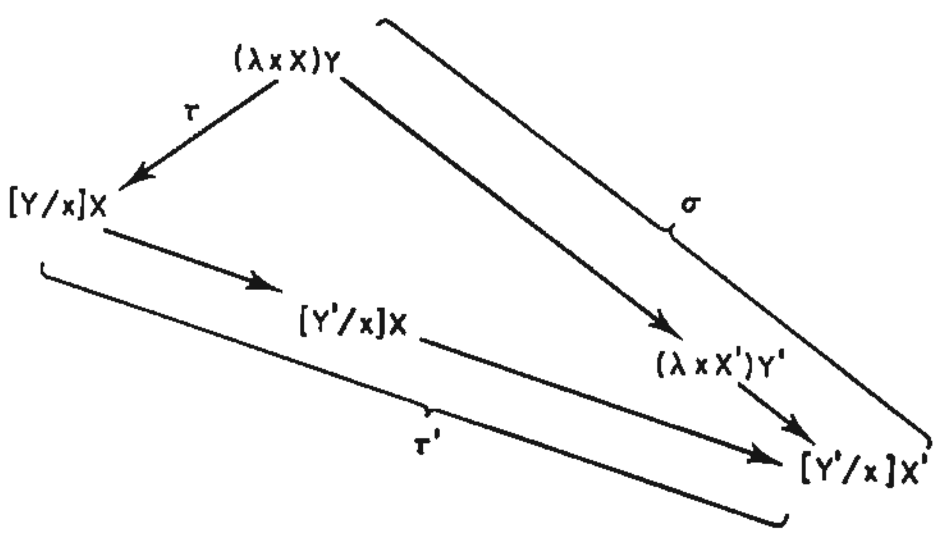


Fig. 2

where $X \to X'$, $Y \to Y'$ by a minimal complete development of $\mathcal{R} \cap \mathcal{R}(X)$ respectively $\mathcal{R} \cap \mathcal{R}(Y)$. Then τ' is constructed as indicated by first developing the residuals of $R\cap R(Y)$ in the substituted Y', and then by a minimal complete development of the residuals of $R\cap R(X)$ in $[Y'/x]X$. That the minimal complete developments

$$\tau'':[Y/x]X \to [Y'/x]X,$$
$$\tau''':[Y'/x]X \to [Y'/x]X'$$

exist can be proved by induction on the formation of X in the first case and by induction on the length of the minimal complete development $X \to X'$ in the second case.

Subcase 2. $M = ((\langle M_i\rangle X)Y)$, $\langle M_i\rangle X \notin R$, we construct Figure 3,

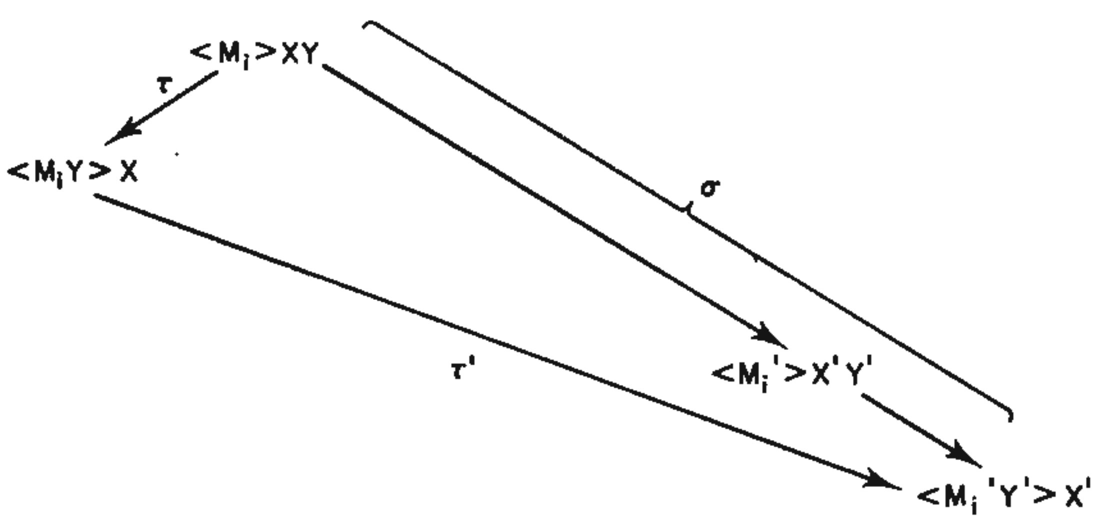


Fig. 3

in the obvious way.

Subcase 3. $M = \langle M_i\rangle nY$, $\langle M_i\rangle n\varepsilon R$ construct Figure 4

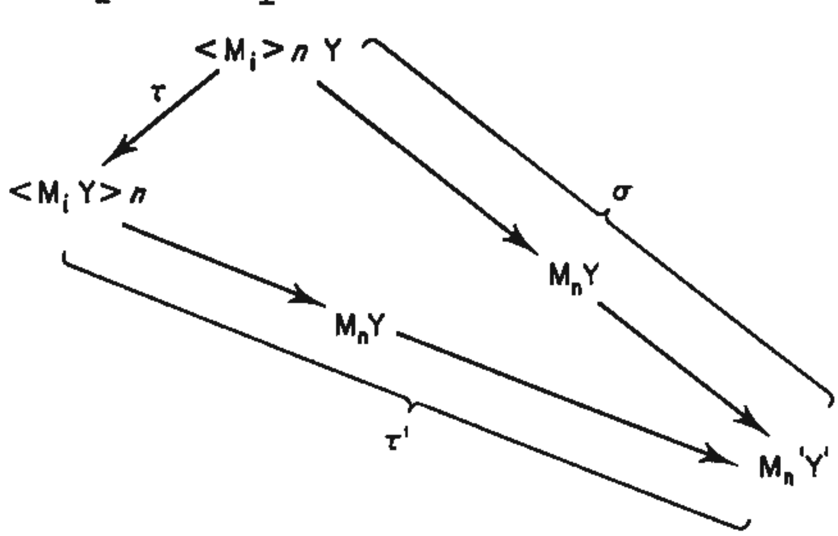


Fig. 4

in the obvious way.

Subcase 4. $M = \langle M_i \rangle n$ is trivial. Now we proceed with our main induction

$|\tau| = \alpha+1$. We have Figure 5,

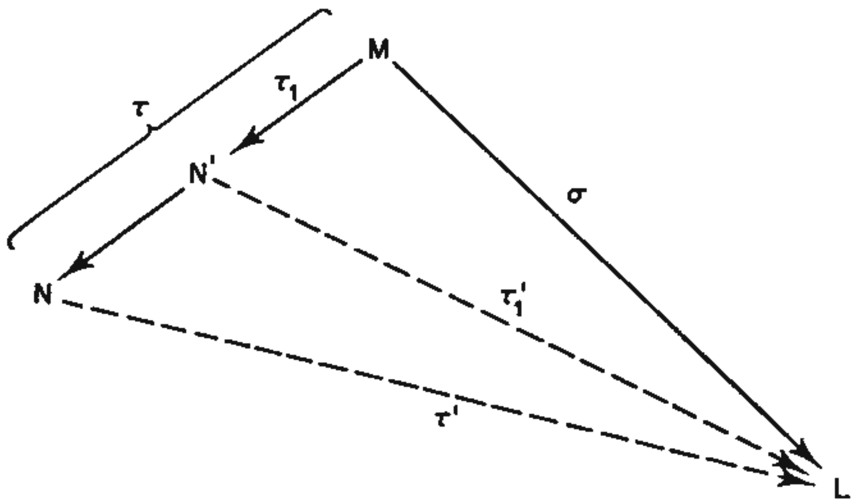


Fig. 5

where $N' \overset{R}{\to} N$ is a one step reduction. By induction hypothesis there is a minimal complete development $\tau'_1 : N' \to L$ of the residuals of R after τ_1, now R is in this set because τ is a development of R, so we can apply the case $|\tau| = 1$.

$|\tau| = \lambda$ a limit ordinal.

By lemma 1.1 there is a context C[ ] such that τ has the form $M \xrightarrow{\tau'} C[\langle M_i \rangle] \xrightarrow{C[\langle \tau_i \rangle]} C[\langle N_i \rangle] = N$ with $|\tau'| < |\tau|$ and $|\tau_i| < |\tau|$. Now we have the situation in Figure 6,

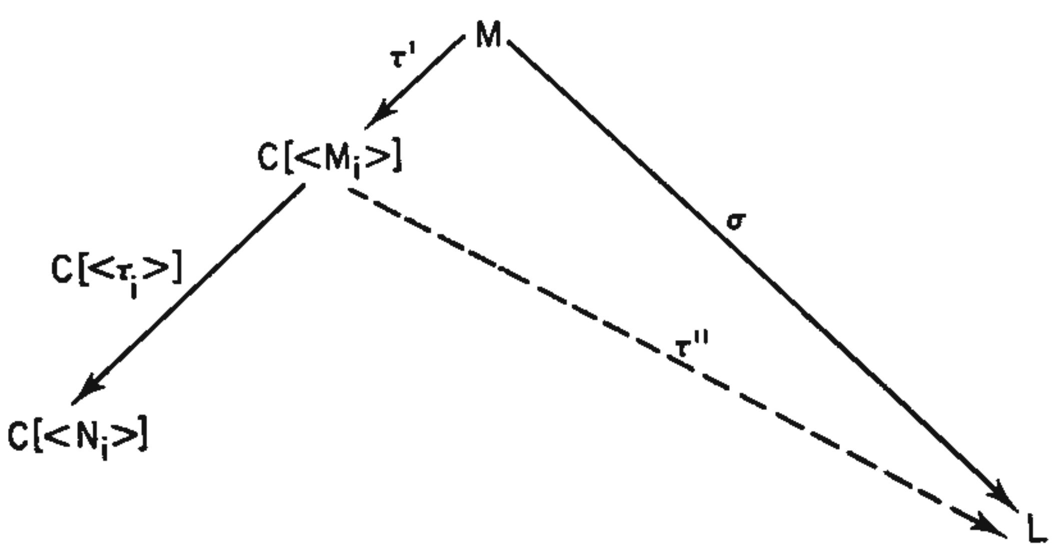


Fig. 6

where τ'' is a minimal complete development of the residuals of R after τ'. Now it is easy to choose τ'' such that it has the form

$$C[\langle M_i\rangle] \xrightarrow{C[\langle \sigma_i\rangle]} C[\langle L_i\rangle] \xrightarrow{\tau'''} L$$

where $\sigma_i : M_i \to L_i$ are minimal complete developments of the residuals of R after τ' in M_i. By induction hypothesis we have Figure 7:

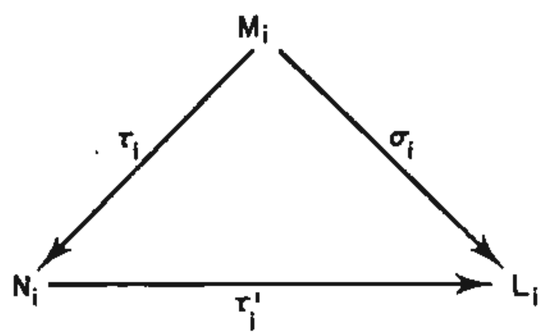


Fig. 7

and τ_i' are minimal complete developments of the residuals of R in N_i. Thus we finally find Figure 8:

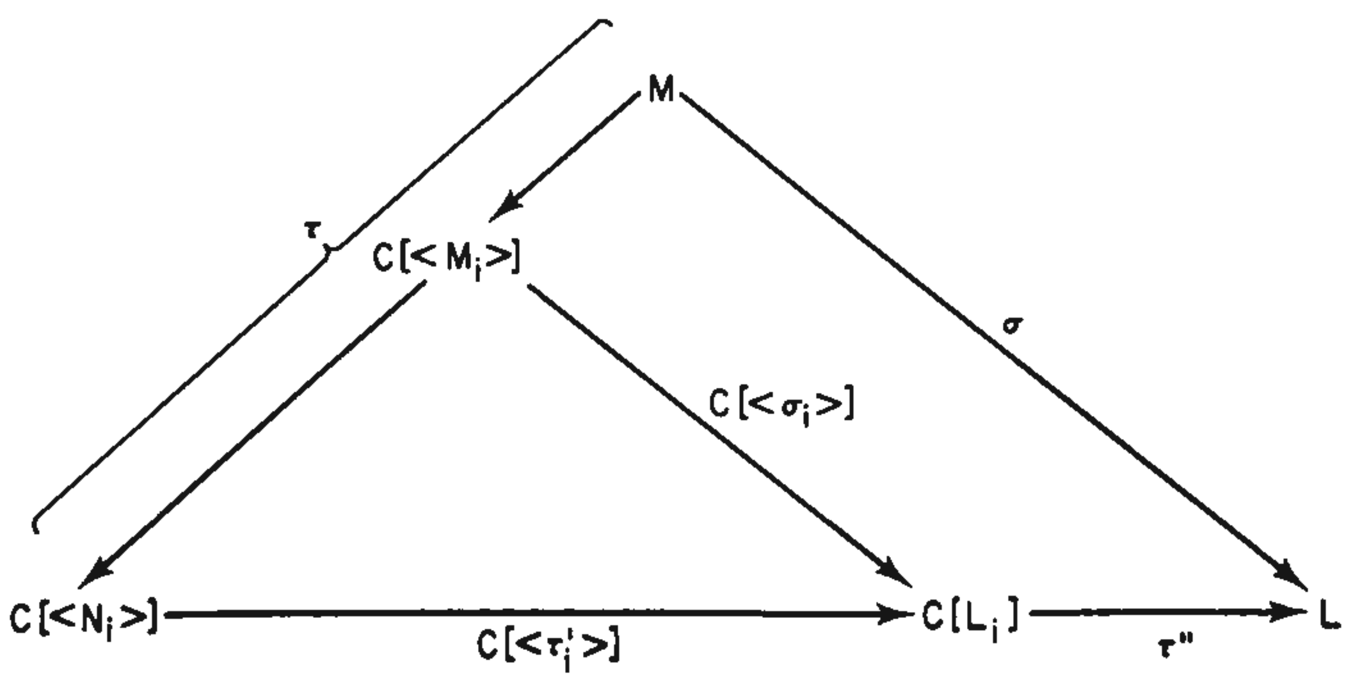


Fig. 8

Now $C[\langle\tau_i'\rangle]+\tau'''$ is a minimal complete development of the residuals of R after τ.

Addendum

By going through the proof of lemma 1.7 one can easily see that $\sigma \sim_H \tau+\tau'$.

As a corollary one gets like in the finite case the strong form of Curry's property (E).

Property (E^+)

Any set R of redex occurrences in a term X has a complete development and all complete developments of R are strongly equivalent.

In the finite λ-calculus the Church-Rosser-theorem is an immediate consequence of property (E^+). Due to the existence of reductions of limit length we have to do some more work in our infinitary case.

1.8 THEOREM (Church-Rosser theorem). If $\sigma{:}M\to N_1$, $\tau{:}M\to N_2$, then there exists N_3 and reductions $\sigma'{:}N_2\to N_3$, $\tau'{:}N_1\to N_3$.

Proof. The proof using (E^+) is by a double induction on $|\sigma|$ and $|\tau|$; we omit the rather lengthy but straightforward details. Naturally the theorem is also an immediate consequence of Maass' proof, Maass (1974).

REFERENCES

Curry, H.B., Feys, R., 1958, 'Combinatory Logic', I, North Holland Publishing Company, Amsterdam.

Hindley, J.R., 1969, An abstract form of the Church-Rosser-Theorem, *J. Symbolic Logic* 34, 545-560.

Maass, W., 1974, Church-Rosser-Theorem für λ-Kalküle mit unendlich langen Termen. *In* 'Proof Theory Symposium, Kiel 1974', Lecture Notes in Mathematics No. 500, Springer Verlag, Berlin-Heidelberg-New York.

Schwichtenberg, H., 1973, Einige Anwendungen von unendlichen Termen und Wortfunktionalen. Habilitationsschrift, Universität Münster.

Tait, W., 1965, Infinitely long terms of transfinite type. *In* 'Formal Systems and Recursive Functions' (ed. J.N. Crossley/ M.A.E. Dummett), North Holland Co.

NUMERATIONS, λ-CALCULUS & ARITHMETIC

Albert Visser

Mathematical Institute, Budapestlaan 6
3508 TA Utrecht, The Netherlands.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

ABSTRACT

Applications of complete and precomplete numerations (as introduced by Eršov) to term models of λ-calculus, structures associated with partial recursive functions and Peano arithmetic. Some results: a version of Gödel's First Incompleteness Theorem for λ-calculus (a consequence is that any countable p.o. can be embedded in the p.o. of RE λ-theories); a topological explanation of the Range Theorem of λ-calculus; representability of the partial recursive functions in any RE λ-theory and in any RE extension of PA.

§0 INTRODUCTION

In this paper I use some of Eršov's concepts to prove results about structures such as term models of λ-calculus. These results and their proofs are "coördinate free" i.e. they use (nearly) no specific properties of e.g. λ-calculus. As a consequence theorems from arithmetic carry over to λ-calculus and vice versa. For example there is a version of the Gödel-Rosser-Mostowski-Myhill-Kripke Theorem for RE λ-theories.

§1 introduces the relevant concepts from the theory of numerations. §2 is a small study of "intensional" phenomena in precomplete numerations by means of topological considerations.

E.g. I bring out the topological content of the well known Range Theorem of λ-calculus. §3 contains some consequences of the Fixed Point Theorem for precomplete numerations. §4 gives a construction which one could call: "How to Kleene a Curry", i.e. a construction to form a complete numeration from a precomplete numeration by identifying certain elements.
To read the paper a general background in recursion theory and λ-calculus (e.g. Rogers (1967) and Barendregt (1978)) should be sufficient. I took some care not to presuppose knowledge of Eršov (1973) and Eršov (1975). Actually, I think the present paper could be considered as an introduction to the idea of precomplete and complete numerations.

§1 INTRODUCTION TO SOME FUNDAMENTAL CONCEPTS

1.1 DEFINITION. A *numeration* γ is a pair (ν,S), where ν is a surjective function from $\mathbb{N}$ to S.
If $\gamma_1 = (\nu_1,S_1)$ and $\gamma_2 = (\nu_2,S_2)$ are numerations then μ is a *morphism* from γ_1 to γ_2 if μ is a function from S_1 to S_2 and if there is a recursive ψ s.t.

$$\begin{tikzcd} \mathbb{N} \arrow[r, "\psi"] \arrow[d, "\nu_1"] & \mathbb{N} \arrow[d, "\nu_2"] \\ S_1 \arrow[r, "\mu"'] & S_2 \end{tikzcd} \quad \text{commutes.}$$

Numerations with their morphisms form the *category of numerations*.

1.2 DEFINITION. If $\gamma = (\nu,S)$ is a numeration then we will say $m \sim_\gamma n$ for $\nu(m) = \nu(n)$.

1.3 DEFINITION. We will call a numeration *positive* if $\sim_\gamma$ is an

RE relation.

1.4 DEFINITION. A numeration $\gamma = (\nu,S)$ is called <u>precomplete</u> if for every partial recursive φ there is a total recursive ψ s.t. for every $n \in \mathrm{Dom}\,\varphi$ $\varphi(n) \sim_\gamma \psi(n)$.
We shall say that ψ <u>makes</u> φ <u>total</u> <u>modulo</u> γ.

1.5 DEFINITION. A numeration $\gamma = (\nu,S)$ is <u>complete</u> if there is an $a \in S$ s.t. for every partial recursive φ there is a total recursive ψ s.t. for every $n \in \mathrm{Dom}\,\varphi$ $\varphi(n) \sim_\gamma \psi(n)$ and for every $n \notin \mathrm{Dom}\,\varphi$ $\nu(\psi(n)) = a$.
We will call "a" a special element of γ.

1.6 EXAMPLES OF NUMERATIONS

1.6.1 TERM MODELS OF $\lambda\beta$-CALCULUS. Let Λ be the set of closed λ-terms. A λ-theory T is a consistent set of identities between elements of Λ closed under the rules of $\lambda\beta$-calculus. Suppose $\ulcorner\;\urcorner$ is an elementary bijective coding of Λ in $\mathbb{N}$ ("elementary" means: easy to give, primitive recursive etc.).
Let for $M \in \Lambda$: $[\![M]\!]_T := \{N \in \Lambda \mid (M=N) \in T\}$ and take $\mathbb{M}_T := \{[\![M]\!]_T \mid M \in \Lambda\}$. Define: $\lambda_T(\ulcorner M\urcorner) := [\![M]\!]_T$. Now the term model $\mathcal{M}_T$ of T is the numeration $(\lambda_T, \mathbb{M}_T)$.

1.6.1.1 THEOREM. $\mathcal{M}_T$ is precomplete.

Proof. Let φ be partially recursive. There is a representation F of φ in $\lambda\beta$-calculus. Let $E \in \Lambda$ be the universal constructor i.e. the λ-term s.t. $\lambda\beta \vdash E\underline{\ulcorner M\urcorner} = M$, for all $M \in \Lambda$. Take $\psi(n) := \ulcorner E(F\underline{n})\urcorner$, then it is easy to see that ψ makes φ total modulo $\mathcal{M}_T$. $\square$

1.6.2 THE RECURSIVE NATURAL NUMBERS. The recursive natural numbers are the 0-place partial recursive functions. Define:

$$\{n\}^0 :\cong \{n\} <> (\equiv \{n\}\, 0).$$

Let

$$\mathbb{N}^* := \mathbb{N} \cup \{\uparrow\}.$$

Further take

$$*(n) := \begin{cases} \{n\}^0 & \text{if } \{n\}^0 \downarrow \\ \uparrow & \text{else.} \end{cases}$$

The numeration N^* of recursive natural numbers is $(*,\mathbb{N}^*)$.

1.6.2.1 THEOREM. N^* is complete.

Proof. Let φ be partially recursive. Suppose φ has index p. Take an index p' s.t. $\{p'\}n \cong \{\{p\}n\}^0$ for all n.
Construct a primitive recursive S_0^1 with $\{S_0^1(m,n)\}^0 \cong \{m\}n$.
Take $\psi(n) \cong S_0^1(p',n)$. It is easy to see that ψ makes φ total modulo N^*. Moreover if $\varphi(n)\uparrow$ we have $*(\psi(n)) = \uparrow$. It is possible to prove that $\uparrow$ is the unique special element of N^*. □

1.6.2.2 REMARK. Define term models of λI-calculus in the obvious way. Call them M_T^I. Take T_0 the closure under the rules of λI of $\{M = N \mid M,N \text{ have no } \lambda I \text{ normal form}\}$. Then T_0 is consistent and $M_{T_0}^I$ is isomorphic (in the sense of the category of numerations) with N^*. No term model of λβ-calculus is isomorphic with N^*.

1.6.3 THE PARTIAL RECURSIVE FUNCTIONS. Let $\mathbb{P}$ be the set of partial recursive functions. Let $K(e) := (\lambda x \in \mathbb{N}.\{e\}x)$. Then $P := (K,\mathbb{P})$ is complete with as unique special element the nowhere defined function. (λ is an informal λ.)

1.6.4 THE RE SETS. Let $\mathbb{R}$ be the set of RE sets. Let $\Pi(e) := W_e$ (W_e is the domain of the partial recursive function with index e). Then $R := (\Pi,\mathbb{R})$ is complete with as unique special element the empty set.

1.6.5 THE 'INTENSIONAL' PARTIAL RECURSIVE FUNCTIONS. Let T be a theory, $T \supseteq PA$. (The language of T may extend the language of PA). Define: $[\![e]\!]^T := \{f \in \mathbb{N} \mid T \vdash \forall x(\{\underline{e}\}x \cong \{\underline{f}\}x)\}$.
Take $K^T(e) := [\![e]\!]^T$ and $\mathbb{P}^T := \{[\![e]\!]^T \mid e \in \mathbb{N}\}$.
Then $P^T = (K^T, \mathbb{P}^T)$ is the numeration of the partial recursive functions intensional in T.

1.6.5.1 THEOREM. P^T is precomplete.

Proof (sketch). Let φ be partially recursive. Suppose p is an index of φ. Define: $\psi(n) := \Lambda k . \{\{p\} n\}k$, where $(\Lambda k . \{\{p\} n\}k)$ is an index q s.t. $PA \vdash \forall x\{\underline{q}\}x \cong \{\{\underline{p}\}\underline{n}\}x$.
$(\Lambda k . \{\{p\} n\}k)$ can be found in a primitive recursive way from p and n. Then ψ makes φ total modulo P^T. □

1.6.6 THE Σ_n^0-FORMULAE WITH FIXED FREE VARIABLES OF A THEORY EXTENDING PA.

Let T be a consistent theory $T \supseteq PA$. Let $\Sigma_n^0(x_0,\ldots,x_{k-1})$ be the class of Σ_n^0 formulae (not necessarily in prenex normal form) of the language of PA with free variables among $x_0,\ldots,x_{k-1}$. We shall abbreviate $(x_0,\ldots,x_{k-1})$ to $\vec{x}$. Take $\ulcorner\,\urcorner$ to be an elementary bijective coding of $\Sigma_n^0(\vec{x})$. Define for $A(\vec{x}) \in \Sigma_n^0(\vec{x})$:

$$[\![A(\vec{x})]\!]^T_{n,\vec{x}} := \{B(\vec{x}) \in \Sigma_n^0(\vec{x}) \mid T \vdash \forall\vec{x}(A(\vec{x}) \leftrightarrow B(\vec{x}))\}$$

and

$$\mathrm{sig}^T_{n,\vec{x}}(\ulcorner A(\vec{x})\urcorner) := [\![A(\vec{x})]\!]^T_{n,\vec{x}}.$$

Put:

$$\mathrm{Sig}^T_{n,\vec{x}} := \{[\![A(\vec{x})]\!]^T_{n,\vec{x}} \mid A(\vec{x}) \in \Sigma_n^0(\vec{x})\}.$$

Then

$$\mathcal{S}ig^T_{n,\vec{x}} := (\mathrm{sig}^T_{n,\vec{x}}, \mathrm{Sig}^T_{n,\vec{x}})$$

is the numeration of $\Sigma_n^0(\vec{x})$ formulae of T.

1.6.6.1 THEOREM. $\mathcal{S}^T_{n,\vec{x}}$ is precomplete.

Proof. Consider φ partially recursive. Let p be an index of φ.

It is well known that there exists a $\Sigma^0_n(y,\vec{x})$ predicate $T_n(y,\vec{x})$ s.t. for all $A(\vec{x}) \in \Sigma^0_n(\vec{x})$: $PA \vdash \forall\vec{x}(A(\vec{x}) \leftrightarrow T_n(\underline{\ulcorner A(\vec{x})\urcorner}, \vec{x}))$.
Now take:

$$\psi(m) :\cong (\ulcorner \exists y\{\underline{p}\}\underline{m} \cong y \;\&\; T_n(y,\vec{x})\urcorner).$$

If $\varphi(m) \cong \ulcorner A(\vec{x})\urcorner$ we have:

$$T \vdash \forall\vec{x}(\exists y\{\underline{p}\}\underline{m} \cong y \;\&\; T_n(y,\vec{x}) \leftrightarrow T_n(\ulcorner A(\vec{x})\urcorner,\vec{x}) \leftrightarrow A(\vec{x})).$$

So ψ makes φ total modulo $Sig^T_{n,\vec{x}}$.

1.7 REMARK. Definition 1.1-1.5 are from Eršov (1973).
The expression "making φ total modulo γ" is new. That $M_{\lambda\beta}$ is precomplete was pointed out to me by Henk Barendregt.

§2 A LITTLE EXCURSION INTO TOPOLOGY

2.1 DEFINITION OF $\mathcal{O}_\gamma$. Let $\gamma = (\nu,S)$ be a numeration. Let $\mathcal{B}_\gamma := \{S_0 \subseteq S \mid \nu^{-1}(S_0) \text{ in } \Pi^0_1\}$. Clearly $\mathcal{B}_\gamma$ is a basis for a topology. Call this topology $\mathcal{O}_\gamma$.

2.2 THEOREM. Morphisms are continuous.

Proof. Trivial. □

2.3 REMARK. As far as I know this topology doesn't occur in the literature. In Eršov (1975), Eršov uses a topology based on Σ^0_1 instead of Π^0_1 sets. But doing this one loses Thm. 2.5.

2.4 ILLUSTRATIVE EXAMPLE. The topology $\mathcal{O}_\gamma$ is meant to capture something of the idea of nearness w.r.t. information content. The following example purports to illustrate this.
Let $(T_i)_{i\in\mathbb{N}}$ be a sequence of consistent RE theories containing PA s.t. $T_{i+1} \subseteq T_i$ and $\bigcap_{i\in\mathbb{N}} T_i = PA$. Suppose $(A_i)_{i\in\mathbb{N}}$ is a sequence of Σ^0_1-sentences s.t. $T_i \vdash A_i$. Consider $Sig^{PA}_{1,<>}$ with topology $\mathcal{O}_{Sig^{PA}_{1,<>}}$, then $[\![\underline{0} = \underline{0}]\!]^{PA}_{1,<>}$ is the unique limit of $([\![A_i]\!]^{PA}_{1,<>})_{i\in\mathbb{N}}$.

2.4.1 REMARK. There is a sequence $(T_i)_{i \in \mathbb{N}}$ of RE theories containing PA s.t. $T_{i+1} \subsetneq T_i$ and $\bigcap_{i \in \mathbb{N}} T_i = PA$.

Proof of remark. It can be shown that for any RE theories U,V with $U \supsetneq V \supseteq PA$ there is a sentence A s.t. $V \subsetneq (V+A) \subsetneq U$ (see §3 of this paper).

Let T be RE, $T \supsetneq PA$. Let $B_0, B_1, \ldots$ enumerate the sentences of L_T. Define: $T_0 := T$; B_{i_k} is the first sentence s.t.

$$PA \subsetneq (PA+B_{i_k}) \subsetneq T_k; \; T_{k+1} := PA + B_{i_k}.$$

It is easy to see that the construction works and satisfies the desiderata. □

Proof of 2.4. First we show that $[\![\underline{0} = \underline{0}]\!]^{PA}_{1,<>}$ is a limit. Suppose it is not. Then there is an RE set U closed under $\sim_{Sig^{PA}_{1,<>}}$ s.t. for infinitely many i, $\ulcorner A_i \urcorner \in U$ and $\ulcorner \underline{0} = \underline{0} \urcorner \notin U$.

Let B(x) be a $\Sigma^0_1(x)$-formula which represents U in PA.

Let $C :\equiv (\exists x\, B(x) \wedge T_1(x))$. Then: $\forall i \in \mathbb{N}\; T_i \vdash C$, but clearly $PA \nvdash C$ (because PA satisfies the existence property for Σ^0_1-sentences). Contradiction.

For unicity: assume $[\![A]\!]^{PA}_{1,<>}$ is also a limit and $[\![A]\!]^{PA}_{1,<>} \neq [\![\underline{0} = \underline{0}]\!]^{PA}_{1,<>}$. Then there is an N s.t. $T_N \nvdash A$. But $O_A := \{[\![A']\!]^{PA}_{1,<>} \mid T_N \nvdash A'\}$ is open and $[\![A]\!]^{PA}_{1,<>} \in O_A$.

But : $\forall i \geq N\; [\![A_i]\!]^{PA}_{1,<>} \notin O_A$ contradiction.

2.5 THEOREM. If γ is precomplete then $\mathcal{O}_\gamma$ is hyperconnected i.e. every two non-empty open sets intersect.

Proof. Clearly it is sufficient to prove the theorem for elements of $\mathcal{B}_\gamma$. So going over to complements, we have to show: for any RE sets U,V closed under $\sim_\gamma$, if $U \cup V = \mathbb{N}$ then $U = \mathbb{N}$ or $V = \mathbb{N}$. Suppose U,V RE, closed under $\sim_\gamma$ and $U \cup V = \mathbb{N}$. We are done if $U \subseteq V$ or $V \subseteq U$. So assume $n_U \in U \setminus V$ and $n_V \in V \setminus U$, in order to derive a contradiction. Let Σ^+, $\Sigma^- \subseteq \mathbb{N}$ be two RE, recursively inseparable sets. Define the partial recursive function φ as follows:

$$\varphi(n) :\cong \begin{cases} n_U & \text{if } n \in \Sigma^+ \\ n_V & \text{if } n \in \Sigma^- \\ \uparrow & \text{else} \end{cases}$$

Let ψ make φ total modulo γ.
Consider:

$$W_U := \{m \mid \psi(m) \in U\}$$
$$W_V := \{m \mid \psi(m) \in V\}$$

Then we have:

1. $W_U \cup W_V = \mathbb{N}$, since ψ is total and $U \cup V = \mathbb{N}$.
2. W_U, W_V are RE.
3. $\Sigma^- \cap W_U = \Sigma^+ \cap W_V = \emptyset$.

By familiar arguments we can construct a recursive set which separates Σ^+ and Σ^-, contradiction.

2.6 COROLLARY. Let γ be precomplete. Let $S_0 \subseteq S$ be discrete in $\mathcal{O}_\gamma$. (S_0 is discrete if the induced topology on S_0 is discrete). Let μ be continuous from $S \to S$ and $\mu(S) \subseteq S_0$. Then μ is constant on S.

Proof. This is an elementary topological fact. □

2.7 DEFINITION. Let $\gamma = (\nu, S)$. $S_0 \subseteq S$ is RE without repetitions if *(i)* S_0 is finite and for every $s_0 \in S_0 \{n \mid \nu(n) = s_0\}$ is RE.
or
(ii) There is a total recursive ψ s.t. $\nu \circ \psi$ is injective, $\nu \circ \psi(\mathbb{N}) = S_0$ and $m \sim_\gamma \psi(n)$ is an RE relation in m and n.

2.8 FACT. Let $\gamma = (S, \nu)$. If $S_0 \subseteq S$ is RE without repetitions then S_0 is discrete.

Proof. Trivial. □

2.9 FACT. Consider M_T. Define: $\mu_M : M_T \to M_T$ as $\mu_M([\![N]\!])_T = [\![MN]\!]_T$. Then μ_M is a morphism.

Proof. Trivial. □

2.10 APPLICATIONS. Consider $M_{\lambda\beta}$. One easily sees that

(i) All finite $S_0 \subseteq \mathbb{M}_{\lambda\beta}$ are RE without repetitions.

(ii) $\mathbb{NF} = \{[\![M]\!]_{\lambda\beta}$ M is in normal form$\}$ is RE without repetitions.

Remember that morphisms are continuous; combining 2.6, 2.8, 2.9, we find:

(i') If there is an $N_1,\ldots,N_k$ s.t. for any $P \in \Lambda$, $MP =_{\lambda\beta} N_i$ for some $1 \leqslant i \leqslant k$ then there is an N s.t. for all P: $MP =_{\lambda\beta} N$.

(ii') Suppose for any P, MP has a normal form, then there is a normal form N s.t. for all P: $MP =_{\lambda\beta} N$.

Of course *(i')* is the range theorem of λ-calculus.

2.11 DEFINITION.

(i) C, the set of <u>contexts</u> in the language $L_{PA} + \Box + p$, is the smallest set s.t.

- *(a)* $A \in L_{PA} \Rightarrow A \in C$ (We allow that A contains free variables)
- *(b)* $p \in C$
- *(c)* $A,B \in C \Rightarrow A \wedge B,\ A \vee B,\ A \to B,\ \neg A \in C$
- *(d)* For any $x_i \in \mathrm{Var}$: $A \in C \Rightarrow \forall x_i A, \exists x_i A \in C$
- *(e)* $A \in C \Rightarrow \Box A \in C$.

(ii) We define $\mathrm{Sub} : L_{PA} \times C \to L_{PA}$ by:

- *(a)* $A \in L_{PA} \Rightarrow \mathrm{Sub}(D,A) = A$
- *(b)* $\mathrm{Sub}(D,p) = D$
- *(c)* $\mathrm{Sub}(D, A \mathbin{\Delta} B) = \mathrm{Sub}(D,A) \mathbin{\Delta} \mathrm{Sub}(D,B)$, where $\Delta \in \{\wedge,\vee,\to\}$; $\mathrm{Sub}(D, \neg A) = \neg\,\mathrm{Sub}(D,A)$.

(d) $\mathrm{Sub}(D, Qx_i A) = Qx_i\ \mathrm{Sub}(D,A)$. $Q \in \{\forall, \exists\}$.
We allow that x_i is among the free variables of D.

(e) $\mathrm{Sub}(D, \Box A(\vec{x})) = \mathrm{prov}(\ulcorner \mathrm{Sub}(D, A(\vec{x}))\urcorner, \vec{x})$,
where $\vec{x}$ contains all free variables of A in order of first occurrence.

We will write for $C \in \mathcal{C}$: $C[p]$ and for $\mathrm{Sub}(D,C)$: $C[D]$.

2.12 LEMMA. $PA \vdash A \leftrightarrow B \Rightarrow PA \vdash C[A] \leftrightarrow C[B]$.

Proof. Use: $PA \vdash \forall\vec{x}(A(\vec{x}) \leftrightarrow B(\vec{x})) \Rightarrow$
$PA \vdash \forall\vec{x}(\mathrm{Prov}(\ulcorner A(\vec{x})\urcorner, \vec{x}) \leftrightarrow \mathrm{Prov}(\ulcorner B(\vec{x})\urcorner, \vec{x}))$. □

2.13 APPLICATIONS.

2.13.1 Let $C[p], D[p] \in \mathcal{C}$. Suppose that for all $A \in \Sigma^0_1$ $PA \vdash C[A]$ or $PA \vdash D[A]$. Then

for all $A \in \Sigma^0_1$, $PA \vdash C[A]$,

or

for all $A \in \Sigma^0_1$, $PA \vdash D[A]$.

Proof. $\Gamma := \{A \in \Sigma^0_1 \mid PA \vdash C[A]\}$,

$\Delta := \{A \in \Sigma^0_1 \mid PA \vdash D[A]\}$.

Well $\Gamma \cup \Delta = \Sigma^0_1$; Γ, Δ RE; Γ, Δ closed under provable equivalence. Apply hyperconnectedness. □

2.13.2 Suppose $\Box C[p]$, $\Box D[p] \in \mathcal{C}$, where C and D contain no free variables. Then

for all $A \in \Sigma^0_1$ $PA \vdash (\Box C[A] \vee \Box D[A])$

implies

for all $A \in \Sigma^0_1$ $PA \vdash \Box C[A]$

or

for all $A \in \Sigma^0_1$ $PA \vdash \Box D[A]$.

Proof. Because:

$PA \vdash \Box C[A] \vee \Box D[A] \Rightarrow$

$\mathbb{N} \vdash \Box C[A] \vee \Box D[A] \Rightarrow$

$$\mathbb{N} \vdash \Box C[A] \text{ or } \mathbb{N} \vdash \Box D[A] \Rightarrow$$
$$PA \vdash \Box C[A] \text{ or } PA \vdash \Box D[A].$$

(The last step is because $\Box C[A]$, $\Box D[A]$ are Σ^0_1). Now apply 2.13.1. $\Box$

2.13.3 Let $C(x)[p] \in \mathcal{C}$. Suppose that for every $A \in \Sigma^0_1$, there is precisely one n s.t. $PA \vdash C(\underline{n})[A]$ then there is precisely one n_0 s.t. for all $A \in \Sigma^0_1$ $PA \vdash C(\underline{n_0})[A]$.

Proof. $\varphi(\ulcorner A \urcorner) :=$ (the unique n s.t. $PA \vdash C(\underline{n})[A]$) induces a morphism from $Sig^{PA}_{1,<>}$ to $N = \begin{array}{c}\mathbb{N}\\ \downarrow \mathrm{id}\\ \mathbb{N}\end{array}$; and 0_N is discrete.

$\Box$

2.13.4 REMARK. For a different proof of 2.13.2 see Boolos (1979) page 106.

§3 THE FIXED POINT THEOREM

3.1 REMARK. Let γ be precomplete. By definition there is for every partial recursive φ a total recursive ψ which makes φ total modulo γ. We can even show that there is a total recursive $\chi : \mathbb{N}^2 \to \mathbb{N}$ s.t. for every index e $\lambda n\, \chi(e,n)$ makes $\lambda n\{e\}n$ total modulo γ. For consider $\rho(z) := \{z_0\}z_1$. Let ρ^* make ρ total and take $\chi(x,y) := \rho^*(\langle x,y\rangle)$. We shall write $\{x\}^\gamma y$ for $\chi(x,y)$.

3.2 FIXED POINT THEOREM (Eršov). Let γ be precomplete, then for every partial recursive φ we can find an n (effectively from an index of φ) s.t. $\varphi(n)\downarrow \Rightarrow \varphi(n) \sim_\gamma n$.

Proof. (We certainly can afford the space).
Let p be an index of φ. Let q be an index of $\lambda x\,.\,\{p\}(\{x\}^\gamma x)$. Say $\{q\}^\gamma q \cong q^*$. Suppose $\{p\}q^*\downarrow$ then:

$$\{p\}q^* \cong \{p\}(\{q\}^\gamma q) \cong \{q\}q \sim_\gamma \{q\}^\gamma q \cong q^*. \qquad \Box$$

Alternative Proof. Let ψ make $\lambda x\{p\}(\{x\}x)$ total. Let r be an index of ψ. Let r^* be $\{r\}r$. Suppose $\{p\}r^*\downarrow$ then:

$$\{p\}r^* \cong \{p\}(\{r\}r) \sim_\gamma \{r\}r \cong r^*.$$

3.3 DEFINITION. We say that a sequence $(U_i)_{i\in\mathbb{N}}$ of subsets of $\mathbb{N}$ is <u>recursive</u> if there is an RE relation $R(i,k)$ s.t. $k \in U_i \Leftrightarrow R(i,k)$. r is an index of the sequence if r is an index of R.

3.4 DEFINITION. Let $R(x,\vec{y})$ be an RE relation, then we can write R as $\exists z\, R_0(z,x,\vec{y})$, where R_0 is recursive. Define:

$$\varepsilon x R(x,\vec{y}) : \cong (\mu u R_0(u_0,u_1,\vec{y}))_1.$$

So $\varepsilon x.R(x,\vec{y})$ gives an element of $\{x|R(x,\vec{y})\}$ is there is one. Note that ε depends on the choice of R_0.

3.5 INDEX AVOIDING THEOREM. Suppose γ is precomplete and $(V_i)_{i\in\mathbb{N}}$ is a recursive sequence s.t.

(i) Each V_i is closed under $\sim_\gamma$

(ii) $i \notin V_i$;

then we can find (effectively from an index of $(V_i)_{i\in\mathbb{N}}$) an i_0 s.t. $V_{i_0} = \emptyset$.

Proof. Take i_0 a fixed point of $\lambda i(\varepsilon n\, .\, n \in V_i)$. □

3.6 THE GÖDEL-ROSSER-MOSTOWSKI-MYHILL-KRIPKE THEOREM. Consider $Sig^{PA}_{n,\vec{x}}$. Let $(T_i)_{i\in\mathbb{N}}$ be a recursive sequence of theories s.t. $T_i \supseteq PA$. Let $(U_i)_{i\in\mathbb{N}}$ be a recursive sequence of codes of formulae (say the coding is " ") s.t. $U_i \subseteq L_{T_i}$ and $T_i \nvdash A$ for all "A" $\in U_i$. (We could say that T_i leaves U_i out). Then there is a $\Sigma^0_n(\vec{x})$ formula A_0 s.t.

$$\forall A \in \Sigma^0_n(\vec{x})\ \forall i \in \mathbb{N}\ \forall "B" \in U_i\ T_i + \forall\vec{x}(A_0 \leftrightarrow A) \nvdash B.$$

Proof. Apply the Index Avoiding Theorem. Remember $\ulcorner\urcorner$ is a bijective coding of $\Sigma^0_n(\vec{x})$ formulae of L_{PA} in the natural numbers. Take V_i of the theorem as follows:

$V_{\ulcorner A \urcorner} := \{\ulcorner C \urcorner \mid C \in \Sigma^0_n(\vec{x})$ *and* $\exists i \in \mathbb{N} \exists "B" \in U_i \ T_i + \forall \vec{x}(A \leftrightarrow C) \vdash B\}$. □

3.7 COROLLARY. Let $T \supseteq PA$ be consistent, RE. Then there is an infinite recursive sequence of Σ^0_1-sentences $B_0, B_1, \ldots$ s.t. for any sequence of Σ^0_1-sentences $C_0, C_1, \ldots$ $T + C_0 \leftrightarrow B_0, C_1 \leftrightarrow B_1, \ldots$ is consistent.

Proof. Apply 3.6 for $Sig^{PA}_{1,x}$ and $T_i = T$ and $U_i = \{"\underline{0} = \underline{1}"\}$. Let A_0 be the formula given by the theorem.
Take $B_0 = A_0(\underline{0})$, $B_1 = A_0(\underline{1}), \ldots$. It is sufficient to prove that for any n: $T + C_0 \leftrightarrow A_0(\underline{0}) + \ldots\ C_n \leftrightarrow A_0(\underline{n})$ is consistent.
Take $D(x) \equiv (x = \underline{0} \wedge C_0) \vee \ldots (x = \underline{n} \wedge C_n)$, then
$T + \forall x(D(x) \leftrightarrow A_0(x))$ is consistent. But
$T + \forall x(D(x) \leftrightarrow A_0(x)) \vdash C_0 \leftrightarrow A_0(\underline{0}), \ldots\ C_n \leftrightarrow A_0(\underline{n})$. □

3.8 THEOREM. Consider $M_{\lambda\beta}$. Let $(T_i)_{i \in \mathbb{N}}$ be a recursive sequence of $\lambda\beta$-theories. Let $(U_i)_{i \in \mathbb{N}}$ be a recursive sequence of non empty sets of codes of identities of closed λ-terms (say coded by " ") s.t.

$$\forall i \in \mathbb{N}\ \forall "P = Q" \in U_i\ \ T_i \nvdash P = Q.$$

Then there is an $\Omega_0 \in \Lambda$ s.t.

$$\forall M \in \Lambda\ \forall i \in \mathbb{N}\ \ \forall "P = Q" \in U_i\ \ T_i + \Omega_0 = M \nvdash P = Q.$$

Proof. As in 3.6. □

3.9 COROLLARY. Let T be an RE λ-theory. Then there are $\Omega_1, \Omega_2, \ldots$ s.t. for every $M_1, M_2, \ldots$ $T + \Omega_1 = M_1 + \Omega_2 = M_2 + \ldots$ is consistent.

Proof. Apply 3.8. Take $T_i = T$, $U_i = \{"K = I"\}$. Let Ω_0 be given by the theorem. Take: $\Omega_1 = \Omega_0\ \underline{0}$, $\Omega_2 = \Omega_0\ \underline{1}, \ldots$.
It is sufficient to show that for any n $T + \Omega_0\ \underline{0} = M_0, \ldots \Omega_0\,\underline{n} = M_n$ is consistent. Let φ be recursive s.t. $\varphi(0) = \ulcorner M_0 \urcorner, \ldots \varphi(n) = \ulcorner M_n \urcorner$.

Let F represent φ in $\lambda\beta$. Then $T + \Omega_0 = \lambda x E(Fx)$ is consistent. And $T + \Omega_0 = \lambda x E(Fx) \vdash \Omega_0\,\underline{0} = E(F\underline{0}) = E\ulcorner\underline{M_0}\urcorner = M_0, \ldots \Omega_0\,\underline{n} = M_n$.

3.10 COROLLARY. Let $\mathcal{T}_\infty = (\mathbf{T}_\infty, \subseteq)$ be the partial ordering of $\lambda\beta$-theories or of theories in the language of PA, extending PA, then $P\omega$ can be embedded in $\mathcal{T}_\infty$ i.e. there is an $f : P\omega \to \mathbf{T}_\infty$ s.t.

$$\forall x,y \in P\omega\ (x \subseteq y \Longleftrightarrow f(x) \subseteq f(y)).$$

Proof. Let $x \subseteq \mathbb{N}$. In the case of $\lambda\beta$ take $f(x) = \lambda\beta + \{\Omega_0\,\underline{n} = KI \mid n \in x\}$ where Ω_0 is as in 3.9. In the case of PA take $f(x) = PA + \{A_0(\underline{n}) \mid n \in x\}$. Clearly in both cases $x \subseteq y \Rightarrow f(x) \subseteq f(y)$. By 3.9 and 3.7: $x \neq y \Rightarrow f(x) \neq f(y)$. □

3.11 INTRODUCTION TO 3.12. We are going to prove 3.12 both for RE $\lambda\beta$-theories and for RE theories in the language of PA, extending PA. To avoid unnecessary duplication we need a little dictionary:

NOTATION	MEANING FOR PA	MEANING FOR $\lambda\beta$
A,B,C	Σ_1^0-sentences	Identities between closed λ-terms
Ω_0,X,Y,Z	Σ_1^0-sentences	closed λ-terms
$\sim$	$\leftrightarrow$	$=$
Tr	$\underline{0} = \underline{0}$	KI
Fa	$\underline{0} = \underline{1}$	I
$X \vee A$	$X \vee A$	$XM = XN$ where $A \equiv (M = N)$
$X \wedge A$	$X \wedge A$	$X(KTr)M = X(KFa)N$ where $A \equiv (M = N)$
$X \to A$	$X \to A$	$X(KTr) \vee A$
$\mathbf{T}$	The set of RE theories, extending PA, in the language of PA	The set of RE $\lambda\beta$-theories

Of the special logical constants for $\lambda\beta$ we really need only Tr, Fa and $\vee$. We use the following fact:

3.11.1 FACT FOR $\lambda\beta$

(i) $\lambda\beta \vdash \mathrm{Tr} \vee A$

(ii) $\lambda\beta, \mathrm{Fa} \vee A \vdash A$

(iii) $\lambda\beta, A \vdash X \vee A$

(iv) $\lambda\beta, X = Y, X \vee A \vdash Y \vee A$

Proof. All the verifications are trivial for example in case *i)* if $A \equiv (M = N)$ then $\lambda\beta \vdash \mathrm{KIM} = \mathrm{I} = \mathrm{KIN}$. □

3.12 THEOREM. Any countable p.o. $S = (S,\leqslant)$ can be embedded in $T = (T,\subseteq)$.

Let $S = (S,\leqslant)$ be a countable p.o.

3.12.1 DEFINITION. Let $P,Q \subseteq S$; $P,Q \neq \emptyset$. We define:

$P \leqslant Q : \iff \exists p \in P,\ q \in Q\ \ p \leqslant q$.

3.12.2 DEFINITION. A function $f : S \to T$ is called <u>faithful to</u> S if: *(i)* $\forall p,q \in S \quad p \leqslant q \Rightarrow f(p) \subseteq f(q)$

(ii) $(\forall A \bigcap_{p\in P} f(p) \vdash A \Rightarrow \bigcup_{q\in Q} f(q) \vdash A) \Rightarrow P \leqslant Q$, for all $P,Q \neq \emptyset$, $P, Q \subseteq S$.

3.12.3 MINILEMMA. f is faithful $\Rightarrow$ f is an embedding.

Proof. We just have to prove that $f(p_0) \subseteq f(q_0) \Rightarrow p_0 \leqslant q_0$.

Well: $f(p_0) \subseteq f(q_0) \Rightarrow \bigcap_{p\in\{p_0\}} f(p) \subseteq \bigcup_{q\in\{q_0\}} f(q) \Rightarrow$

$$(\forall A \bigcap_{p\in\{p_0\}} f(p) \vdash A \Rightarrow \bigcup_{q\in\{q_0\}} f(q) \vdash A) \Rightarrow$$

$$\{p_0\} \leqslant \{q_0\} \Rightarrow p_0 \leqslant q_0. \quad \square$$

3.12.4 LEMMA. Let $S = (S,\leqslant)$ be finite with a distinct top $\top$ and bottom $\perp$. Suppose f is faithful to S. Let $S_0 = (S_0,\leqslant_0)$ be a p.o.

s.t. *(i)* $S_0 = S \cup \{s_0\}$, $s_0 \notin S$
(ii) $\leqslant \; = \; \leqslant_0 \restriction S$

then there is an f_0 faithful to $\mathcal{S}_0$ with $f_0 \restriction S = f$.
Let's first prove the theorem from the lemma before proving the lemma.

Proof of theorem 3.12. Let $\mathcal{S} = (S,\leqslant)$ be countable. Add to $\mathcal{S}$ a top and a bottom, $\top$ and $\perp$. ($\top,\perp \notin S$). We get $\mathcal{S}' = (S \cup \{\top,\perp\},\leqslant')$ where for any $u,v \in S \cup \{\top,\perp\}$:

$$u \leqslant' v \Leftrightarrow ((u,v \in S \text{ and } u \leqslant v) \text{ or } u = \perp \text{ or } v = \top).$$

Suppose $S = \{s_1,s_2,\ldots\}$ $(i \neq j \Rightarrow s_i \neq s_j)$. Define:

$$S_n := \{\top,\perp,s_1,\ldots s_n\} \quad (n = 0,1,\ldots),$$
$$\mathcal{S}_n := (S_n, \leqslant' \restriction S_n).$$

Let T_0,T_1 be any two elements of $\mathcal{T}$ s.t. $T_0 \subseteq T_1$ and there is an A s.t. $T_0 \nvdash A$ and $T_1 \vdash A$. (Note that in the case of PA $A \in \Sigma_1^0$ so what we ask is stronger than $T_0 \subsetneq T_1$). For example for $\lambda\beta$ take $T_0 = \lambda\beta$, $T_1 = \lambda\beta + \Omega = I$ and for PA take $T_0 = PA$ and $T_1 = PA + \neg con(PA)$.
Take: $f_0 : S_0 \to \mathcal{T}$ as $f_0(\perp) = T_0$, $f_0(\top) = T_1$. Clearly f_0 is faithful to $\mathcal{S}_0$. By the lemma we find $f_0 \subseteq f_1 \subseteq \ldots$, where f_i is faithful to $\mathcal{S}_i$ and thus an embedding. Define $g := \bigcup_{i=0}^{\infty} f_i$ and $f = g \restriction S$. It is easy to see that f is an embedding of $\mathcal{S}$ in $\mathcal{T}$. □

3.12.5 REMARK.

(i) We could also do our proof for the case of PA with Π_1^0-sentences instead of Σ_1^0-sentences. This would give the extra result that we could embed any countable p.o. in the <u>true</u> RE theories extending PA in the language of PA.

(ii) We could also do our proof for the case of PA with $\Sigma_n^0(\vec{x})$ or $\Pi_n^0(\vec{x})$ formulae. From this would follow that for any T_0,T_1 with $PA \subseteq T_0 \subset T_1$ (T_0,T_1 RE and in the language of PA) we could

embed any countable p.o. in the RE-theories in the language of PA between T_0 and T_1.
Because if $T_0 \subset T_1$ then there must be an $n, \vec{x}$ s.t. there is a $\Sigma^0_n(\vec{x})$ or a $\Pi^0_n(\vec{x})$ formula F s.t. $T_0 \nvdash F$ and $T_1 \vdash F$.

Now let's do the proof of the Lemma.

Proof of Lemma 3.12.4.

Define: $\hat{P}_0 := \{p \in S \mid p <_0 s_0\}$

$\check{P}_0 := \{p \in S \mid s_0 <_0 p\}$.

We have $\bot \in \hat{P}_0, \top \in \check{P}_0$ so $\hat{P}_0, \check{P}_0 \neq \emptyset$.
Let $\hat{P}_0, \hat{P}_1, \ldots \hat{P}_N$ be the subsets of S s.t. $\hat{P}_0 \subseteq \hat{P}_i$ and $\top \notin \hat{P}_i$ $(i = 0,\ldots N)$ (clearly $\top \notin \hat{P}_0$).

Let $\check{P}_0, \ldots \check{P}_{N'}$ be the subsets of S s.t. $\check{P}_0 \subseteq \check{P}_j$ and $\bot \notin \check{P}_j$ $(j = 0,\ldots N')$ (clearly $\bot \notin \check{P}_0$).
Define for $i = 0,\ldots N$:

$$\check{Q}_i := \{q \in S \mid \forall p \in \hat{P}_i \; q \not\leq p\},$$

then: $\top \in \check{Q}_i$ so $\check{Q}_i \neq \emptyset$;

$\check{Q}_i \not\leq \hat{P}_i$;

$Q \not\leq \hat{P}_i \Rightarrow Q \subseteq \check{Q}_i$.

Define for $j = 0,\ldots N'$:

$$\hat{Q}_j := \{q \in S \mid \forall p \in \check{P}_j \; p \not\leq q\},$$

then: $\bot \in \hat{Q}_j$ so $\hat{Q}_j \neq \emptyset$;

$\check{P}_j \not\leq \hat{Q}_j$;

$\check{P}_j \not\leq Q \Rightarrow Q \subseteq \hat{Q}_j$.

Let $f(s) = T_s$ for $s \in S$. We will define T_{s_0}. Make $f_0(s) = T_s$ for $s \in S_0$. Because f is faithful to S we have, for $i = 0,\ldots,N$:

$$\check{Q}_i \not\leq \hat{P}_i,$$

thus there is an A_i s.t. $\bigcap_{q\in\check{Q}_i} T_q \vdash A_i$ and $\bigcup_{p\in\hat{P}_i} T_p \nvdash A_i$.

and for $j = 0,\ldots,N'$:

$$\check{P}_j \not\leqslant \hat{Q}_j$$

and thus there is a B_j s.t. $\bigcap_{p\in\check{P}_j} T_p \vdash B_j$ and $\bigcup_{q\in\hat{Q}_j} T_q \nvdash B_j$.

By theorems 3.8 and 3.6 pick Ω_0 s.t. for all $0 \leqslant i \leqslant N$, $0 \leqslant j \leqslant N'$ and for all X

$$\bigcup_{p\in\hat{P}_i} T_p + \Omega_0 \sim X \nvdash A_i$$

$$\bigcup_{q\in\hat{Q}_j} T_q + \Omega_0 \sim X \nvdash B_j$$

Define: $T_{s_0} := \bigcup_{p\in\hat{P}_0} T_p + \Omega_0 \vee B_1 + \ldots \Omega_0 \vee B_{N'}$.

We have to check that f_0 is faithful.

Ad(i) Suppose $u,v \in S_0$ and $u \leqslant_0 v$.

case a) $u,v \in S$, then $T_u \subseteq T_v$.

case b) $u = v = s_0$, then $T_u \subseteq T_v$.

case c) $u \neq s_0$, $v = s_0$. Then $u <_0 s_0$ i.e. $u \in \hat{P}_0$.

So $T_u \subseteq \bigcup_{p\in\hat{P}_0} T_p$. Thus $T_u \subseteq T_{s_0} = T_v$.

case d) $u = s_0$, $v \neq s_0$. Then $s_0 <_0 v$. We have $v \in \check{P}_0$ and thus for any $0 \leqslant j \leqslant N'$ $v \in \check{P}_j$,

so $T_v \supseteq \bigcap_{p\in\check{P}_j} T_p \vdash B_j$. Therefore $T_v \vdash \Omega_0 \vee B_j$.

Moreover because f is faithful and thus an embedding we have:

$$T_v \supseteq \bigcup_{p\in\hat{P}_0} T_p \text{ (for } p \in \hat{P}_0 \Rightarrow p <_0 s_0 <_0 v \Rightarrow p <_0 v \Rightarrow p < v)$$

So

$$T_v \supseteq \bigcup_{p\in\hat{P}_0} T_p + \Omega_0 \vee B_0 + \ldots \Omega_0 \vee B_{N'} = T_{s_0} = T_u.$$

Ad(ii) Suppose $U \not\leq_0 V$ $(u,v \neq \emptyset)$; we prove that there is a C s.t. $\bigcap_{u\in U} T_u \vdash C$ and $\bigcup_{v\in V} T_v \nvdash C$.

We distinguish three cases:

case a) $s_0 \notin U$, $s_0 \notin V$. Then we are ready by the faithfulness of f.

case b) $s_0 \in U$, then of course $s_0 \notin V$.
Take $\widetilde{U} := (U\setminus\{s_0\}) \cup \check{P}_0$. Then $\widetilde{U}$ is one of the $\check{P}_j$, for trivially $\perp \notin \widetilde{U}$.
Suppose $\widetilde{U} = \check{P}_{j_0}$. We find $U \not\leq V$ so $V \subseteq \hat{Q}_{j_0}$. Clearly for every $u \in \widetilde{U}$: $T_u \vdash B_{j_0}$ so certainly $T_u \vdash \Omega_0 \vee B_{j_0}$.
Moreover $T_{s_0} \vdash \Omega_0 \vee B_{j_0}$. So

$$\bigcap_{u\in U} T_u \supseteq \bigcap_{u\in \widetilde{U}\cup\{s_0\}} T_u \vdash \Omega_0 \vee B_{j_0}.$$

Suppose on the other hand that $\bigcup_{v\in V} T_v \vdash \Omega_0 \vee B_{j_0}$, then certainly $\bigcup_{q\in \hat{Q}_{j_0}} T_q \vdash \Omega_0 \vee B_{j_0}$. But then

$$\bigcup_{q\in \hat{Q}_{j_0}} T_q + \Omega_0 \sim Fa \vdash B_{j_0}.$$

This contradicts our choice of Ω_0. So we can take $C := \Omega_0 \vee B_{j_0}$.

case c) $s_0 \in V$, then $s_0 \notin U$.
Take $\widetilde{V} := (V\setminus\{s_0\}) \cup \hat{P}_0$. Then we have: $\widetilde{V}$ is one of the $\hat{P}_i$, for trivially $\top \notin \widetilde{V}$.
Suppose $\widetilde{V} = \hat{P}_{i_0}$. We find $U \not\leq \widetilde{V}$ so $U \subseteq \check{Q}_{i_0}$. Then

$$\bigcap_{u\in U} T_u \supseteq \bigcap_{q\in \check{Q}_{i_0}} T_q \vdash A_{i_0}.$$

Suppose $\bigcup_{v\in V} T_v \vdash A_{i_0}$.
Well: $\bigcup_{v\in V} T_v = \bigcup_{v\in \widetilde{V}} T_v + \Omega_0 \vee B_0 + \ldots \Omega_0 \vee B_{N'}$. It follows that $\bigcup_{v\in \widetilde{V}} T_v + \Omega_0 \sim Tr \vdash A_{i_0}$. But $\widetilde{V} = \hat{P}_{i_0}$, so this contradicts our choice of Ω_0. So take $C := A_{i_0}$. □

3.13 REMARK. Inspection of the proof shows that the theories constructed are all finite over T_0. So using conjunction in case of PA or pairing in case of $\lambda\beta$, we can formulate the theorem for sentences instead of for theories.

3.14 REMARK. The version of the Gödel-Rosser-Mostowski-Myhill-Kripke-Theorem which is closest to mine is that of Kripke in Kripke (1963). Of course we have been a bit too generous using PA: a subsystem like Robinson's Arithmetic would have been sufficient.

§4 THE ANTI DIAGONAL NORMALISATION THEOREM AND SOME OF ITS APPLICATIONS.

4.1 MEDITATION. Consider $M_{\lambda\beta}$. By identifying certain elements, the unsolvables, and closing off under the rules of $\lambda\beta$-calculus we get M_H. M_H is complete. Here we have a natural transformation of a precomplete numeration into a complete one. The elements that we identify first are intuitively the "undefined" elements. By closing off under the rules certain other elements are also identified.
Now if we want to generalize the transition form $\lambda\beta$ to H it would be nice to have some control over the set:

$$A^\gamma := \{\{z_0\}^\gamma z_1 \mid \{z_0\}z_1\uparrow,\ z \in \mathbb{N}\},$$

for arbitrary precomplete γ. The Anti Diagonal Normalisation Theorem seems a step in the right direction; it is not strong enough to get the consistency of the identification of the elements of A^γ in the general case.

4.2 DEFINITION. Suppose $R(\vec{x})$ and $Q(\vec{x})$ are RE-relations. We write them in the form: $\exists y R_0(y,\vec{x})$ and $\exists y Q_0(y,\vec{x})$, where R_0 and Q_0 are recursive. Define:

$$R(\vec{x}) \leqslant Q(\vec{x}) \ :\Leftrightarrow\ \exists y\ \ R_0(y,\vec{x}) \text{ and } \forall z < y \text{ not } Q_0(z,\vec{x}).$$
$$R(\vec{x}) < Q(\vec{x}) \ :\Leftrightarrow\ \exists y\ \ R_0(y,\vec{x}) \text{ and } \forall z \leqslant y \text{ not } Q_0(z,\vec{x}).$$

Then $R(\vec{x}) \leqslant Q(\vec{x})$ and $R(\vec{x}) < Q(\vec{x})$ are RE relations.

4.3 DEFINITION. Let γ be a numeration. A partial recursive Δ is <u>a diagonal function</u> for γ if:

$$\forall x \in \mathrm{Dom}(\Delta) \quad \Delta(x) \not\sim_\gamma x.$$

4.4 THE ANTI DIAGONAL NORMALISATION THEOREM. Given a precomplete numeration γ and a partial recursive diagonal function Δ for γ, there is a total recursive function $\{x\}^{\gamma;\Delta}y$ s.t.

1) $\{x\}y\downarrow \Rightarrow \{x\}y \sim_\gamma \{x\}^{\gamma;\Delta}y$

2) $\{x\}y\uparrow \Rightarrow \{x\}^{\gamma;\Delta}y \notin \mathrm{Dom}(\Delta)$.

Proof. Let δ be an index for Δ. Define:

$$\{d(\delta,e,x)\}y :\cong \begin{cases} \{e\}x \ \mathit{if}\ \{e\}x\downarrow \leqslant \{\delta\}(\{y\}^\gamma y)\downarrow & (*) \\ \{\delta\}(\{y\}^\gamma y) \ \mathit{if}\ \{\delta\}(\{y\}^\gamma y)\downarrow < \{e\}x\downarrow & (**) \end{cases}$$

Let us write $d := d(\delta,e,x)$. Suppose: $\{d\}d\downarrow$ because of clause $(**)$, then $\{d\}d \cong \{\delta\}(\{d\}^\gamma d) \not\sim_\gamma \{d\}^\gamma d \sim_\gamma \{d\}d$. Contradiction. So if $\{d\}d\downarrow$ it does so because of $(*)$. Thus we find :
$\{d\}d\downarrow \Rightarrow \{d\}d \cong \{e\}x$.
Suppose $\{d\}d\uparrow$ then we must have $\{\delta\}(\{d\}^\gamma d)\uparrow$, else $(**)$ would give $\{d\}d$ a value. So $\{d\}d\uparrow \Rightarrow \{d\}^\gamma d \notin \mathrm{Dom}(\Delta)$.
So take: $\{x\}^{\gamma;\Delta}y :\cong \{d(\delta,x,y)\}^\gamma d(\delta,x,y)$.

4.5 REMARK. The idea for the Anti Diagonal Theorem occurred to me when I tried to find another proof of the theorem of Smoryński (1978), which is a consequence of the ADNT. Smoryński's forerunners are Shepherdson's fixed point and ultimately the Rosser Sentence.

4.6 COROLLARY. Let γ be precomplete. Let U be RE, non trivial and closed under $\sim_\gamma$, then U is maximal in the RE m-degrees and hence creative. (This corollary can be proved in other ways, see Eršov (1973)).

Proof. Consider an RE set V. Suppose $n_0 \in U$, $n_1 \notin U$. Take:

$$\Delta x :\cong \begin{cases} n_1 & \text{if } x \in U \\ \uparrow & \text{else} \end{cases}$$

and

$$\varphi(x) :\cong \begin{cases} n_0 & \text{if } x \in V \\ \uparrow & \text{else} \end{cases}$$

Clearly Δ is a diagonal function for γ.
Suppose p is an index of φ, then:

$$x \in V \Rightarrow \{p\}x \cong n_0 \Rightarrow \{p\}^{\gamma;\Delta}x \sim_\gamma n_0 \Rightarrow \{p\}^{\gamma;\Delta}x \in U$$

$$x \notin V \Rightarrow \{p\}x \uparrow \Rightarrow \{p\}^{\gamma;\Delta}x \notin \mathrm{Dom}(\Delta) \Rightarrow \{p\}^{\gamma;\Delta}x \notin U. \qquad \square$$

4.7 COROLLARY OF 4.6.

(i) Every non trivial RE set of λ-terms closed under β-convertibility is creative.

(ii) Any RE set U of identities between λ-terms s.t. if $(M = N) \in U$ and $\lambda\beta \vdash M = M'$, $\lambda\beta \vdash N = N'$ then $M' = N' \in U$ is creative or trivial.

(iii) Any RE $\lambda\beta$-theory is creative.

(iv) Let T be any RE $\lambda\beta$-theory s.t. $T \nvdash M = N$ then: $\{(P = Q) \mid T + P = Q \vdash M = N\}$ is creative.

Proof.

(i) routine from 4.6.

(i) $\Rightarrow$ *(ii)* $\Rightarrow$ *(iii)*, *(iv)* routine. $\square$

4.8 THEOREM. Let $(T_i)_{i\in\mathbb{N}}$ be a recursive sequence of $\lambda\beta$-theories. Let $(U_i)_{i\in\mathbb{N}}$ be a recursive sequence of sets of (codes of) identities between λ-terms s.t.

$$\forall i\ \forall \text{"}M = N\text{"} \in U_i\ \ T_i \nvdash M = N.$$

Let

$$I := \{\Omega_0 \in \Lambda \mid \ \forall P \in \Lambda\ \forall i \in \mathbb{N}\ \forall \text{"}M = N\text{"} \in U_i T_i + \Omega_0 = P \nvdash M = N\}.$$

Then there is an $F \in \Lambda$ s.t. for all $i,p,m,n \in \mathbb{N}$:

$$\{p\}m \cong n \iff \lambda\beta \vdash F\,\underline{p}\,\underline{m} = \underline{n} \iff T_i \vdash F\,\underline{p}\,\underline{m} = \underline{n}.$$

$$\{p\}m\uparrow \iff \exists\,\Omega_0 \in I\ \ \lambda\beta \vdash F\,\underline{p}\,\underline{m} = \Omega_0.$$

Proof. Take:

$$\Delta\ulcorner P\urcorner :\cong \varepsilon\ulcorner Q\urcorner(\exists\, i \in \mathbb{N}\ \exists\ \text{"}M = N\text{"} \in U_i\ \ T_i + P = Q \vdash M = N.$$

Clearly Δ is a diagonal function for $M_{\lambda\beta}$ and $\ulcorner P\urcorner \notin \operatorname{Dom}\Delta \iff P \in I$.
Choose p' s.t.

$$\{p'\}m \cong \begin{cases} \ulcorner \underline{n}\urcorner & \text{if } \{p\}m \cong n \\ \uparrow & \text{else} \end{cases}$$

Let p" be an index of $\{p'\}^{M_{\lambda\beta};\Delta}$.
It is clear that we can find p" from p in an effective way. So there is a λ-term Q which represents: $\psi(p,m) :\cong \{p''\}m$ in $\lambda\beta$.
Take $F :\equiv (\lambda xy\,.\,E(Qxy))$. (Remember that E is the universal constructor i.e. $\forall M \in \Lambda\ \ \lambda\beta \vdash E\ulcorner \underline{M}\urcorner = M$).
We have:

(i) $\{p\}m \cong n \ \Rightarrow\ \{p'\}m \cong \ulcorner \underline{n}\urcorner \ \Rightarrow\ (\{p''\}m \cong \ulcorner M\urcorner$
where $\ulcorner M\urcorner \sim_{M_{\lambda\beta}} \ulcorner \underline{n}\urcorner) \ \Rightarrow\ \lambda\beta \vdash E(Q\,\underline{p}\,\underline{m}) = E(\ulcorner \underline{M}\urcorner) = M = \underline{n}$
$\Rightarrow\ T_i \vdash E(Q\,\underline{p}\,\underline{m}) = \underline{n}.$

(ii) $\{p\}m\uparrow \ \Rightarrow\ \{p'\}m\uparrow \ \Rightarrow\ \{p''\}m = \ulcorner\Omega_0\urcorner$
$\Rightarrow\ \lambda\beta \vdash E(Q\,\underline{p}\,\underline{m}) = E\ulcorner\Omega_0\urcorner = \Omega_0$
$\Rightarrow\ T_i \vdash E(Q\,\underline{p}\,\underline{m}) = \Omega_0$. Where $\Omega_0 \in I$.

Moreover by the properties of elements of I and the fact that numerals cannot be identified consistently with every term it follows $\forall \underline{n}\ \ T_i \nvdash \Omega_0 = \underline{n}$ for $\Omega_0 \in I$.
A moment's reflection shows that the theorem follows.

4.9 REMARK. Obviously a similar theorem can be given for PA.

4.10 COROLLARY. The partial recursive functions can be represented in any RE λ-theory.

4.11 EXAMPLE OF THE CONSTRUCTION OF A COMPLETE NUMERATION BY MEANS OF $M_{\lambda\beta}$.

Take I as in 4.8 for the constant sequence with $T_i = \lambda\beta$ and $U_i = \{\text{"K = I"}\}$. Define:

$$H^- := \lambda\beta + \{(M = N) \mid M,N \in I\}.$$

It is easy to see that if $M \in I$ then M is unsolvable, so H^- is consistent. By 4.8 M_{H^-} is complete.

4.12 REMARKS.

(i) We could do a similar construction for e.g. P^{PA}.

The point where we use "M $\in$ I then M is unsolvable" contains a reference to specific properties of $\lambda\beta$-calculus. It would be nice to eliminate this. As yet I see no way to avoid it. Possibly some natural condition is missing. In the case P^{PA} one uses that the identifications are true.

4.13 REMARK. There are some analogies between λ-calculus and PA that invite further reflection. I will state them without proof. Let for $e,f \in \mathbb{N}$, "$e \sim f$" mean "$\forall x \{e\}x \cong \{f\}x$". Define:

$$T_0 := PA + \{(\underline{e} \sim \underline{f}) \mid \forall n \{e\}n\uparrow,\ \{f\}n\uparrow\};$$

it is easy to see that

$$T_0 = PA + \Sigma_2^0\text{-truth}.$$

Let $T_1 := PA + \{(\underline{e} \sim \underline{f}) \mid e \sim f\}$. $P^{T_1} = P$ up to isomorphism.

We have:

1) $\{(M = N) \mid \lambda\beta \vdash M = N\}$ is Σ_1^0-complete

$\{(e \sim f) \mid PA \vdash \underline{e} \sim \underline{f}\}$ is Σ_1^0-complete.

2) H is Σ_2^0-complete (see Barendregt (1978));

$\{(e \sim f) \mid T_0 - \underline{e} \sim \underline{f}\}$ is Σ^0_2. (Σ^0_2 complete?)

3) $\mathcal{H}^*$ is Π^0_2-complete and the unique maximal extension of $\mathcal{H}$;

$\{(e \sim f) \mid e \sim f\} = \{(e \sim f) \mid T_1 \vdash \underline{e} \sim \underline{f}\}$ is Π^0_2-complete and is the unique maximal (consistent) extension of $\{(e \sim f) \mid T_0 \vdash \underline{e} \sim \underline{f}\}$ in the sense that $T_0 + \underline{e} \sim \underline{f} \nvdash \underline{0} = \underline{1} \Rightarrow T_1 \vdash \underline{e} \sim \underline{f}$.

4.16 REMARK. For a proof that $M_{\mathcal{H}}$ is complete see Barendregt (1975). The results on many-one degrees of λ-theories mentioned in 4.15 are from Barendregt (1978).

REFERENCES.

Barendregt, H.P., Some extensional term models for combinatory logics and λ-calculi, thesis, Utrecht, 1971.

Barendregt, H.P., Solvability in lambda calculi, Colloque International de Logique, 1975, Editions du Centre National de la Recherche Scientifique, pp. 209-219.

Barendregt, H.P., The Type Free Lambda Calculus, Handbook of Mathematical Logic, 1977, North-Holland, pp. 1091-1132.

Barendregt, H.P., Bergstra J., Klop J.W., Volken, H., Degrees of sensible lambda theories, JSL, volume 43, no. 1, 1978, pp. 45-55.

Boolos, G., The Unprovability of Consistency, 1979, Cambridge University Press.

Eršov, Ju.L., Theorie der Numerierungen I, Zeitschrift für Math. Log., Band 19, Heft 4, 1973, pp. 289-388.

Eršov, Ju.L., Theorie der Numerierungen II, Zeitschrift für Math. Log., Band 21, Heft 6, 1975, pp. 473-584.

Eršov, Ju.L., Model $\mathbb{C}$ of Partial Continuous Functionals, Logic Colloquium 1976, North-Holland, 1977, 455-467.

Kripke, S., "Flexible" predicates in formal number theory, Proc. AMS 13, 1963, pp. 647-650.

Rogers, Theory of Recursive Functions and Effective Computability, 1967, McGraw-Hill.

Scott, D., Data types as lattices, Logic Conference, Kiel 1974, Lecture Notes in Mathematics 499, Springer Verlag, 1975, pp. 579-651.

Scott, D., Combinators and classes, λ-Calculus and Computer Science Theory, 1975, Lecture Notes in Computer Science 37, Springer Verlag, pp. 1-26.

Scott D., Some philosophical issues concerning theories of of combinators, λ-calculus & Computer Science Theory, 1975, Lecture Notes in Computer Science 37, Springer Verlag, pp. 346-366.

Smoryński, C., Avoiding Selfreferential Statements, Proceedings of the AMS, volume 70, number 2, July 1978, pp. 181-184.

SEMANTICS

COMPARING SOME CLASSES OF LAMBDA CALCULUS MODELS.

Henk Barendregt and Karst Koymans

Mathematical Institute, Budapestlaan 6,
Utrecht, The Netherlands.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

ABSTRACT

This paper discusses various definitions of models for combinatory logic and the λ-calculus, viz. combinatory algebras, λ-algebras, λ-models and extensional λ-models. One has

combinatory algebras $\supseteq$ λ-algebras
$\supseteq$ λ-models
$\supseteq$ extensional λ-models.

The following will be proved.

1. Not every combinatory algebra can be made into a λ-algebra.

2. Not every λ-algebra can be made into a λ-model.

3. Every λ-algebra can be embedded into a λ-model.

4. Not every λ-model can be embedded into or mapped onto an extensional one.

§0 INTRODUCTION

A <u>combinatory algebra</u> (model of combinatory logic) is a structure $\mathfrak{M} = (|\mathfrak{M}|, \cdot)$, with $\mathrm{Card}(|\mathfrak{M}|) > 1$ and a binary operation $\cdot$, that is <u>combinatory complete</u>, i.e. for any applicative term $A(x_1,\ldots,x_n)$ one has

$$(*)\quad \mathfrak{M} \models \exists a \forall x_1 \ldots x_n \quad a x_1 \ldots x_n = A(x_1,\ldots,x_n).$$

For extensional $\mathfrak{M}$, i.e.

$$\mathfrak{M} \models \forall x(ax = a'x) \rightarrow a = a',$$

the a in (*) is uniquely determined and one can define $\lambda\vec{x}.A(\vec{x})=a$.

If $\mathfrak{M}$ is not extensional, then it still may be possible to define $\lambda\vec{x}.A(\vec{x})$ depending only on the extensional behaviour of $A(\vec{x})$ as a function of $\vec{x}$. Such structures $\mathfrak{M}$ will be called λ-models (weakly extensional λ-algebras in Barendregt (1977)). These structures satisfy

$$(\xi) \qquad \mathfrak{M} \models \forall x \ A(x) = B(x) \rightarrow \lambda x.A(x) = \lambda x.B(x)$$

In Plotkin (1974) it is shown that the set of closed λ-terms (modulo β-convertibility) does not satisfy (ξ). Still such a term model is good enough to interpret λ-terms in an adequate way. Therefore these structures (not satisfying (ξ)) are called λ-algebras.

The 'mathematical' λ-calculus models[1] like D_∞, P_ω and T^ω [2] are all λ-models (D_∞ is even extensional). These structures give a nice modeltheory for λ-calculus. Nevertheless closed term models are interesting enough to make us study also λ-algebras, see e.g. Visser (1979). Further investigations have to determine what are the right concepts in λ-calculus model theory.

§1 THE MODELS

For undefined notations we refer to Barendregt (1977). The definitions of various λ-calculus models will be repeated here.

1.1 *Definition.* An <u>applicative structure</u> $\mathfrak{M} = (|\mathfrak{M}|, \cdot)$ consists of a non empty domain $|\mathfrak{M}|$ and a binary application $\cdot : |\mathfrak{M}|^2 \rightarrow |\mathfrak{M}|$.

1.2 *Definition.*

(i) $T(\mathfrak{M})$ is the set of <u>applicative terms</u> using constants for elements of $\mathfrak{M}$:

$$a \in |\mathfrak{M}| \Rightarrow c_a \in T(\mathfrak{M})$$

---

1) The hypergraph model, see Sanchis (1979), is a combinatory algebra. It is not known yet whether it is also a λ-algebra or λ-model.

2) See e.g. Barendregt and Longo: Equality of lambda terms in the model T^ω, this volume.

$x \in T(\mathfrak{M})$ for $x \in$ Vars (the set of variables),
$A,B \in T(\mathfrak{M}) \Rightarrow (AB) \in T(\mathfrak{M})$.

(ii) If $A \in T(\mathfrak{M})$ and $\rho : \text{Vars} \to |\mathfrak{M}|$, then $(\!| A |\!)_\rho$ is the usual first order interpretation of A in $\mathfrak{M}$ under the valuation ρ.

Now one can use the common modeltheoretic notation such as $\mathfrak{M} \models A = B$ (i.e. $(\!| A |\!)_\rho = (\!| B |\!)_\rho$ for all ρ), $\mathfrak{M} \models \forall a \exists x\, ax = x$.

1.3 *Definition.* Let $\mathfrak{M}$ be an applicative structure.

(i) Combinatory completeness is the following property: for all $\vec{x} = x_1,\dots,x_n$ and $A(\vec{x}) \in T(\mathfrak{M})$ with $FV(A) \subseteq \{\vec{x}\}$ one has
$\mathfrak{M} \models \exists a\ \forall \vec{x}\ \ a\vec{x} = A(\vec{x})$.

(ii) $\mathfrak{M}$ is a combinatory algebra iff $\mathfrak{M}$ satisfies combinatory completeness and $\text{Card}(|\mathfrak{M}|) > 1$.

1.4 FACT. $\mathfrak{M}$ is a combinatory algebra iff there exist $k,s \in |\mathfrak{M}|$ such that $(\mathfrak{M},k,s) \models kxy = x \wedge sxyz = xz(yz) \wedge k \neq s$.

In order to interpret also λ-terms in a combinatory algebra $\mathfrak{M}$, one postulates the following operator λ^*. Usually one can find such a λ^* by the proof that $\mathfrak{M}$ is a combinatory algebra.

1.5 *Definition.* $\mathfrak{M} = (|\mathfrak{M}|, \cdot, \lambda^*)$ is a pre-λ-algebra iff $\mathfrak{M}$ is an applicative structure and $\lambda^* : \text{Vars} \times T(\mathfrak{M}) \to T(\mathfrak{M})$ is such that *(1)* $FV(\lambda^*x.A) \subseteq FV(A) - \{x\}$

(2) $\mathfrak{M} \models (\lambda^*x.A)x = A$

(3) $\lambda^*x.A \equiv \lambda^*y.A[x := y]$ for $y \notin FV(A)$

(4) $\lambda^*x.A[y := z] \equiv (\lambda^*x.A)[y := z]$ for $y,z \not\equiv x$ and $z \notin FV(A)$.

($[y := B]$ denotes substitution of B for y).

REMARK.

(i) (3) and (4) will imply that α-convertible terms are interpreted identically.

(ii) Although (3) and (4) are not strictly needed in full strength to get α-convertible terms interpreted equal, they are very natural.

Moreover note that if some λ^* satisfies (1) and (2) in 1.5, it can be modified so as to satisfy (3) and (4) as well. Indeed, define λ_1^* in terms of λ^* as follows:

$$\lambda_1^* x.A(x,y_1,\ldots,y_n) \underset{D}{\equiv} (\lambda^* x_0.A(x_0,x_1,\ldots,x_n))[x_1,\ldots,x_n := y_1,\ldots,y_n],$$

where $y_1,\ldots,y_n$ are the (distinct) free variables of A other than x in order of first occurrence in A. It is easily checked that (3) and (4) are satisfied for this λ_1^* and that (1), (2) remain valid.

1.6 *Definition.* Let $\mathfrak{M}$ be an applicative structure.

(i) Let X be a set of constants. Then Λ_X (respectively Λ_X^0) is the set of (closed) λ-terms built up from variables and constants in X.

(ii) $\Lambda_{\mathfrak{M}}^{(0)} = \Lambda_X^{(0)}$ with $X = \{c_a \mid a \in |\mathfrak{M}|\}$

(iii) λ_X is the set of equations between elements of Λ_X^0 provable by the usual axioms and rules of the λ-calculus (allowing constants).

1.7 *Definition.* Let $\mathfrak{M}$ be a pre-λ-algebra and $M \in \Lambda_{\mathfrak{M}}$.

(i) $M^* \in T(\mathfrak{M})$ is obtained from M by replacing each symbol λ by λ^*.

(ii) The <u>interpretation</u> of M in $\mathfrak{M}$ under the valuation ρ is defined by $[\![M]\!]_\rho = (\!| M^* |\!)_\rho$.

Now one can use common modeltheoretic notations for pre-λ-algebras.

1.8 *Definition.* Let $\mathfrak{M}$ be a pre-λ-algebra with $\mathrm{Card}(|\mathfrak{M}|) > 1$. Then $\mathfrak{M}$ is a <u>λ-algebra</u> iff $\mathfrak{M} \models \lambda_{\mathfrak{M}}$.

1.9 *Definition.* Let $\mathfrak{M}$ be a pre-λ-algebra.

(i) $\mathfrak{M}$ satisfies ξ_0 iff for all $A,B \in T(\mathfrak{M})$

$\mathfrak{M} \models A = B \Rightarrow \mathfrak{M} \models \lambda^*x.A \equiv \lambda^*x.B$

(ii) $\mathfrak{M}$ satisfies ξ_1 iff for all $A,B \in T(\mathfrak{M})$

$\mathfrak{M} \models \forall x\ A = B \rightarrow \lambda^*x.A = \lambda^*x.B$.

1.10 LEMMA. Let $\mathfrak{M}$ be a λ-algebra. Then in $\mathfrak{M}$

$(\lambda^*x.A(x,y))[y := c_a] = \lambda^*x.A(x,c_a)$

Proof. $(\lambda^*yx.A(x,y))c_a = \lambda^*x.A(x,c_a)$, since $\mathfrak{M} \models \lambda_{\mathfrak{M}}$;
$(\lambda^*yx.A(x,y))c_a = (\lambda^*x.A(x,y))[y := c_a]$. □

Because A,B may contain arbitrary constants from $\mathfrak{M}$, we have the following.

1.11 PROPOSITION

(i) Let $\mathfrak{M}$ be a λ-algebra. Then

$\mathfrak{M}$ satisfies $\xi_0 \Leftrightarrow \mathfrak{M}$ satisfies ξ_1.

(ii) A pre-λ-algebra satisfying ξ_1 is a λ-algebra.

Proof.

(i) Note that $(\!| A(\vec{x}) |\!)_\rho$ can be written as $(\!| A(\vec{c}) |\!)$ with $c_i = \rho(x_i)$. Now use 1.10.

(ii) First show by induction on the length of $M \in \Lambda_{\mathfrak{M}}$ that

$\mathfrak{M} \models (M[x := N])^* = M^*[x := N^*]$.

Then by induction on the length of proof

$\lambda_{\mathfrak{M}} \vdash M = N \Rightarrow \mathfrak{M} \models M = N$. □

1.12 *Definition.* A <u>λ-model</u> $\mathfrak{M}$ is a pre-λ-algebra satisfying ξ_1.

1.13 *Definition.* Let $\mathfrak{M}_i = (|\mathfrak{M}_i|, \cdot, \lambda_i^*)$ with $i = 1,2$ be two λ-algebras. Then $\varphi : |\mathfrak{M}_1| \rightarrow |\mathfrak{M}_2|$ is a homomorphism iff

(1) $\varphi(ab) = \varphi(a)\varphi(b)$ for all $a,b \in |\mathfrak{M}_1|$.

(2) For all $A \in T(\mathfrak{M}_1)$ one has

$$\mathfrak{M}_2 \models \overline{\varphi}(\lambda_1^*x.A) = \lambda_2^*x.\overline{\varphi}(A),$$

where $\overline{\varphi} : T(\mathfrak{M}_1) \rightarrow T(\mathfrak{M}_2)$ is defined by

$$\overline{\varphi}(c_a) = c_{\varphi(a)}$$

$\overline{\varphi}(x) = x$

$\overline{\varphi}(AB) = \overline{\varphi}(A)\overline{\varphi}(B)$.

1.14 *Definition* (Term models). Let X be some set of constants (possibly empty) and T be a theory extending λ_X

(i) $M \sim_T N$ iff $T \vdash M = N$

$[M]_T = \{N \in \Lambda_X \mid M \sim_T N\}$.

(ii) The (open) term model of T, notation $\mathfrak{M}(T)$, is defined by $|\mathfrak{M}(T)| = \{[M]_T \mid M \in \Lambda_X\}$

$[M]_T \cdot [N]_T = [MN]_T$

$\lambda^* x.A(x,\vec{y},\vec{c}_{[M]_T}) = c_{[\lambda\vec{y}x.A(x,\vec{y},\vec{M})]_T}\vec{y}$.

(iii) The closed term model of T, notation $\mathfrak{M}^0(T)$, is defined by relativizing (i) and (ii) to Λ^0_X.

1.15 FACT. Let T be a theory extending λ_X. Then

(i) $\mathfrak{M}(T)$ is a λ-model.

(ii) $\mathfrak{M}^0(T)$ is a λ-algebra.

§2 A COMBINATORY ALGEBRA THAT CANNOT BE MADE INTO A λ-ALGEBRA

Consider $\mathfrak{M} \equiv \mathfrak{M}(CL)$, the termmodel of combinatory logic. Due to the existence of $S,K \in C$, the terms of CL, $\mathfrak{M}$ is combinatory complete and hence is a combinatory algebra.

2.1 THEOREM. $\mathfrak{M}$ cannot be turned into a λ-algebra. I.e. no operator λ^* exists such that $(\mathfrak{M},\lambda^*)$ is a λ-algebra.
The proof occupies 2.2 - 2.7.

2.2 *Definition.* The theory $\underline{CL}$ is determined by

(i) $\text{Alphabet}_{\underline{CL}} = \text{Alphabet}_{CL} \cup \{\text{—},\text{═}\}$

(ii) Inductive definition of $\underline{C}$, the set of $\underline{CL}$-terms

a) $M \in C \Rightarrow M, \underline{M}, \underline{\underline{M}} \in \underline{C}$

b) $A,B \in \underline{C} \Rightarrow (AB) \in \underline{C}$

[Notation. M,N,L,... range over C

$A, B, C, \ldots$ range over $\underline{C}$

$|A| \in C$ is the same term as $A \in \underline{C}$ except that underlining is left out].

(iii) Reduction on $\underline{C}$ is defined by the following contraction rules:

$$\begin{aligned}
KAB &\rightarrow A \\
SABC &\rightarrow AC(BC) \\
\underline{M}B &\rightarrow \underline{\underline{M|B|}} \\
\underline{\underline{M}}B &\rightarrow \underline{\underline{M|B|}} \\
\underline{M} &\rightarrow \underline{N} \qquad \text{if } M \twoheadrightarrow_{CL} N \\
\underline{\underline{M}} &\rightarrow \underline{\underline{N}} \qquad \text{if } M \twoheadrightarrow_{CL} N
\end{aligned}$$

$\twoheadrightarrow_{\underline{CL}}$ is the resulting reduction.

2.3 LEMMA. Let $A \in \underline{C}$, $M, N \in C$ be given such that $|A| = M \twoheadrightarrow_{CL} N$ Then there exists a $B \in \underline{C}$ such that $A \twoheadrightarrow_{\underline{CL}} B$ and $|B| = N$.

$$\begin{tikzcd}
A \arrow[r, "|.|"] \arrow[d, dashed, two heads, "\underline{CL}"'] & M \arrow[d, two heads, "CL"] \\
B \arrow[r, dashed, two heads, "|.|"'] & N
\end{tikzcd}$$

Proof. Induction on the generation of $M \twoheadrightarrow_{CL} N$. $\square$

2.4 *Definition.* $\varphi_{vw} : \underline{C} \rightarrow C$ is defined by

$$\varphi_{vw}(M) = M \qquad \varphi_{vw}(\underline{M}) = v \qquad \varphi_{vw}(\underline{\underline{M}}) = w$$

$$\varphi_{vw}(AB) = \varphi_{vw}(A)\, \varphi_{vw}(B)$$

2.5 LEMMA. Assume $A \twoheadrightarrow_{\underline{CL}} B$. Let A^+ be obtained from $\varphi_{vw}(A)$ by replacing every occurrence of w by $v\vec{\square}$ where $\vec{\square}$ is some non-empty sequence of CL-terms ($\vec{\square}$ may vary).
Then there exists a $B^+ \in C$, in a similar way obtained from $\varphi_{vw}(B)$ such that $A^+ \twoheadrightarrow_{CL} B^+$.

Proof. Induction on the generation of $A \twoheadrightarrow_{\underline{\mathrm{CL}}} B$. □

2.6 LEMMA. Assume $A \twoheadrightarrow_{\underline{\mathrm{CL}}} B$. Then

$$\forall \underline{N} \subset B \quad \exists \underline{M} \subset A \quad M \twoheadrightarrow_{\mathrm{CL}} N,$$

where $\subset$ denotes the subterm relation.

Proof. Induction on the generation of $A \twoheadrightarrow_{\underline{\mathrm{CL}}} B$. □

2.7 *Proof of 2.1*. Suppose $(\mathfrak{M},\lambda^*)$ is a λ-algebra. Then, as an easy consequence of some λ-calculus theorems, two closed terms S',K' would exist such that the following equations are provable in CL:

$$\mathit{I} \quad S'xyz = xz(yz)$$

$$\mathit{II} \quad K'xy = x$$

$$\mathit{III} \quad S'(K'x)(K'y) = K'(xy).$$

Using *III* and the Church-Rosser theorem for $\twoheadrightarrow_{\mathrm{CL}}$ we find reductions *R* and *R'* such that

$$\begin{tikzcd}
M \underset{\mathrm{def}}{\equiv} S'(K'x)(K'y) \arrow[rr, equal] \arrow[dr, two heads, "R"', "\mathrm{CL}" near end] & & K'(xy) \arrow[dl, two heads, "R'", "\mathrm{CL}"' near end] \\
& N &
\end{tikzcd}$$

Because of *R'* we have

(*) *1.* (xy) occurs in N

2. If x or y occurs in N, then it does in the combination (xy).

Let $A \underset{\mathrm{def}}{\equiv} S'(\underline{K'x})(K'y) \in \underline{C}$ and

$$A^+ \underset{\mathrm{def}}{\equiv} S'v(K'y) \equiv \varphi_{vw}(A), \text{ with v,w fresh variables.}$$

By lemmas 2.3 and 2.5 one has:

$$\begin{tikzcd}
A^+ \arrow[d, "R''"', "\mathrm{CL}"] & A \arrow[l, "\varphi^+_{vw}"'] \arrow[r, "|\cdot|"] \arrow[d, "\underline{R}", "\underline{\mathrm{CL}}"'] & M \arrow[d, "R", "\mathrm{CL}"'] \\
B^+ & B \arrow[l, "\varphi^+_{vw}"] \arrow[r, "|\cdot|"'] & N
\end{tikzcd}$$

Using (*), lemma 2.6 and some easy arguments we deduce:

(**) Every singly underlined subterm of B occurs in the form $\underline{x}y$.

Passing from B to B^+ we find, using (**):

(***) For every occurrence of v in B^+ we have:
v is active, i.e. occurs as $(v\Box)$ for some $\Box$,
or v is not outermost, i.e. v occurs in $\vec{\Box}$, where $\vec{\Box}$ occurs as $v\vec{\Box}$.

Now we have

$$\begin{tikzcd}
A^+z \arrow[r, equal] \arrow[d, "R''z"', "\mathrm{CL}"] & vzy \\
B^+z \arrow[ur, dashed, "\widetilde{R}"', "\mathrm{CL}"] &
\end{tikzcd}$$

where z is a fresh variable.
By the Church-Rosser theorem again $\widetilde{R}$ exists. But this reduction contradicts (***), because no z can intrude at a v.
So our initial assumption is false. $\Box$

§3 A λ-ALGEBRA THAT CANNOT BE MADE INTO A λ-MODEL

Let $\mathcal{H}$ be the λ-theory obtained by identifying the unsolvables, see Barendregt (1977).

3.1 THEOREM. $\mathfrak{M}^0(\mathcal{H})$ is a λ-algebra that cannot be made into a λ-model (by changing the operator λ^*).

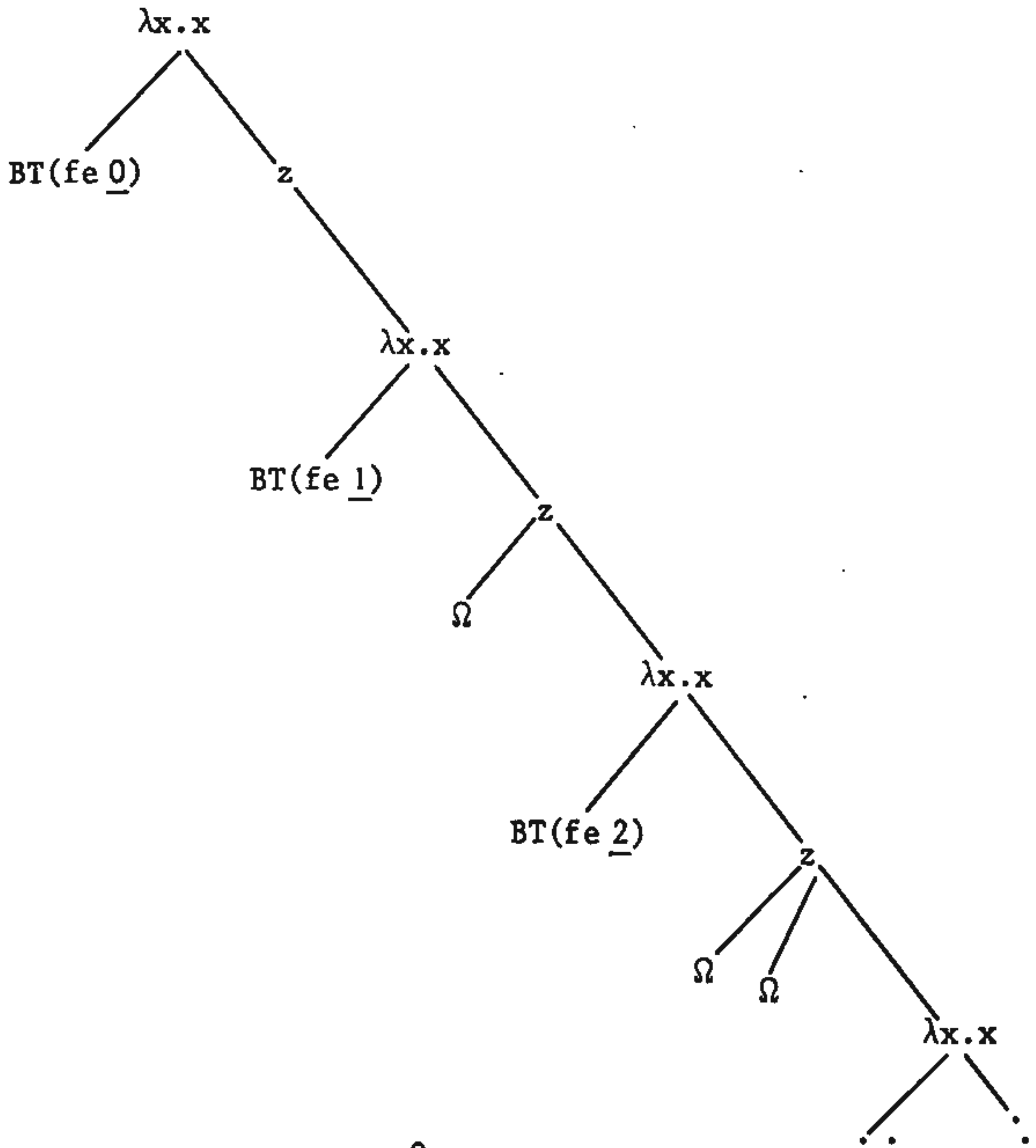
The proof occupies 3.2 - 3.5.

3.2 LEMMA. $\mathcal{H}$ is Σ^0_2 - complete as follows.
Let R(a,b,c) be recursive. Then there exist $F, F' \in \Lambda^0$ such that

$$\exists b \forall c\, R(a,b,c) \iff \mathcal{H} \vdash F\underline{a} = F'\underline{a}\ .$$

Proof. See Barendregt et al. (1978). □

3.3 *Definition.* Let A be some closed term, to be constructed by familiar techniques, such that the Böhm-tree of Afez (where f, e,z are variables) looks like:



3.4 LEMMA. For all $F, F' \in \Lambda^0$ and all $e \in \omega$:

$$\forall n \quad \mathcal{H} \vdash F\underline{e}\,\underline{n} = F'\underline{e}\,\underline{n} \iff \forall Z \in \Lambda^0 \quad \mathcal{H} \vdash AF\underline{e}\,Z = AF'\underline{e}\,Z\ .$$

Proof. $\Rightarrow$: Use the fact that for all $Z \in \Lambda^0$ there exists an $n \in \omega$ such that $\mathcal{H} \vdash Z\Omega^{\sim n} = \Omega$, see Barendregt et al. (1978).

$\Leftarrow$: Usual Böhm out technique. □

3.5 *Proof of 3.1.* As noticed in fact 1.14(ii) $\mathfrak{M}^0(\mathcal{H})$ with the standard λ^*-operator from 1.13(ii) is a λ-algebra.
Now assume some (other) λ^* exists such that $(\mathfrak{M}^0(\mathcal{H}),\lambda^*)$ is a λ-model. Using the two lemmas 3.2 and 3.4, we can reduce any Π^0_3-predicate to a Σ^0_2-predicate which gives a recursion theoretic contradiction, thereby proving the theorem.

So let $R(a,b,c,e)$ be any recursive predicate. Then for suitable $F,F' \in \Lambda^0$ (by lemma 3.2) and σ we have:

$$\begin{array}{lll} & \forall a \exists b \forall c\ R(a,b,c,e) & \\ \Longleftrightarrow & \forall a\ \ \mathcal{H} \vdash F\,\underline{e}\,\underline{a} = F'\underline{e}\,\underline{a} & \text{(Lemma 3.2)} \\ \Longleftrightarrow & \forall Z \in \Lambda^0\ \ \mathcal{H} \vdash AF\,\underline{e}\,Z = AF'\underline{e}\,Z & \text{(Lemma 3.4)} \\ \Longleftrightarrow & \forall Z \in \Lambda^0\ \ \mathfrak{M}^0(\mathcal{H}) \models AF\,\underline{e}\,Z = AF'\underline{e}\,Z & \\ \Longleftrightarrow & \mathfrak{M}^0(\mathcal{H}) \models \forall z\ AF\,\underline{e}\,z = AF'\underline{e}\,z & \\ \Longleftrightarrow & \mathfrak{M}^0(\mathcal{H}) \models \lambda z.AF\,\underline{e}\,z = \lambda z.AF'\underline{e}\,z & (\text{by } \xi_1) \\ \Longleftrightarrow & \mathcal{H} \vdash \lambda z.AF\,\underline{e}\,z = \lambda z.AF'\underline{e}\,z\,. & \end{array}$$

By lemma 3.2 again this is a Σ^0_2-predicate, using that the map $e \to (\lambda z.AF\,\underline{e}\,z)$ is effective. □

§4 EMBEDDING λ-ALGEBRAS IN λ-MODELS

4.1 THEOREM. Every λ-algebra $\mathfrak{M}$ can be embedded in a λ-model $\mathfrak{N}$.

The proof occupies 4.2 - 4.5.

4.2 *Definition.* $\mathrm{Th}(\underset{\sim}{\mathfrak{M}}) = \{M = N \mid M,N \in \Lambda^0_{\mathfrak{M}} \text{ and } \mathfrak{M} \models M = N\}$, where $\mathfrak{M}$ is a λ-algebra and $\Lambda^0_{\mathfrak{M}}$ is defined in 1.6(ii).

4.3 LEMMA. For any λ-algebra $\mathfrak{M}$ we have:

$$\mathrm{Th}(\underset{\sim}{\mathfrak{M}}) \vdash M = N \;\Rightarrow\; \mathfrak{M} \models \lambda\vec{x}.M = \lambda\vec{x}.N,$$

where $\{\vec{x}\} \supseteq \mathrm{FV}(MN)$.

Proof. Induction on the length of proof of M = N. □

4.4 COROLLARY. Con($\mathrm{Th}(\underset{\sim}{\mathfrak{M}})$), i.e. $\mathrm{Th}(\underset{\sim}{\mathfrak{M}})$ is consistent.

Proof. Indeed, according to 4.3 one has $\mathrm{Th}(\underset{\sim}{\mathfrak{M}}) \nvdash \underline{0} = \underline{1}$. □

REMARK. For $\mathrm{Th}^{\mathrm{open}}(\underset{\sim}{\mathfrak{M}})$ containing open equations true in $\mathfrak{M}$, this is false, see. 4.9. This may seem strange, but is caused by the fact that truth in λ-algebra's is not closed under deduction (ξ does not need to hold).

4.5 *Proof of 4.1.* Let $T \equiv \mathrm{Th}(\underset{\sim}{\mathfrak{M}})$ and consider $\mathfrak{N} \equiv \mathfrak{M}(T)$.

Claim: The map $\eta : |\mathfrak{M}| \to |\mathfrak{N}|$, defined by $\eta(a) = [\underline{a}]_T$ is an embedding of $\mathfrak{M}$ into $\mathfrak{N}$.

Indeed,

(i) η is injective (use $\mathfrak{M} \models T$).

(ii) η is a homomorphism with respect to · (use $\mathfrak{M} \models T$).

(iii) $\mathfrak{N} \models (\lambda^*\vec{x}.A)[\vec{c}_a := \vec{c}_{\eta a}] = \lambda^*\vec{x}.(A[\vec{c}_a := \vec{c}_{\eta a}])$

(make some simple calculations).

Now we will turn to the question "$\mathfrak{M} \models \Gamma \Rightarrow \mathrm{Con}(\Gamma)$?".
Since the rule ξ does not need to be valid in a λ-algebra $\mathfrak{M}$ it may happen that $\mathrm{Th}^{\mathrm{open}}(\mathfrak{M})$ is actually inconsistent, hence answering the above question with "no". In this subsection such an $\mathfrak{M}$ is constructed using an idea from Jacopini (1975).

4.6 *Definition.* Let $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$.
Define on Λ the following reduction rules γ:

$$\Omega\,\underline{0}\,Z \to \Omega\,\underline{1}\,Z \quad \text{for all } Z \in \Lambda^0$$

$$\Omega(\lambda x.\Omega\ \underline{0}\ x) \to \underline{0}$$
$$\Omega(\lambda x.\Omega\ \underline{1}\ x) \to \underline{1}$$

4.7 LEMMA.

(i) Pure γ-reduction is Church-Rosser.

(ii) β- and γ-reduction commute (in the sense of the lemma of Hindley-Rosen, see Staples, (1975)).

(iii) $\beta\gamma$-reduction is Church-Rosser.

Proof.

(i) Note that distinct γ-redexes in a term are always disjoint. Therefore the diamond property for γ-reduction is immediate.

(ii) By some case distinction one easily shows

$$\begin{tikzcd} \cdot \arrow[r, "\gamma"] \arrow[d, "\beta"'] & \cdot \arrow[d, dashed, "\beta"] \\ \cdot \arrow[r, dashed, "\gamma"'] & \cdot \end{tikzcd} \quad ;$$

from this one has

$$\begin{tikzcd} \cdot \arrow[r, two heads, "\gamma"] \arrow[d, two heads, "\beta"'] & \cdot \arrow[d, two heads, "\beta"] \\ \cdot \arrow[r, dashed, two heads, "\gamma"'] & \cdot \end{tikzcd}$$

by a diagram chase.

(iii) By the lemma of Hindley-Rosen. $\square$

4.8 COROLLARY. Let T be the extension of the λ-calculus axiomatized by

$$\Omega\ \underline{0}\ Z = \Omega\ \underline{1}\ Z \quad , Z \in \Lambda^0$$
$$\Omega(\lambda x.\Omega\ \underline{i}\ x) = \underline{i}\ , \ i \in \{0,1\}.$$

Then T is consistent.

Proof. By 4.7. □

4.9 PROPOSITION. Let $\mathfrak{M} \equiv \mathfrak{M}^0(T)$. Then $\mathfrak{M}$ is a λ-algebra with inconsistent $\mathrm{Th}^{\mathrm{open}}(\underset{\sim}{\mathfrak{M}})$.

Proof. Since T is consistent, $\mathfrak{M}$ is a λ-algebra. Note that $\mathfrak{M} \models T$ and $\mathfrak{M} \models \Omega\,\underline{0}\,x = \Omega\,\underline{1}\,x$, since x ranges over $\mathfrak{M}^0(T)$. Hence

$$\mathrm{Th}^{\mathrm{open}}(\mathfrak{M}) \vdash \underline{0} = \Omega(\lambda x.\Omega\,\underline{0}\,x) = \Omega(\lambda x.\Omega\,\underline{1}\,x) = \underline{1},$$

which means inconsistency. □

REMARK. If $\mathfrak{M}$ is a λ-model, then $\mathrm{Th}^{\mathrm{open}}(\mathfrak{M})$ is always consistent.

Finally it will be shown that in general λ-models cannot be made extensional. This is different from the situation for the typed λ-calculus, see Troelstra (1973), theorem 2.4.5. The result is taken from Barendregt (1971).

4.10 PROPOSITION. There is a consistent extension T of the λ-calculus such that $T\eta = T +$ extensionality is inconsistent.

Proof. Let T be the λ-calculus extended by

$$\Omega(\lambda x.x) = \underline{0}$$
$$\Omega(\lambda xy.xy) = \underline{1}$$

As in 4.7 and 4.8 one shows that T is consistent. Since $\lambda\eta \vdash \lambda x.x = \lambda xy.xy$ it follows that $T\eta \vdash \underline{0} = \underline{1}$ and hence is inconsistent. □

4.11 COROLLARY. There is a λ-model $\mathfrak{M}$ such that $\mathfrak{M}$ cannot be embedded into, nor mapped onto an extensional λ-model.

Proof. Take $\mathfrak{M} \equiv \mathfrak{M}(T)$, where T is as in 4.10. If $\mathfrak{M} \hookrightarrow \mathfrak{M}_1$ or $\varphi : \mathfrak{M} \twoheadrightarrow \mathfrak{M}_1$, onto, with $\mathfrak{M}_1$ extensional, then

$$T\eta \subseteq \mathrm{Th}(\mathfrak{M}_1).$$

Hence by 4.10 $\mathrm{Th}(\mathfrak{M}_1)$ is inconsistent, a contradiction. □

REFERENCES

Barendregt, H.P. (1971).
Some extensional termmodels for combinatory logics and λ-calculi, thesis, Utrecht.

Barendregt, H.P. (1977).
The type free lambda calculus, in: Handbook of mathematical logic, ed. J. Barwise, North Holland 1977, 1091-1132.

Barendregt, H.P., J.A. Bergstra, J.W. Klop and H. Volken (1978).
Degrees of sensible λ-algebras, JSL, 43 no. 1, 1978, 45-55.

Hindley, J.R. and G. Longo (1980).
Lambda calculus models and extensionality, to appear in Zeit.f.Math. Logik Grund. Math, 26 no. 2, 1980.

Jacopini, G.
Principio di estensionalita nel calcolo dei combinatori, Calcolo II, 1974 no. 4, 1975, 465-471.

Plotkin, G. (1974).
The lambda calculus is omega incomplete, JSL, 39 no. 2, 1974, 313-317.

Sanchis, L.E. (1979).
Reducibilities in two models for combinatory logic, JSL, 44 no. 2, 1979, 221-234.

Staples, J. (1975).
Church-Rosser theorems for replacement systems, in: Algebra and Logic, ed. J.N. Crossley, LNM 450, Springer, Berlin, 1975, 291-307.

Troelstra, A.S. (1973).
Metamathematical investigations of intuitionistic arithmetic and analysis, LNM 344, Springer, Berlin, 1973.

Visser, A. (1979).
Numerations, λ-calculus and arithmetic, preprint nr. 124, Sept. 1979, Department of Mathematics, Utrecht, The Netherlands. To appear in this Volume, 1980.

EQUALITY OF LAMBDA TERMS IN THE MODEL T^{ω}.

Henk Barendregt* and Giuseppe Longo**

**Mathematical Institute, Budapestlaan 6*
3508 TA Utrecht, The Netherlands.
***Istituto di Scienze dell'Informazione*
Corso Italia 40, 56100 Pisa, Italy.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

ABSTRACT

The λ-calculus model T^{ω} is introduced in [6]. For a λ-term M, let BT(M) be its Böhm tree introduced in [1]. Establishing a conjecture of Plotkin it will be proved (in the spirit of an improved version of [4]) that

(i) $T^{\omega} \models M = N \Longleftrightarrow BT(M) = BT(N)$,

(ii) $T^{\omega} \models M \sqsubseteq N \Longleftrightarrow BT(M) \subseteq BT(N)^{1}$.

This in contrast with the model $P\omega$, where e.g. $P\omega \models \lambda x.x \subseteq \lambda xy.xy$.

CONTENTS

PRELIMINARIES AND NOTATIONS

The model T^{ω} will be introduced in a way differing from [6]. Therefore acquaintance with that paper is not required. Notations and concepts are taken from [1]; hence the reader is supposed to be familiar with this latter paper, in particular §§4, 6 and 7.

The following notations will be used.

$\mathbb{N} = \{0,1,2,\ldots\}$ natural numbers.

$P\omega = \{A \mid A \subseteq \mathbb{N}\}$, $\bar{A} = \{x \mid x \in \mathbb{N} \wedge x \notin A\}$ for $A \subseteq \mathbb{N}$.

Λ set of λ-terms, Λ^0 set of closed λ-terms.

$\lambda\!\!\!\lambda$ meta lambda (e.g. if for $M \in \Lambda$ one writes $\#M$ for the Gödel-number of M, then one can define $\lambda\!\!\!\lambda M.\#M$).

§0 COMPLETE PARTIAL ORDERS

Instead of working with complete lattices in order to construct λ-calculus models, see [7], one may use complete partial orders. This is convenient since the latter structures are more abundant. Since most of the results of this § are analogous to those for complete lattices as treated e.g. in [1], proofs are often omitted.

0.1 *Definition.* Let $X = \langle X, \sqsubseteq \rangle$ be a partially ordered set.

(i) $Y \subseteq X$ is <u>directed</u> iff $Y \neq \emptyset$ and $\forall x,y \in Y\ \exists z \in Y$ $[x \sqsubseteq z \wedge y \sqsubseteq z]$.

(ii) X is a <u>complete partial order</u> (<u>cpo</u>) iff

1. $Y \subseteq X$ is directed $\Rightarrow$ the supremum $\bigsqcup Y \in X$ exists
2. there is a least element $\perp \in X$.

0.2 *Definition.* Let X be a cpo. The <u>Scott topology</u> on X is defined as follows.

$O \subseteq X$ is <u>open</u> iff 1. $x \in O \wedge x \sqsubseteq y \Rightarrow y \in O$
and 2. $[Y \subseteq X$ is directed and $\bigsqcup Y \in O]$
$\Rightarrow Y \cap O \neq \emptyset$.

As in [1] p. 1110 one shows that this defines a T_0 topology which in general is not T_1.

0.3 PROPOSITION. Let X,X' be cpo's and $f: X \to X'$. Then f is continuous $\Leftrightarrow$ $f(\bigsqcup Y) = \bigsqcup f(Y)$ for all directed $Y \subseteq X$.

In particular all continuous f are monotonic.

0.4 *Definition.* Let X,X' be cpo's.

(i) $X \times X'$ is the cartesian product partially ordered by $\langle x,x' \rangle \sqsubseteq \langle y,y' \rangle \Leftrightarrow x \sqsubseteq y \wedge x' \sqsubseteq y'$.

(ii) $[X \to X'] = \{f: X \to X' \mid f \text{ continuous}\}$ with the pointwise partial order.

As in [1] 4.6 one shows that $X \times X'$ and $[X \to X']$ are again cpo's.

0.5 LEMMA. $f: X \times X' \to X''$ is continuous $\Longleftrightarrow$ f is continuous in its variables separately.

0.6 PROPOSITION. (Continuity of application and abstraction).

(i) Define $A_p: [X \to X'] \times X \to X'$ by $A_p(f,x) = f(x)$. Then A_p is continuous.

(ii) For $f \in [X \times X' \to X'']$ define $g_f(x) = \lambda x'.f(x,x') \in [X' \to X'']$. Then $g_f: X \to [X' \to X'']$ is continuous.

In fact the category of cpo's with continuous maps is cartesian closed.

0.7 *Definition.* Let $\langle X, \cdot \rangle$ be a structure with a binary operation called application and let $f: X^n \to X$. Then

(i) f is <u>representable</u> iff $\exists a \forall x_1 \dots x_n \ \ a x_1 \dots x_n = f(x_1, \dots, x_n)$ (using association to the left).

(ii) f is <u>algebraic</u> iff $f(x_1, \dots, x_n)$ can be defined from $x_1, \dots, x_n$ and constants from X using application.

The following is taken from [9].

0.8 THEOREM. Let for every $n \geq 0$ a class F_n of n-ary functions be given on $X = \langle X, \cdot \rangle$ such that

(1) $\cup F_n$ contains all algebraic functions on X and is closed under substitution of constants.

(2) There exists a function $\Box : F_1 \to X$ such that $\forall f \in F_1 \ \forall a \in X \quad \Box f.a = f(a)$.

(3) $\forall f \in F_{n+1} \ \lambda \vec{y}. \Box(\lambda x.f(x,\vec{y})) \in F_n$, where x is any one of the arguments of f and $\vec{y}$ are the remaining ones.

Then $\langle X, \cdot \rangle$ can be considered as a w.e. λ-algebra (see [1], p. 1099), such that $\cup F_n$ is exactly the class of representable functions on X.

Proof. For $f \in F_{k+1}$, define $\lambda x.f(x,\vec{y}) = \Box(\lambda x.f(x,\vec{y}))$.

Then

(4) $\forall x,\vec{y}(\lambda x.f(x,\vec{y}))x = f(x,\vec{y})$

and

(5) $\lambda\vec{y}.(\lambda x.f(x,\vec{y})) \in F_k$.

It follows that every $f \in \bigcup F_n$ is representable: let $a = \lambda\vec{x}.f(\vec{x})$, with association to the right for $\lambda\vec{x}$. Then by (5) $a \in F_0 \subseteq X$. By (4) $\forall\vec{x}\ a\vec{x} = f(\vec{x})$.

For every $a \in X$, denote the corresponding constant by $\underline{a}$. Given a term $t = t(x,y)$ in the language of $\langle X,\cdot\rangle$ (using constants) define

$$\lambda^* x.t(x,\vec{y}) = \underline{\lambda\vec{y}x.[\![t(\underline{x},\underline{\vec{y}})]\!]}\vec{y}$$

where $[\![\ \ ]\!]$ is the ordinary interpretation of closed terms in $\langle X,\cdot\rangle$. Then for any valuation ρ one has

$$[\![\lambda^* x.t(x,\vec{y})]\!]_\rho = \lambda x.[\![t(\underline{x},\underline{\vec{y}})]\!]_\rho$$

Therefore

$$\langle X,\cdot\rangle \models (\lambda^* x.t(x,y))x = t(x,y)$$

i.e. $\langle X,\cdot,\lambda^*\rangle$ is a pre-λ-algebra. Moreover since $\Box$ acts on functions in extenso, one has

$$(\xi)\quad \langle X,\cdot,\lambda^*\rangle \models [\forall x t(x) = t'(x)] \to \lambda^* x.t(x) = \lambda^* x.t'(x),$$

hence $\langle X,\cdot\rangle$ is a w.e. λ-algebra[2].

0.9 COROLLARY. Let X be a cpo such that for some $\Box \in [[X \to X] \to X]$ and $\mathrm{Fun} \in [X \to [X \to X]]$ one has $\mathrm{Fun}(\Box(f)) = f$ for all $f \in [X \to X]$. Define for $x,y \in X$

$$x.y = \mathrm{Fun}(x)(y).$$

Then $\langle X,\cdot\rangle$ can be considered as a w.e. λ-algebra in which exactly the continuous functions are representable.

Proof. Let $F_n = [X^n \to X]$. Then by 0.6 these families satisfy (1), (2) and (3) of 0.8. $\Box$

If $\mathfrak{M}$ is a λ-algebra let $\Lambda(\mathfrak{M})$ be the class of λ-terms using constants from $\mathfrak{M}$ (for a in the domain of $\mathfrak{M}$, let $\underline{a}$ be the corresponding constant).

0.10 COROLLARY. Let X be as in 0.9 and let $\mathfrak{M} = \langle X, \cdot \rangle$. For $M \in \Lambda(\mathfrak{M})$ the interpretation in $\mathfrak{M}$ can be defined as follows (relative to a valuation ρ)

$$\begin{aligned} [\![x]\!]_\rho &= \rho(x) \\ [\![\underline{a}]\!]_\rho &= a \\ [\![MN]\!]_\rho &= [\![M]\!]_\rho [\![N]\!]_\rho \\ [\![x.M]\!]_\rho &= \lambda x.[\![M(\underline{x})]\!]_\rho \end{aligned}$$

where $\lambda x.f(x) := \square(\lambda x.f(x))$.

Note that we are using "x" in two different senses in the fourth equation above; on the left, it is a normal variable of $\Lambda(\mathfrak{M})$, and on the right it is a variable of the informal meta-language, ranging over members of X. In this notation $[\![\lambda x.x]\!]_\rho = \lambda x.[\![\underline{x}]\!]_\rho = \lambda x.x$. Indeed for all $M \in \Lambda^0$ one has $[\![M]\!] = M$, using the proper interpretation.

§1 THE MODEL T^ω

Now the model T^ω will be constructed. In doing so we use notations rather different from [6].

Let T be the c.p.o. $\overset{0\ \ 1}{\searrow\!\!\swarrow}_{\!\perp}$ (i.e. $0 \sqsupseteq \perp \sqsubseteq 1$). Then T^ω is the c.p.o. of all functions from $\mathbb{N}$ into T, with respect to the pointwise ordering.

In what follows we shall identify T^ω with the class of disjoint pairs of subsets of $\mathbb{N}$. In fact a map $f: \mathbb{N} \to T$, is uniquely determined by two non overlapping subsets of $\mathbb{N}$; namely $a_- = \{n \mid f(n) = 0\}$ and $a_+ = \{n \mid f(n) = 1\}$. Of course $\overline{a_- \cup a_+} = \{n \mid f(n) = \perp\}$.

1.1 *Definition.*

(i) $T^\omega = \{\langle A,B \rangle \mid A,B \in P\omega \text{ and } A \cap B = \emptyset\}$

(ii) If $a \in T^\omega$, then write $a = \langle a_-, a_+ \rangle$.

(iii) T^ω is partially ordered by

$$\langle a_-, a_+ \rangle \sqsubseteq \langle b_-, b_+ \rangle \iff a_- \subseteq b_- \wedge a_+ \subseteq b_+.$$

Clearly T^ω is a cpo: $\bigsqcup_i a_i = \langle \cup a_{i-}, \cup a_{i+} \rangle$ for directed

$\{a_i \mid i \in I\}$.

1.2 *Definition.*

(i) Let $a \in T^\omega$. Then $|a| = a_- \cup a_+$.

a is called <u>finite</u> if $|a|$ is so.

(ii) $\{b_n \mid n \in \mathbb{N}\}$ is an enumeration of the finite elements of T^ω defined as follows:

$$b_n = \langle A,B\rangle \text{ iff } n = \sum_{i\in A} 2.3^i + \sum_{i\in B} 3^i.$$

(In order to compute b_n, one writes n in ternary notation.)

(iii) For $n,m \in \mathbb{N}$ define:

$$(n,m) = \tfrac{1}{2}(n+m)(n+m+1) + m.$$

This is a bijection from $\mathbb{N}^2$ onto $\mathbb{N}$. Moreover define

$$(-n;m) = (n,2m) \text{ and } (+n;m) = (n,2m+1).$$

(iv) For $n,m \in \mathbb{N}$ define

$$n\uparrow m \Longleftrightarrow \exists a \in T^\omega \quad b_n, b_m \sqsubseteq a.$$

Moreover define the sets

$$D_{(+n;m)} = \{(-n';m) \mid n' \uparrow n \wedge (-n';m) \leqslant (+n;m)\}$$

$$D_{(-n;m)} = \{(+n';m) \mid n' \uparrow n \wedge (+n';m) \leqslant (-n;m)\}.$$

Note that the sets D... are all finite.

From now on we use the following conventions.

a,b,c....	range over T^ω
n,m,...,p,q,...	range over $\mathbb{N}$
A,B,...	range over $P\omega$
x,y,...	are bound variables.

1.3 *Definition.*

(i) Let $f \in [T^\omega \to T^\omega]$ define $\lambda x.f(x) := \Box f :=$

$$:= \langle \{(-n;m) \mid m \in f(b_n)_-\} \cup \{(+n;m) \mid m \in f(b_n)_+\},$$
$$\{(-n;m) \mid \exists b_{n'} \sqsupseteq b_n \quad m \in f(b_{n'})_+\} \cup$$
$$\{(+n;m) \mid \exists b_{n'} \sqsupseteq b_n \quad m \in f(b_{n'})_-\}\rangle.$$

In short $:= \langle \{(\mp n;m) \mid m \in f(b_n)_\mp\}, \{(\mp n;m) \mid \exists b_{n'} \sqsupseteq b_n \quad m \in f(b_{n'})_\pm\}$.

(ii) For $c,d \in T^\omega$ define $c.d := \mathrm{Fun}(c)(d) :=$

$$= \langle \{m \mid \exists b_n \sqsubseteq d(-n;m) \in c_- \wedge D_{(-n;m)} \subseteq c_+\},$$
$$\{m \mid \exists b_n \sqsubseteq d(+n;m) \in c_- \wedge D_{(+n;m)} \subseteq c_+\}\rangle.$$

In (i) the LHS of $\square f$ (i.e. $\square f_-$) describes the behaviour of the function on finite elements of T^ω (compare with the definition of "graph" in the model $P\omega$). For continuous f, this behaviour completely determines f. One has also to take into account that if, say, $m \in f(b_n)_\pm$ then for no $b_{n'} \sqsupseteq b_n$ $m \in f(b_{n'})_\mp$ (because of monotonicity of f): this is what $\square f_+$ does.
In (ii) the sets $D_{(\mp n;m)}$ are used in order to insure that $c.d_-$ and $c.d_+$ are disjoint for any c,d. In fact for an important class of c's the D's can be omitted, namely when $c = \square f$, for some continuous f (see 1.6).

1.4 LEMMA.

(i) $\forall c,d \in T^\omega \quad c.d \in T^\omega$

(ii) $\forall f \in [T^\omega \to T^\omega] \quad \square f \in T^\omega$.

Proof.

(i) Suppose $m \in (c.d)_- \cap (c.d)_+$. Then

$$\exists b_{n_1} \sqsubseteq d(-n_1;m) \in c_- \wedge D_{(-n_1;m)} \subseteq c_+$$
$$\exists b_{n_2} \sqsubseteq d(+n_2;m) \in c_- \wedge D_{(+n_2;m)} \subseteq c_+.$$

Let, say, $(-n_1;m) \leqslant (+n_2;m)$. Then $(-n_1;m) \in D_{(+n_2;m)} \subseteq c_+$. But also $(-n_1;m) \in c_-$, contradicting $c_- \cap c_+ = \emptyset$.

(ii) Similarly. $\square$

1.5 PROPOSITION.

(i) $\cdot : (T^\omega)^2 \to T^\omega$ is continuous

(ii) $\mathrm{Fun}: T^\omega \to [T^\omega \to T^\omega]$ is continuous

(iii) $\square : [T^\omega \to T^\omega] \to T^\omega$ is continuous.

Proof.

(i) By 0.5 it is sufficient to show that c.d is continuous in c and in d. This is done using 0.3:

$$\begin{aligned} c.\sqcup d_i &= \langle\{m \mid \exists b_n \sqsubseteq \sqcup d_i \ldots \\ &= \sqcup_i \langle \{m \mid \exists b_n \sqsubseteq d_i \ldots \\ &= \sqcup_i\ c.d_i. \end{aligned}$$

(Since the b_n are finite and the d_i are directed one has

$b_n \sqsubseteq \sqcup d_i \Rightarrow \exists d_{i_1},\ldots,d_{i_n},\ d_{i_p}\ b_n \sqsubseteq d_i \sqcup \ldots \sqcup d_{i_n} \sqsubseteq d_{i_p}$.)

Similarly $\sqcup c_i.d = \sqcup(c_i.d)$.

(ii) By (i) using 0.6(ii).

(iii) Similar to (i). $\square$

The following lemma shows that for $c = \lambda x.cx(=\square(\lambda x.cx))$ the definition of ca can be given more simply.

1.6 LEMMA. Let $f \in [T^\omega \to T^\omega]$. Then

$$\begin{aligned} \square f.a = \langle &\{m \mid \exists b_n \sqsubseteq a(-n;m) \in \square f_-\}, \\ &\{m \mid \exists b_n \sqsubseteq a(+n;m) \in \square f_-\} \ . \end{aligned}$$

Proof. It suffices to show

$$(\mp n;m) \in \square f_- \Rightarrow D_{(\mp n;m)} \subseteq \square f_+.$$

Indeed, let $(-n;m) \in \square f_-$ and let $(+n';m) \in D_{(-n;m)}$. Then $m \in f(b_n)_-$ and $n' \uparrow n$. Take $b_{n''} \sqsupseteq b_n, b_{n'}$. Then by monotonicity of f one has $m \in f(b_{n''})_-$, i.e. $(+n';m) \in \square f_+$.

Similarly for $(+n;m)$. $\square$

1.7 COROLLARY. $\forall a\ \square f.a = f(a)$.

Proof.
$$\begin{aligned} (\square f.a)_- &= \{m \mid \exists b_n \sqsubseteq a\ \ m \in f(b_n)_-\} \\ &= \cup\{f(b_n)_- \mid b_n \sqsubseteq a\} = f(a)_- \ , \end{aligned}$$

by continuity of f.

Similarly $(\square f.a)_+ = f(a)_+$. $\square$

1.8 THEOREM. $\langle T^\omega, \cdot\rangle$ is a w.e. λ-algebra.

Proof. By 0.9, 1.5(ii), (iii) and 1.7.

Finally we need some facts about the hardware (n,m), b_n and $D_{(\pm n;m)}$.

1.9 LEMMA.

(i) $(0,0) = 0$; $(1,0) = 1$; $(0,1) = 2$.

(ii) $b_0 = \langle \emptyset, \emptyset \rangle = \perp$

$b_1 = \langle \emptyset, \{0\} \rangle$

$b_2 = \langle \{0\}, \emptyset \rangle$

(iii) $D_{(-0;m)} = \emptyset$; $D_{(+0;m)} = \{(-0;m),(-1;m)\}$ for all m.

Proof. Easy. $\square$

1.10 LEMMA.

(i) $n,m \leqslant (n,m)$

(ii) $m = (n,m) \Longleftrightarrow n = m = 0$;

$n = (n,m) \Longleftrightarrow n = m = 0 \vee (n = 1 \wedge m = 0)$.

(iii) $|b_n| \subseteq \{0,\dots,n-1\}$

Proof. Easy. $\square$

NOTATIONS. $\vec{a}^k$ denotes a sequence of length k; $ab_- = (ab)_-$.

§2 THE APPROXIMATION THEOREM.

For the characterization of equality in the models D_∞ and $P\omega$, the approximation theorem plays an essential role, see [1]. As is stated in [6], this theorem also holds in T^{ω}.
Since the proof is a tiny bit different, we give here some of the details, but refer mostly to [1].

2.1 *Definition.* Let $n \in \mathbb{N}$ and $a \in T^{\omega}$. Then

(i) $\underline{n} = \{0,\dots,n\}$

(ii) $a^n = \langle a_- \cap \underline{n},\ a_+ \cap \underline{n} \rangle$

2.2 LEMMA. Let $a,c \in T^{\omega}$. Then

(i) $a = \bigcup_n a^n$

(ii) $(a^n)^m = a^{\min\{n,m\}}$

(iii) $a^0 c \sqsubseteq (a\perp)^0$

(iv) $a^{n+1} c \sqsubseteq (a\, c^n)^n$

Proof. *(i)*, *(ii)* Trivial.

(iii) By definition

$$a^0c = \langle \{m \mid \exists b_n \sqsubseteq c(-n;m) \in a_- \cap \underline{0} \wedge D_{(-n;m)} \subseteq a_+ \cap \underline{0}\}$$
$$\{m \mid \exists b_n \sqsubseteq c(+n;m) \in a_- \cap \underline{0} \wedge D_{(+n;m)} \subseteq a_+ \cap \underline{0}\}$$

$$(a\bot)^{\underline{0}}_- = \{m \mid (-0;m) \in a_- \wedge D_{(-0;m)} \subseteq a_+\} \cap \underline{0}$$
$$= \{0\} \text{ if } 0 = (-0;0) \in a \text{ (since always } D_{(-0;m)} = \emptyset)$$
$$= \emptyset \text{ else.}$$

Now let $m \in a^0c_-$. Then for some n one has $(-n;m) \in a_- \cap \{0\}$, hence $0 = (-n;m) \in a_-$ and therefore $m \in (a\bot)^0$.
Thus $(a^0c)_- \subseteq (a\bot)^{\underline{0}}_-$. On the other hand, since never $(+n;m) = (n,2m+1) \in \{0\}$, one has $(a^0c)_+ = \emptyset$ and the statement follows.

(iv) By definition

$$(a^{n+1}c)_- = \{m \mid \exists b_p \sqsubseteq c(-p;m) \in a_- \cap \underline{n+1} \wedge D_{(-p;m)} \subseteq a_+ \cap \underline{n+1}\}$$
$$(ac^n)^{\underline{n}}_- = \{m \mid \exists b_p \sqsubseteq c^n(-p;m) \in a_- \wedge D_{(-p;m)} \subseteq a_+\} \cap \underline{n}.$$

Note that

$$(-p;m) \in a_- \cap \underline{n+1} \Rightarrow m < 2m \leqslant (p,2m) \leqslant n+1 \quad (\text{unless } m = 0)$$
$$\Rightarrow m \leqslant n \wedge p \leqslant n+1$$
$$\Rightarrow m \in \underline{n} \wedge |b_p| \subseteq \underline{n}, \text{ by 1.10(iii).}$$

Hence $(a^{n+1}c)_- \subseteq (ac^n)^{\underline{n}}_-$. Similarly for the RHS. □

2.3 LEMMA.

(i) $\bot^0 = \bot$

(ii) $\lambda x.\bot = \bot$

(iii) $\bot a = \bot$ for all $a \in T^\omega$.

Proof. *(i)*, *(ii)* By definition.

(iii) By definition one even has $\langle \emptyset, B\rangle\, a = \bot$ for all $B \in P\omega$ and $a \in T^\omega$. □

See [1] 7.2, 7.18 and 7.19 for the definition of $\lambda\,\underline{\Omega}$-terms, indexed $\Lambda\,\underline{\Omega}$-terms and reduction relations on these. Let $\Lambda\,\underline{\Omega}$ ($\Lambda\,\underline{\Omega}^{\mathbb{N}}$) be the class of (indexed) $\lambda\,\underline{\Omega}$-terms and let $\vec{\underline{\Omega}}_{\mathbb{N}}$ denote indexed reduction on this set. If $M \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$, then M^* is M without indices.

2.4 *Definition.*

(i) The map $[\![\ ]\!]_\rho : \Lambda \to T^\omega$ will be extended to $\underline{\Omega}^{\mathbb{N}}$ by adding

the clauses $[\![\underline{\Omega}]\!]_\rho = \bot$ and $[\![M^n]\!]_\rho = [\![M]\!]^n_\rho$.

(ii) For $M,N \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$ write $T^{\omega} \models M \sqsubseteq N$ iff for all $\rho:\mathrm{Var} \to T^{\omega}$ one has $[\![M]\!]_\rho \sqsubseteq [\![N]\!]_\rho$. Similarly for $T^{\omega} \models M = N$.

(iii) For $M \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$ define BT(M) to be BT(|M|), where |M| is obtained from M by omitting all indexes and replacing all constants $\underline{\Omega}$ by the λ-term $\Omega = (\lambda x.xx)(\lambda x.xx)$.

2.5 LEMMA. Let $M,N \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$ and $M \underset{\underline{\Omega}\mathbb{N}}{\rightarrow} N$. Then

(i) $BT(N) \subseteq BT(M)$

(ii) $T^{\omega} \models M \sqsubseteq N$

(Note the different order in (i) and (ii)).

Proof.

(i) In [1] it is proved that $BT(C[\underline{\Omega}]) \subseteq BT(C[Q])$ for all context C[ ] and $Q \in \Lambda\Omega$. Hence the approximation comes in at steps like $(\lambda x.P(x))^0\, Q \underset{\underline{\Omega}\mathbb{N}}{\rightarrow} P(\underline{\Omega}^0)$.

(ii) Immediate by 2.2(iii), (iv) and the definition of $\underset{\underline{\Omega}\mathbb{N}}{\rightarrow}$ reduction.

NOTATION. $A(M) = \{P \in \Lambda\,\underline{\Omega} \mid BT(P) \subseteq BT(M) \wedge P \text{ is an } \underline{\Omega}\text{-nf}\}$, for $M \in \Lambda$.

2.6 COROLLARY. Let $M \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$ be a completely indexed term (see [1] 7.18). Then there exists an $N \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$ such that

(i) $N^* \in A(M^*)$

(ii) $T^{\omega} \models M \sqsubseteq N$

Proof. By [1] 7.23 there exists an $N \in \Lambda\,\underline{\Omega}^{\mathbb{N}}$ in $\underline{\Omega}\,\mathbb{N}$-nf such that $M \underset{\underline{\Omega}\mathbb{N}}{\rightarrow} N$. Then (i) and (ii) follow from 2.5(i), (ii).

2.7 LEMMA. Let $N \in A(M)$. Then $T^{\omega} \models N \sqsubseteq M^3$.

Proof. By assumption $BT(N) \subseteq BT(M)$, N in $\underline{\Omega}$-nf.
Hence BT(N) is finite and BT(M) results from it by replacing some $\underline{\Omega}$'s by other trees.
Since $\underline{\Omega}$ is interpreted as $\bot$ and the operations in T^{ω} are monotonic, the result follows. □

2.8 APPROXIMATION THEOREM. Let $M \in \Lambda$. Then in T^ω

$$M = \bigsqcup\{P \mid P \in A(M)\}.$$

Proof. As in [1] 7.24 for D_∞ one has in T^ω

$$\begin{aligned} M &= \bigsqcup\{M^I \mid I \text{ indexing for } M\} \text{ by 2.2(i)} \\ &\sqsubseteq \bigsqcup\{L \mid L \in \Lambda\underline{\Omega}^{\mathbb{N}} \text{ and } L^* \in A(M)\} \text{ by 2.6} \\ &\sqsubseteq \bigsqcup\{P \mid P \in \Lambda\underline{\Omega} \text{ and } P \in A(M)\} \text{ since } L \subseteq L^* \\ &= \bigsqcup\{P \mid P \in A(M)\} \\ &\sqsubseteq M \text{ by 2.7.} \quad \square \end{aligned}$$

Let for $M \in \Lambda$ the term $M^k \in \Lambda\underline{\Omega}$ be defined as in [1] 7.4(iv). One has $M^k \in A(M)$.

2.9 COROLLARY. Let $M,N \in \Lambda$. Then

(i) $T^\omega \models M = \bigsqcup_k M^k$

(ii) $BT(M) \subseteq BT(N) \Rightarrow T^\omega \models M \sqsubseteq N$

(iii) $BT(M) = BT(N) \Rightarrow T^\omega \models M = N$.

Proof.

(i) As for [1] 7.8

(ii)
$$\begin{aligned} BT(M) \subseteq BT(N) &\Rightarrow \forall k\ BT(M^k) \subseteq BT(N) \\ &\Rightarrow \forall k\ T^\omega \models M^k \sqsubseteq N \text{ by 2.7} \\ &\Rightarrow T^\omega \models M \sqsubseteq N \text{ by (i)} \end{aligned}$$

(iii) Immediate by (ii). $\square$

2.10 COROLLARY.

(i) T^ω is sensible; in fact $M \in \Lambda$ is unsolvable $\Leftrightarrow T^\omega \models M = \bot$.

(ii) Let $Y \in \Lambda$ be the fixed point combinator. Then $[\![Y]\!]$ represents the least fixed point operator on T^ω.

Proof.

(i) As for [1] 7.9.

(ii) Let $Y_{\text{Curry}} = Y_C = [\![Y]\!] \in T^\omega$ and let $Y_{\text{Tarski}} = Y_T \in T^\omega$ be defined by $Y_T f = \bigsqcup_k f^k(\bot)$.

Now as in [1] 7.12.

$$\forall d \in T^\omega \qquad Y_C d = Y_T d.$$

Hence by weak extensionality $\lambda d\,.\,Y_C d = \lambda d\,.\,Y_T d$.

But then $Y_C = Y_T$, since both Y's satisfy $\lambda d \,.\, Yd = Y$. □

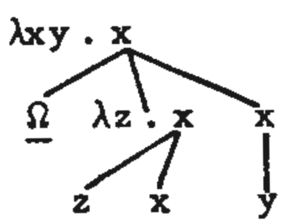
§3 THE CHARACTERIZATION THEOREM.

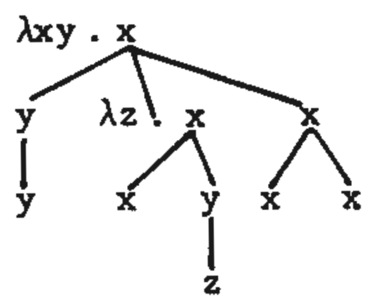
This section is devoted to proving the converse implications in 2.9. One lemma will be borrowed from the next section.

Remember that for $M \in \Lambda$ its Böhm tree $BT(M)$ is a function from sequence numbers to labels. If α is a node in $BT(M)$, then $BT(M)(\alpha)$ denotes the label (either of the form $\lambda\vec{x}.y$ or $\underline{\Omega}$) at node α. If α is not a node in $BT(M)$, then $BT(M)(\alpha) = *$ (representing undefined).

3.1 *Definition.* Let $M,N \in \Lambda$ and α be a sequence number. Then $M \sqsubseteq_{\alpha} N$ iff $BT(M)(\alpha) = *$ or if $BT(M)(\alpha) = \lambda\vec{x}.y$, then $[BT(M)(\alpha) = BT(N)(\alpha)$ and the node α has the same number of successors in $BT(M)$ as in $BT(N)]$.

EXAMPLE. Let M,N have respectively the following Böhm trees:

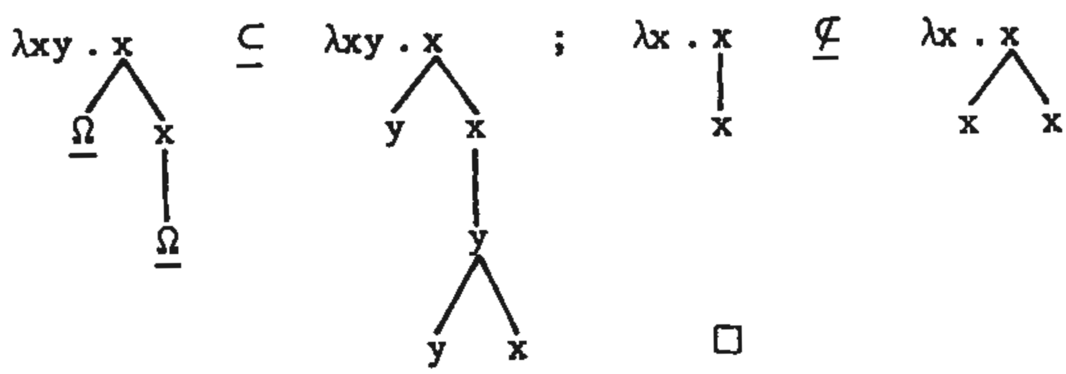
M: 

N: 

Then for $\alpha \in \{\langle\ \rangle,\langle 0\rangle,\langle 1\rangle,\langle 1,1,0\rangle\}$ one has $M \sqsubseteq_{\alpha} N$; for $\alpha \in \{\langle 2\rangle,\langle 1,1\rangle\}$ this is not so.

3.2 LEMMA. $BT(M) \subseteq BT(N) \Leftrightarrow \forall\alpha M \sqsubseteq_{\alpha} N$.

Proof. By definition of $\subseteq$ and $\sqsubseteq_{\alpha}$.

E.g. 

□

In the next lemma the combinators C_p needed for the technique in [1] §6, are changed appropriately. This idea was suggested by M. Hyland in a lecture (Utrecht, September 1977) on an improved version of his [4] dealing with the analysis of $P\omega$. In that case the definition of the new C_p^- is straightforward. A full §4 is needed to make a similar construction for T^ω.

3.3 C-LEMMA. Let $C_p = \lambda x_0 \ldots x_{p+1}.x_{p+1}x_0 \ldots x_p \in \Lambda^0$.
For each $p \in \mathbb{N}$ there is an element $C_p^- \in T^\omega$ such that:

(i) $T^\omega \models C_p^- x_0 \ldots x_{p+1} = C_p x_0 \ldots x_{p+1}$

(ii) For all $n \neq n'$, for all $m,m' < p$ and for all $\sigma_0,\ldots,\sigma_m$, $\tau_0,\ldots,\tau_{m'} \in \Lambda(T^\omega)$ one has
$T^\omega \not\models \lambda x_0 \ldots x_n.C_p^-\sigma_0 \ldots \sigma_m \sqsubseteq \lambda x_0 \ldots x_{n'}.C_p^-\tau_0 \ldots \tau_{m'}$.

Proof. See §4. □

3.4 *Definition.*

(i) A <u>substitutor</u> + is an operator $+ : \Lambda \to \Lambda(T^\omega)$ such that for some variables $y_0,\ldots,y_n$ and some $\sigma_0,\ldots,\sigma_n \in \Lambda(T^\omega)$ one has for all $P \in \Lambda$:

(1) $P^+ \equiv P[y_0 := \sigma_0] \ldots [y_n := \sigma_n]$.

(ii) Let α be a common node of BT(M) and BT(N). An <u>α-M-N substitutor</u> + is such that in (1) $\sigma_0 \equiv C_p^- y_0,\ldots,\sigma_n = C_p^- y_n$ for some p larger than the number of successor nodes of any node $\beta \leqslant \alpha$ in BT(M) and BT(N). [for $\alpha = \langle 1,1 \rangle$ in the example after 3.1 one has $p > 3$].

(iii) Write <u>$+ \in \alpha$-M-N</u> if + is an α-M-N substitutor.

3.5 LEMMA. Let $\sigma,\tau \in \Lambda(T^\omega)$ with $\sigma \equiv x\sigma_1 \ldots \sigma_n$, $\tau \equiv \tau t_1 \ldots \tau_m$

(i) if $x \not\equiv y$, then $T^\omega \not\models \sigma \sqsubseteq \tau$

(ii) if $x \equiv y$ and $m \neq n$, then $T^\omega \not\models \sigma \sqsubseteq \tau$.

Proof.

(i) Just take $a,b \in T^\omega$ such that $a \not\sqsubseteq b$ and a valuation such that $\rho(x) = \lambda x_1 \ldots x_n.a$ and $\rho(y) = \lambda x_1 \ldots x_m.b$.

(ii) Suppose $T^\omega \models \sigma \sqsubseteq \tau$ and write $\sigma z^{\sim k}$ for $\sigma\underbrace{z \ldots\ldots z}_{k\text{-times}}$.

Let, say, $m = n+k$. We have $T^\omega \models \sigma z^{\sim k} w \sqsubseteq \tau z^{\sim k} w$. Substitute in both sides for $x \equiv y$ the term $\lambda x_1 \ldots x_{m+1} . x_{m+1}$.
Then $T^\omega \models z^{\sim k} w \sqsubseteq w$, contradicting (i).

3.6 LEMMA. Let $M,N \in \Lambda$. Suppose $M \not\sqsubseteq_{\sim\alpha} N$ and $\forall\beta < \alpha \; M \sqsubseteq_{\sim\beta} N$. Then

$$\forall + \in \alpha\text{-}M\text{-}N \quad T^\omega \not\models M^+ \sqsubseteq N^+$$

Proof. Induction on the length k of α.
<u>k = 0.</u> Then $\alpha = \langle\ \rangle$. So we must show that $M \not\sqsubseteq_{\sim\langle\ \rangle} N$ implies $\forall + \in \langle\ \rangle\text{-}M\text{-}N \quad T^\omega \not\models M^+ \sqsubseteq N^+$.
By assumption $BT(M)(\langle\ \rangle) \neq *, \underline{\Omega}$ (otherwise $M \sqsubseteq_{\sim\langle\ \rangle} N$); so M is solvable with hnf, say

(1) $\quad M = \lambda x_1 \ldots x_n . y M_1 \ldots M_m$.

CASE 1. N is unsolvable. Then by 2.10(i)

(2) $\quad T^\omega \models N^+ = \bot$

for any substitutor +. Claim:

(3) $\quad \forall + \in \alpha\text{-}M\text{-}N \quad T^\omega \not\models M^+ = \bot$.

Indeed $T^\omega \models M^+ = \bot$, implies

$$T^\omega \models C_p^- y M_1^+ \ldots M_m^+ = \bot x_1 \ldots x_n = \bot$$

(apply both sides to $x_1 \ldots x_n$ and, if it was not done already, make the substitution $[y := C_p^- y]$).
Thus

$$T^\omega \models C_p^- y M_1^+ \ldots M_m^+ z_{m+1} \ldots z_{p+1} = \bot \text{ and}$$
$$T^\omega \models z_{p+1}\, y M_1^+ \ldots M_m^+ z_{m+1} \ldots z_p = \bot, \text{ by (i) in the C-lemma.}$$

Since z_{p+1} can be interpreted arbitrarily, this is a contradiction. This establishes (3) and hence by (2)

$$T^\omega \not\models M^+ \sqsubseteq N^+.$$

CASE 2. N is solvable, say with hnf.

(4) $\quad N = \lambda x_1 \ldots x_{n'} \cdot y' N_1 \ldots M_{m'}$.

Suppose that for some $+ \in \alpha\text{-}M\text{-}N$ one has

(5) $\quad T^\omega \models M^+ \sqsubseteq N^+$

in order to derive a contradiction. By the assumption that

$M \not\sqsubseteq_{\langle\ \rangle} N$ it follows from (1) and (4) that $y \not\equiv y' \vee n \neq n' \vee m \neq m'$.

CASE 2.1. $y \not\equiv y'$. Let, say, $n \geq n'$. Applying both sides of (5) to $x_1 \ldots x_n$ and making possibly (if it was not done yet) the substitution $[y := C_p^- y][y' := C_p^- y']$ one obtains

$$(6) \qquad T^\omega \models C_p^- y M_1^+ \ldots M_m^+ \sqsubseteq C_p^- y' N_1^+ \ldots N_{m'}^+ x_{n'+1} \ldots x_n.$$

By substituting I for $x_{n'+1}, \ldots, x_n$ and applying both sides of (6) to enough I's, it follows that

$$T^\omega \models C_p^- y M_1^{+1} \ldots M_m^{+1} I^{\sim k} \sqsubseteq C_p^- y' N_1^{+1} \ldots N_m^{+1} I^{\sim k'},$$

for some large k,k'. Since $p > m,m'$ it follows by (i) in the C-lemma that

$$T^\omega \models I y M_1^{+1} \ldots M_m^{+1} I^{\sim k-1} \sqsubseteq I y' N_1^{+1} \ldots N_{m'}^{+1} I^{\sim k'-1}$$

contradicting 3.5(i).

CASE 2.2. $y \equiv y'$ and $n \neq n'$. Let, say, $n > n'$. Applying both M^+ and N^+ to $x_1, \ldots, x_n$, (and possibly performing the substitution $[y := C_p^- y]$) one obtains

$$(7) \qquad T^\omega \models \lambda x_{n'+1} \ldots x_n . C_p^- y M_1^{+1} \ldots M_m^{+1} \sqsubseteq C_p^- y N_1^{+1} \ldots N_{m'}^{+1}.$$

This contradicts (ii) of the C-lemma.

CASE 2.3. $y \equiv y'$, $n = n'$ and $m \neq m'$. Then (5) implies

$$(8) \qquad T^\omega \models C_p^- y M_1^{+1} \ldots M_m^{+1} \sqsubseteq C_p^- y N_1^{+1} \ldots N_{m'}^{+1}.$$

By applying both sides to $x^{\sim q}$ for q large, it follows by (i) of the C-lemma that

$$T^\omega \models z y M_1^+ \ldots M_m^+ z^\sim \sqsubseteq z y N_1^+ \ldots N_{m'}^+ z^\sim.$$

This contradicts 3.5(ii).

The proof for $k = 0$ is now complete.

<u>$k > 0$.</u> Then $\alpha = \langle i \rangle * \beta$. By assumption one has (1) and (4) with $y = y'$, $n = n'$ and $m = m'$. By the induction hypothesis

$$(9) \qquad \forall +_1 \in \beta\text{-}M_i\text{-}N_i \quad T^\omega \not\models M_i^{+1} \sqsubseteq N_i^{+1}.$$

Now suppose for some $+ \in \alpha$-M-N one has $T^{\omega} \models M^{+} \sqsubseteq N^{+}$ in order to derive a contradiction. Then

$$T^{\omega} \models C_p^{-} y M_1^{+_1} \ldots M_m^{+_1} \sqsubseteq C_p^{-} y\ N_1^{+_1} \ldots N_m^{+_1}.$$

(First perform +, then (if not already done) substitute $[y := C_p^{-} y]$ thus obtaining $+_1$).
Now applying both sides to enough copies of the term $\lambda y_0 \ldots y_p . y_i$ we have

$$T^{\omega} \models M_i^{+_1} \sqsubseteq N_i^{+}.$$

Since clearly $+_1 \in \beta$-M_i-N_i, this contradicts (9). □

3.7 COROLLARY. Let $M,N \in \Lambda$. Then

$$BT(M) \not\sqsubseteq BT(N) \Rightarrow T^{\omega} \not\models M \sqsubseteq N.$$

Proof. By assumption and 3.2, for some α one has $M \not\sqsubseteq_{\alpha} N$. Take α of minimal length. Then 3.6 applies and by taking the empty substitutor it follows that $T^{\omega} \not\models M \sqsubseteq N$. □

3.8 THEOREM. Let $M,N \in \Lambda$. Then

(i) $T^{\omega} \models M \sqsubseteq N \Leftrightarrow BT(M) \sqsubseteq BT(N)$

(ii) $T^{\omega} \models M = N \Longleftrightarrow BT(M) = BT(N)$

Proof.

(i) By 2.9(ii) and 3.7

(ii) By (i). □

Let $a,b \in T^{\omega}$. Define $a \uparrow b$ iff $\exists c \in T^{\omega}\ a,b \sqsubseteq c$.
Using the methods of this section and [2], thm. 10.1.23 one can show the following.

3.9 THEOREM. Let $M,N \in \Lambda$. Then $T^{\omega} \models M \uparrow N \Leftrightarrow \exists L \in \Lambda$ $BT(M), BT(N) \subseteq BT(L)$.

§4. MAIN LEMMA

In order to prove the C-lemma (stated in 3.3, proved in 4.33), for each $p \in \mathbb{N}$ we build up a $C_p^- \in \Lambda(T^\omega)$ (in 4.17 and 4.31) which "behaves" like C_p, when it is given at least p+2 arguments (see 4.32 (1)), but which is essentially not comparable with C_p (see 4.21, 4.23, 4.32 (2)).

This section contains two parts: the first is devoted to "minimize" the left side (i.e. the $\ldots_-$ side) of graphs of continuous functions over T^ω in a way that they preserve their functional behaviour. The second is devoted to modify the right side of the elements of T^ω so obtained, also preserving their functional behaviour.

PART I

First a slightly deeper insight into the structure of T^ω is required.

4.0 NOTATION. $(\pm n_1;\pm n_2;\ldots;\pm n_k;m) = (\pm n_1;(\pm n_k;m)\ldots)))$.
E.g; $(+n_1;-n_2;-n_3;m) = (+n_1;(-n_2;(-n_3;m)))$.

4.1 LEMMA. Let k>0 and let $f\in[(T^\omega)^k \to T^\omega]$. Then

$$\forall \vec{a}^k \in T^\omega\ \lambda x_1\ldots x_k.f(x_1,\ldots,x_k)\vec{a}^k_- = \{m \mid \exists b_{n_k} \sqsubseteq a_k,\ldots,\exists b_{n_1} \sqsubseteq a_1\ (-n_1;\ldots;-n_k;m) \in \lambda x_1\ldots x_k.f(x_1,\ldots,x_k)_-\}.$$

Proof. Induction on k.

<u>k = 1</u>. By lemma 1.6.

<u>k > 1</u>. $((\lambda x_1\ldots x_{k-1}.(\lambda x_k.f(x_1,\ldots,x_k)))\vec{a}^{k-1})a_{k-} =$

$$= \{m \mid \exists b_{n_k} \sqsubseteq a_k\ (-n_k;m) \in \{p \mid \exists b_{n_{k-1}} \sqsubseteq a_{k-1}\ldots \exists b_{n_1} \sqsubseteq a_1$$
$$(-n_1;\ldots;-n_{k-1};p) \in \lambda x_1\ldots x_k.f(x_1,\ldots,x_k)_-\}$$
$$\wedge D_{(-n_k;m)} \subseteq \lambda x_k.f(a_1,\ldots,a_{k-1},x_k)_+\},$$

by the induction hypothesis.

Similarly to 1.6, it has to be shown that the second condition is always satisfied when the first is.

Let $g(x_k) = f(b_{n_1},\dots,b_{n_{k-1}},x_k)$. Then

$$(-n_k;m) \in \lambda x_k.g(x_k)_- \wedge (+n';m) \in D_{(-n_k;m)} \Rightarrow$$

$$m \in g(b_{n_k})_- \wedge n'\uparrow n_k \Rightarrow$$

$$m \in g(b_{n'} \sqcup b_{n_k})_-$$

since g is monotonic.

Since f is monotonic it follows that

$$\exists b_q = b_{n'} \sqcup b_{n_k} \sqsupseteq b_{n'} \quad m \in f(a_1,\dots,a_{k-1},b_q)_-$$

and hence

$$D_{(-n_k;m)} \subseteq \lambda x_k . f(a_1,\dots,a_{k-1},x_k)_+ . \boxtimes$$

4.2 LEMMA. Let $k>0$ and $b \in T^{\omega}$, then

$$\forall \vec{a}^k \in T^{\omega}\ b a^k_- = \{m \mid \exists b_{n_k} \sqsubseteq a_k,\dots,\exists b_{n_1} \sqsubseteq a_1$$
$$(-n_1;\dots;-n_k;m) \in b_-$$
$$\wedge D_{(-n_1;\dots;-n_k;m)} \subseteq b_+$$
$$\wedge D_{(-n_2;\dots;-n_k;m)} \subseteq b a_{1+}$$
$$\vdots$$
$$\wedge D_{(-n_k;m)} \subseteq b a_1 \dots a_{k-1+}\}.$$

Proof. Easy: just apply several times definition 1.3 . $\boxtimes$

4.3 DEFINITION

(i) Let $k > 0$. S_k is the set of the 2^k maps (signs) from $\{1,\dots,k\}$ into $\{-,+\}$. If $s \in S_k$, then $(n_1;\dots;n_k)_s$ stands for $(s(1)n_1;\dots;s(k)n_k;m)$.

(ii) If A is a predicate of k+1 arguments over $\mathbb{N}$, then $A_s = \{(n_1;\dots;n_k;m)_s \mid A(n_1,\dots,n_k,m)\}$.

4.4 LEMMA. Let $k > 0$ and $f \in [(T^{\omega})^k \to T^{\omega}]$. Then there exist predicates ${}^sA \subseteq \mathbb{N}^{k+1}$ such that for the corresponding ${}^sA_s \subseteq \mathbb{N}$

1) $\forall s,s' \in S_k \quad s \neq s' \Rightarrow {}^sA_s \cap {}^{s'}A_{s'} = \emptyset$

2) $\lambda x_1 \ldots x_k . f(x_1,\ldots,x_k)_- = \bigcup_{s \in S_k} {}^sA_s$

Proof. Obvious: just recall definition 1.3 (i) and observe that different signs make sets disjoint.

$$(\text{E.g. } (\lambda x_0x_1x_3.x_3x_0x_1)_- = \{(-n_1;-n_2;\pm n_3;m) \mid m \in b_{n_3}b_{n_1}b_{n_2\pm}\}$$

$$\cup \{(-n_1;+n_2;\pm n_3;m) \mid \exists b_{n_3'} \sqsupseteq b_{n_3} \quad m \in b_{n_3'}b_{n_1}b_{n_2\mp}\}$$

$$\cup \{(+n_1;+n_2;\pm n_3;m) \mid \exists b_{n_2'} \sqsupseteq b_{n_2} \; m \in b_{n_3}b_{n_1}b_{n_2'\pm}\}$$

$$\cup \{(+n_1;-n_2;\pm n_3;m) \mid \exists b_{n_2'} \sqsupseteq b_{n_2} \quad \exists b_{n_3'} \sqsupseteq b_{n_3}$$

$$m \in b_{n_3'}b_{n_1}b_{n_2'\mp}\}$$

and each set of this disjoint union divides into two further disjoint sets, because of the signs of n_3). ◻

4.5 DEFINITION. Let $k > 0$, $s \in S_k$ and $1 \le j \le k$. In a $k+1$-tuple $(n_1;\ldots,n_k;m)_s$, we say n_j is <u>increasable</u> iff $\neg \; \exists \; i < j \; s(i)=+$.

4.6 LEMMA. Let $k > 0$, $f \in [(T^{\omega})^k \to T^{\omega}]$ and $s \in S_k$. Then, if $(n_1;\ldots;n_k;m)_s \in \lambda x_1 \ldots x_k . f(x_1,\ldots,x_k)_-$ and n_j is increasable one has

$$\forall b_{n_j'} \sqsupseteq b_{n_j} \quad (n_1;\ldots;n_j';\ldots n_k;m)_s \in \lambda x_1 \ldots x_k . f(x_1,\ldots,x_k)_- .$$

Proof. Easy: by definition 1.3 (i) and a simple inductive argument. ◻

The aim of next lemmas is to construct, for any k and any continuous f, an element of T^{ω} which "behaves" similarly to $\lambda x_1 \ldots x_k . f(x_1,\ldots,x_k)$ (see 4.12) but which has an essentially different left part. This will be done by a "minimizing technique".

4.7 DEFINITION. Let $\langle X, \lhd\rangle$ be a poset and let A be a 3-ary predicate on X.

(i) We say x_1, x_3 are <u>minimally such that</u> $A(x_1,x_2,x_3)$, notation x_1,x_3 mins $A(x_1,x_2,x_3)$, iff

<u>1</u>. $A(x_1,x_2,x_3)$ holds.
<u>2</u>. If $x_1' \le x_1$ and $x_3' \le x_3$ and $\langle x_1,x_3\rangle \ne \langle x_1',x_3'\rangle$, then $\neg A(x_1',x_2,x_3')$.

(ii) If A is a k-ary predicate on X and $i_1,\ldots,i_m \in \{1,\ldots,k\}$ are arbitrary, then it is clear how to define analogously $x_{i_1},\ldots,x_{i_m}$ is mins $A(x_1,\ldots,x_k)$.

4.8 DEFINITION.

(i) Let A be a k+1-ary predicate on $\mathbb{N}$. We partially order $\mathbb{N}$ by $n \lhd m$ iff $b_n \sqsubseteq b_m$. Remember

$$A_s = \{(n_1;\ldots;n_k;m)_s \mid A(n_1,\ldots,n_k,m)\}.$$

Define

$$\tilde{A_s} = \{(n_1;\ldots;n_k;m)_s \mid \text{ the increasable } n_j\text{'s are mins } A(n_1,\ldots,n_k,m)\}.$$

(ii) Let $f \in [(T^\omega)^k \to T^\omega]$. By 4.4 let ${}^sA \subseteq \mathbb{N}^{k+1}$ be such that $\lambda x_1\ldots x_k.f(x_1,\ldots,x_k)_- = \bigcup_{s\in S_k} {}^sA_s$.

Define

$$\tilde{\lambda} x_1\ldots x_k.f(x_1,\ldots,x_k) = \langle \bigcup_{s\in S_k} {}^s\tilde{A_s}, \lambda x_1\ldots x_k.f(x_1,\ldots,x_k)_+\rangle.$$

(E.g.: $\tilde{\lambda} x_0x_1x_2.x_2x_0x_{1-} =$

$\{(-n_1;-n_2;\pm n_3;m) \mid n_1,n_2,n_3 \text{ mins } m\in (b_{n_3}b_{n_1}b_{n_2})_{\pm}\}$

$\cup\ \{(-n_1;+n_2;\pm n_3;m) \mid n_1,n_2$

mins $\exists b_{n'_3} \sqsupseteq b_{n_3}\ m \in (b_{n'_3}b_{n_1}b_{n_2})_{\mp}\}$

$\cup\ \{(+n_1;+n_2;\pm n_3;m)\,|\,n_1$

mins $\exists\, b_{n_2'} \sqsupseteq b_{n_2}\ \ m\ \in (b_{n_3} b_{n_1} b_{n_2'})_{\pm}\}$

$\cup\ \{(+n_1;-n_2;\pm n_3;m)\,|\,n_1$

mins $\exists\, b_{n_2'} \sqsupseteq b_{n_2}\ \ \exists\, b_{n_3'} \sqsupseteq b_{n_3}$

$m\ \in (b_{n_3'} b_{n_1} b_{n_2'})_{\mp}\}$,

i.e. we take n_1,n_2,n_3 so that b_{n_1},b_{n_2},b_{n_3} are minimal such that $m \in (b_{n_3} b_{n_1} b_{n_2})_+$ holds etc.)

4.9 LEMMA. Let $k > 0$ and $f \in [(T^\omega)^k \to T^\omega]$. Then

(i) $\tilde{\lambda} x_1 \ldots x_k . f(x_1,\ldots,x_k) \sqsubseteq \lambda x_1,\ldots,x_k . f(x_1,\ldots,x_k)$

(ii) the inclusion in (i) is strict iff $\exists\, \vec{a}^k \in T^\omega$ $f(a_1,\ldots,a_k) \neq \bot$ iff $\tilde{\lambda} x_1 \ldots x_k . f(x_1,\ldots,x_k)_- \neq \phi$.

Proof. (i) By definition.

(ii) First equivalence $\Rightarrow$: if $\forall \vec{a}^k \in T^\omega\ f(a_1,\ldots,a_k) = \bot$, then by definition, $\lambda x_1 \ldots x_k . f(x_1,\ldots,x_k) = \bot$.

$\Leftarrow$: let $a_1 \in T^\omega$ be such that $(\lambda x_1 \ldots x_k . f(x_1,\ldots,x_k))a_1 \neq \bot$, then $\exists\, m\, \exists\, b_{n_1} \sqsubseteq a_1$ such that $(\pm n_1;m) \in \lambda x_1 \ldots x_k . f(x_1,\ldots,x_k)_-$, by the proof of 2.3 (iii).

Now n_1 is increasable and hence by 4.6

$$\forall b_p \not\sqsupseteq b_{n_1} (\pm p;m) \in \lambda x_1 \ldots x_k . f(x_1,\ldots,x_k)_-$$

but $(\pm p;m) \notin \tilde{\lambda} x_1 \ldots x_k . f(x_1,\ldots,x_k)_-$ by definition.

The second equivalence follows by a similar argument. ¤

4.10 LEMMA. Let $k > 0$ and $f \in [(T^\omega)^k \to T^\omega]$. Then $\forall i \le k$ $\forall \vec{a}^i \in T^\omega\ (\tilde{\lambda} x_1 \ldots x_k . f(x_1,\ldots,x_k))\vec{a}^i_+ = (\lambda x_1 \ldots x_k . f(x_1,\ldots x_k))\vec{a}^i_+$.

Proof. Induction on i.

i = 0. By definition.

i > 0. Assume the thesis for any $j \leq i-1$. Then one has

$$((\lambda^{\sim} x_1 \ldots x_k . f(x_1,\ldots,x_k))\vec{a}^{i-1})a_{i+} =$$

$$= \{m | \exists b_{n_i} \sqsubseteq a_i \ (+n_i;m) \in (\lambda^{\sim} x_1 \ldots x_k . f(x_1,\ldots,x_k))\vec{a}_-^{i-1} \wedge$$
$$D_{(+n_i;m)} \subseteq (\lambda x_1 \ldots x_k . f(x_1,\ldots,x_k))\vec{a}_+^{i-1}\},$$

by definition and the induction hypothesis,

$$= \{m | \exists b_{n_i} \sqsubseteq a_i \ldots \exists b_{n_1} \sqsubseteq a_1$$
$$(-n_1;\ldots;-n_{i-1};+n_i;m) \in \lambda^{\sim} x_1 .. x_k . f(x_1,\ldots,x_k)_-$$
$$\wedge D_{(+n_i;m)} \subseteq (\lambda x_1 .. x_k . f(x_1,\ldots,x_k))\vec{a}_+^{i-1}$$
$$\wedge D_{(-n_{i-1};+n_i;m)} \subseteq (\lambda x_1 .. x_k . f(x_1,\ldots x_k))\vec{a}_+^{i-2}$$
$$\vdots$$
$$\wedge D_{(-n_1;..;-n_{i-1};+n_i;m)} \subseteq \lambda x_1 .. x_k . f(x_1,..,x_k)_+\},$$

by 4.2 and the induction hypothesis,

$$= \{m | \exists b_{n_i} \sqsubseteq a_i \ldots \exists b_{n_1} \sqsubseteq a_1 \ \ n_1,..,n_i \text{ mins } (-n_1;..;+n_i;m) \in$$
$$\lambda x_1 .. x_k . f(x_1,\ldots,x_k)_-$$
$$\wedge D_{(+n_i;m)} \subseteq \ldots$$
$$\vdots$$
$$\wedge D_{(-n_1;\ldots;+n_i;m)} \subseteq \ldots\},$$

by 4.8,

$= (\lambda x_1 \ldots x_k . f(x_1,\ldots,x_k))\vec{a}_+^{i}$, by 4.2 and the fact that the finite $b_{n_j} \sqsubseteq a_j$ can be chosen minimally. ⌑

4.11 DEFINITION.

(i) $A \subseteq \mathbb{N}$ is <u>meager</u> iff for all $m,n \in \mathbb{N}$

$(+n;m) \in A \Rightarrow \exists b_{n'} \sqsupseteq b_n (+n';m) \notin A.$

$(-n;m) \in A \Rightarrow \exists b_{n'} \sqsupseteq b_n (-n';m) \notin A$

(ii) $a \in T^\omega$ is <u>meager</u> iff a_- is so.

Let $p \in \mathbb{N}$ and $a,a' \in T^\omega$.

(iii) a is <u>p-meager</u> iff

$\forall_i \le p \quad \forall \vec{c}^{\,i} \in T^\omega \quad a\vec{c}^{\,i}$ is meager.

(iv) $a \underset{p}{\sim} a'$ iff $\forall \vec{c}^{\,p+1} \in T^\omega \quad a\vec{c}^{\,p+1} = a'\vec{c}^{\,p+1}$

4.12 LEMMA. Let $p \in \mathbb{N}$ and $f \in [(T^\omega)^{p+1} \to T^\omega]$. Then

(i) $\tilde{\lambda} x_0 \ldots x_p . f(x_0,\ldots,x_p) \sim_p \lambda x_0 \ldots x_p . f(x_0,\ldots,x_p)$

(ii) $\tilde{\lambda} x_0 \ldots x_p . f(x_0,\ldots,x_p)$ is p-meager.

Proof. (i) Concerning the right part, the result follows from 4.10, for k = p+1. As to the left part,

$$(\tilde{\lambda} x_0 \ldots x_p . f(x_0,\ldots,x_p))\vec{a}_-^{\,p+1} =$$

$$= \{m \mid \exists b_{n_{p+1}} \sqsubseteq a_{p+1} \ldots \exists b_{n_1} \sqsubseteq a_1$$

$$(-n_1;\ldots;-n_{p+1};m) \in \tilde{\lambda}\, x_0 \ldots x_p . f(x_0,\ldots,x_p)_-$$

$$\wedge\ D_{(-n_{p+1};m)} \subseteq (\lambda x_0 \ldots x_p . f(x_0,\ldots,x_p))\vec{a}_+^{\,p}$$

$$\vdots$$

$$\wedge\ D_{(-n_1;\ldots;-n_{p+1};m)} \subseteq \lambda x_0 \ldots x_p . f(x_0,\ldots,x_p)_+\},$$

by 4.2 and 4.10,

$= f(a_1,\ldots,a_{p+1})_-$, as in the proof of 4.10.

(ii) Let $i \le p$. If $(\pm n_{i+1};m) \in (\tilde{\lambda} x_0 \ldots x_p . f(x_0,\ldots,x_p))\vec{a}_-^{\,i}$, then by 4.2. $\exists b_{n_i} \sqsubseteq a_i \ldots \exists b_{n_1} \sqsubseteq a_1 (-n_1;\ldots;-n_i;\pm n_{i+1};m) \in$

$$\tilde{\lambda} x_0 \ldots x_p . f(x_0,\ldots,x_p)_-.$$

Thus $\pm n_{i+1}$ is still increasable and, by 4.8, for no

$b_p \not\sqsupseteq b_{n_{i+1}}$ $(\pm p;m) \in (\lambda^{\sim} x_0 \ldots x_p . f(x_0,\ldots,x_p))\vec{a}^i_-$. ¤

The lemmas proved so far will now be applied to some particular continuous functions.

4.13 LEMMA. Let $k > 0$ and $m \in \mathbb{N}$. Then

$$\exists n_1 \in \mathbb{N} \quad \forall \vec{a}^k \in T^\omega \quad (-n_1;m) \in \lambda z.z\vec{a}^k_-.$$

Proof. Just take n_1 such that $b_{n_1} \sqsupseteq \{(-0;\ldots;-0;m)\}$, with k times-0. Then by 4.2, one has $m \in b_{n_1}\vec{\bot}^k_-$, since $b_0 = \bot$ and $\forall h \ \ D_{(-0;h)} = \emptyset$.
Thus $(-n_1;m) \in \lambda z.z\vec{\bot}^k_- \subseteq \lambda z.z\vec{a}^k_-$ for any $\vec{a}^k \in T^\omega$. ¤

4.14 DEFINITION. Let $k > 0$ and $(-n_1;m) \in \lambda z.z\vec{\bot}^k_-$; since n_1 is increasable by 4.6 one has $\forall b_n \sqsupseteq b_{n_1}$ $(-n;m) \in \lambda z.z\vec{\bot}^k_-$.
Take some $b_n \not\sqsupseteq b_{n_1}$. Then for all $\vec{a}^k \in T^\omega$ define

$$\lambda^\circ z.z\vec{a}^k = \langle \lambda z.z\vec{a}^k_- \backslash \{(-n;m)\}, \lambda z.z\vec{a}^k_+ \rangle.$$

Note that $\lambda^\circ z.z\vec{a}^k$ depends on the selected n; however this "stealing process" does not modify the functional behaviour of $\lambda z.z\vec{a}^k$, as next lemma shows.

4.15 LEMMA. Let $k > 0$.

(i) $\forall \vec{a}^{k+1} \in T^\omega \quad (\lambda^\circ z.z\vec{a}^k)a_{k+1} = a_{k+1}\vec{a}^k$

(ii) If $f(a_1,\ldots,a_k) = \lambda^\circ z.z\vec{a}^k$, then $f \in [(T^\omega)^k \to T^\omega]$.

Proof. (i) Obvious.

(ii) Let $\{(a^i_1,\ldots,a^i_k)\}_{i\in I} \subseteq (T^\omega)^k$ be a directed set.

Then also $\{a^i_j\}_{i\in I} \subseteq T^\omega$ is directed and $f(\bigsqcup_i (a^i_1;\ldots,a^i_k)) =$

$= \langle \lambda z.z(\bigsqcup_i a^i_1)\ldots(\bigsqcup_j a^j_k)_- \backslash \{(-n;m)\}, \lambda z.z(\bigsqcup_i a^i_1)\ldots(\bigsqcup_j a^j_k)_+ \rangle$,

$= \langle \bigsqcup_i (\lambda z.za^i_1\ldots a^i_k)_- \backslash \{(-n;m)\}, \bigsqcup_j (\lambda z.za^j_1\ldots a^j_k)_+ \rangle$, ⊠

by continuity of $g(a_1,\dots,a_k) = \lambda z.z\vec{a}^k$ and completeness of T^ω,
$= \bigsqcup_i f(a_1^i,\dots,a_k^i)$. ☒

4.16 DEFINITION. Remember $C_p = \lambda x_0\dots x_{p+1}.x_{p+1}x_0\dots x_p$. Define

$$C_p^\sim = \lambda^\sim x_0\dots x_p.(\lambda^\circ x_{p+1}.x_{p+1}x_0\dots x_p).$$

4.17 THEOREM. Let $p \in \mathbb{N}$. Then in T^ω

(i) $C_p^\sim$ is p-meager.

(ii) $C_p^\sim \sqsubseteq \lambda x_0\dots x_p.C_p^\sim x_0\dots x_p$.

(iii) $\exists m_0 \in \mathbb{N} \quad \forall \vec{a}^{p+1} \in T^\omega \quad m_0 \notin |C_p^\sim \vec{a}^{p+1}|$.

(iv) $C_p^\sim \sim_{p+1} C_p$.

Proof. (i) By 4.12 (ii).

(ii)
$$\begin{aligned} C_p^\sim &= \lambda^\sim x_0\dots x_p.(\lambda^\circ x_{p+1}.x_{p+1}x_0\dots x_p) \\ &\sqsubseteq \lambda x_0\dots x_p.(\lambda^\circ x_{p+1}.x_{p+1}x_0\dots x_p) && \text{by 4.9 (i).} \\ &= \lambda x_0\dots x_p.C_p^\sim x_0\dots x_p && \text{by 4.12 (i).} \end{aligned}$$

(iii) Let $n,m \in \mathbb{N}$ be such that for $m_0 = (\neg n;m)$ one has

$$\forall \vec{a}^{p+1} \in T^\omega \; m_0 \in \lambda z.z\vec{a}^{p+1} \wedge m_0 \notin \lambda^\circ z.z\vec{a}^{p+1}_-, \text{ by 4.14.}$$

Thus $m_0 \notin \lambda^\circ z.z\vec{a}^{p+1}_- \cup \lambda^\circ z.z\vec{a}^{p+1}_+$ (note $\lambda^\circ z.z\vec{a}^{p+1}_- \cap \lambda^\circ z.z\vec{a}^{p+1}_+ = \emptyset$) and 4.12 gives the result.

(iv) By 4.12 (i) and 4.15 (i), where $k = p+1$. ☒

PART II

The main notion introduced in this part is 'lopsided element' (4.20). Lopsided elements of T^{ω} are not comparable (4.21,4.23) with graphs of continuous functions. Finally 4.31 and 4.17 give the C-lemma.

4.18 DEFINITION.

Let $m,p,q \in \mathbb{N}$. Define

(i) $K^p m = \{(\pm n_0;\ldots;\pm n_p;m) \mid n_0,\ldots,n_p \in \mathbb{N}\}$.

(ii) $K^{\circ}_q m = \{(\pm n_0;m) \mid b_q \sqsubseteq b_{n_0}\}$.

(iii) Each number q may be written in a unique way as $q = (\pm n_0;\ldots;\pm n_p;q_p)$. Define $[q]_p = n_p$.

$K^{\circ}m$ is called the <u>cone</u> over m; $K^p m$ the p^{th} <u>hypercone</u> over m; $K^{\circ}_q m$ a <u>subcone</u> over m.

4.19 DEFINITION.

Let $A \subseteq \mathbb{N}$:

(i) A is <u>saturated</u> iff for all n,m

$(+\ n;m) \in A \Rightarrow \forall b_{n'} \sqsupseteq b_n\ (+\ n';m) \in A.$

$(-\ n;m) \in A \Rightarrow \forall b_{n'} \sqsupseteq b_n\ (-\ n';m) \in A.$

(ii) A is <u>fat</u> iff $\exists\ q,m \in \mathbb{N}\ K^{\circ}_q m \subseteq A$.

4.20 DEFINITION.

Let $a \in T^{\omega}$

(i) a is <u>saturated</u> (<u>fat</u>) iff $a_-(a_+)$ is so.

(ii) a is <u>lopsided</u> iff a_- is meager (see 4.11) and a_+ is fat.

4.21 DEFINITION.

Let $\sigma,\tau \in \Lambda(T^{\omega})$:

$\sigma * \tau$ iff $T^{\omega} = \sigma \not\sqsubseteq \tau \wedge \tau \not\sqsubseteq \sigma$.

4.22 LEMMA.

Let $f \in [T^\omega \to T^\omega]$. Then $\lambda x.f(x)$ is saturated and not fat.

Proof. As to saturatedness, suppose $(\mp n;m) \in (\lambda x.f(x))_-$.
Then $m \in f(b_n)_{\mp}$ and by monotonicity of f for all $b_{n'} \sqsupseteq b_n$
$m \in f(b_{n'})_{\mp}$.
Hence $(\mp n';m) \in (\lambda x.f(x))_-$.
To show that $(\lambda x.f(x))_+$ is not fat, suppose that for r,m
$K_r^\circ m \subseteq (\lambda x.f(x))_+$.
Then since $(-r;m) \in K_r^\circ m$, $(-r;m) \in (\lambda x.f(x))_+$, i.e. for some
$b_q \sqsupseteq b_r$,
(1) $m \in f(b_q)_+$.
Also $(+q;m) \in K_r^\circ m$; thus $(+q;m) \in (\lambda x.f(x))_+$, i.e. for some
$b_{n'} \sqsupseteq b_q$,
(2) $m \in f(b_{n'})_-$.
But by (1), (2) and monotonicity of $f, m \in (f(b_{n'}))_- \cap (f(b_{n'}))_+$,
a contradiction. ⊠

4.23 COROLLARY.
Let $a \in T^\omega$ be lopsided and let $f \in [T^\omega \to T^\omega]$ be such that
$\exists c\ f(c) \neq \bot$. Then $a * \lambda x.f(x)$.

Proof. Suppose $a \sqsubseteq x.f(x)$. Then $\exists\ r,m\ K_r^\circ m \subseteq a_+ \subseteq (\lambda x.f(x))_+$;
hence $\lambda x.f(x)$ is fat, contradicting 4.22.
Suppose $\lambda x.f(x) \sqsubseteq a$. Then $(\lambda x.f(x))_- \subseteq a_-$.
By assumption and 2.3.(iii) one has $(\lambda x.f(x))_- \neq \emptyset$. Let
$(\mp n;m) \in (\lambda x.f(x))_- \subseteq a_-$; then since $\lambda x.f(x)$ is saturated one
has $\forall b_{n'} \sqsupseteq b_n\ (\mp n';m) \in (\lambda x.f(x))_- \subseteq a_-$, contradicting the
meagerness of a. ⊠

4.24 DEFINITION.

(i) Let $a, b \in T^\omega$; define
$a \uparrow b$ iff $\exists\ c \in T^\omega\ a, b \sqsubseteq c$ (i.e. iff $\langle a_- \cup b_-$,

$a_+ \cup b_+\rangle \in T^{\omega}$, i.e. iff $a \sqcup b$ exists).

(ii) Let $n \in \mathbb{N}$; write

$(-n;A) = \{(-n;m)/m \in A\}$. Similarly for $(+n;A)$.

(iii) Let $a \in T^{\omega}$; define

$\underline{a} = \langle(-0;a_-) \cup (+0;a_+),\ (-0;a_+) \cup (-1;a_+)\rangle$

Clearly $\underline{a} \in T^{\omega}$.

4.25 LEMMA

Suppose $a\uparrow\underline{b}$. Then for all c one has $(ac)\uparrow b$ and $(a \sqcup \underline{b})c =$ $= (ac) \sqcup b$.

Proof. First we show that for all c

(1) $(a \sqcup \underline{b})\underline{c} = (ac) \sqcup b_-$.

By definition

$(a \sqcup \underline{b})c_- = \{m \mid \exists b_n \sqsubseteq c\ (-n;m) \in a_- \cup \underline{b}_- \wedge\ D_{(-n;m)} \subseteq a_+ \cup \underline{b}_+\}$

$(ac) \sqcup b_- = \{m \mid \exists b_n \sqsubseteq c\ (-n;m) \in a_- \wedge D_{(-n;m)} \subseteq a_+\} \cup b_-$.

Suppose that $m \in (a \sqcup \underline{b})c_-$ in order to show that $m \in (ac) \sqcup b_-$. Then for some $b_n \sqsubseteq c$, $(-n;m) \in a_- \cup \underline{b}_-$ and $D_{(-n;m)} \subseteq a_+ \cup \underline{b}_+$. If $m \in b_-$, this part of the proof is complete; so suppose that $m \notin b_-$. Then $(-n;m) \notin \underline{b}_-$ and hence $(-n;m) \in a_-$. Similarly $D_{(-n;m)} \subseteq a_+$, since for no n' one has $(+n';m) \in \underline{b}_+$. Therefore $m \in (ac) \sqcup b_-$.

Suppose now that $m \in (ac) \sqcup b_-$. If $m \in ac_-$, then trivally $m \in (a \sqcup \underline{b})c_-$. If $m \in b_-$, then $(-0;m) \in \underline{b}_- \subseteq a_- \cup \underline{b}_-$ and $D_{(-0;m)} = \emptyset \subseteq a_+ \cup \underline{b}_+$. Hence again $m \in (a \sqcup \underline{b})c_-$.

Similarly one can show that

$(a \sqcup \underline{b})c_+ = (ac) \sqcup b_+$ using $D_{(+0;m)} = \{(-0;m),(-1;m)\}$.

Since $a \sqcup \underline{b}$, $c \in T^{\omega}$ also $(a \sqcup \underline{b})c \in T^{\omega}$, i.e. $(ac) \sqcup b$ exists. ⊠

4.26 LEMMA.

Let $q_i = 3^i$. Then $b_{q_i} = \langle \emptyset, \{i\} \rangle$ and $\forall i \neq j\ b_{q_i} \not\sqsubseteq b_{q_j}$

Proof. ⊠

4.27 LEMMA.

Suppose that for some $m_0, p \in \mathbb{N}$ and $a \in T^\omega$ one has $K^p m_0 \cap a_- = \emptyset$ and $A \subseteq K^p m_0$. Then $\forall b \in T^\omega$ $(a \sqcup \langle \emptyset, A\rangle)b = ab$.

Proof. First we show $(a \sqcup \langle \emptyset, A\rangle)b_- = ab_-$.

Let $m \in (a \sqcup \langle \emptyset, A\rangle)b_- = \{m \mid \exists b_n \sqsubseteq b\ (-n;m) \in a_-$

$$\wedge D_{(-n;m)} \subseteq a_+ \cup A\}.$$

For some n such that $(-n;m) \in a_- \wedge D_{(-n;m)} \subseteq a_+ \cap A$, write $(-n;m) = (-n_0;\underline{+}n_1;\ldots;\underline{+}n_p;m')$: by assumption $m' \neq m_0$. Therefore $D_{(-n;m)} \cap A = \emptyset$, hence $D_{(-n;m)} \subseteq a_+$ and it follows that $m \in ab_- = \{m \mid \exists b_n \sqsubseteq b\ (-n;m) \in a_- \wedge D_{(-n;m)} \subseteq a_+\}$.

On the other hand $ab_- \subseteq (a \sqcup \langle \emptyset, A\rangle)b_-$ holds trivially.

Similarly one can show that $(a \sqcup \langle \emptyset, A\rangle)b_+ = ab_+$. ⊠

4.28 LEMMA.

For every $m \in \mathbb{N}$ there exists a sequence $d_0, d_1, \ldots \in T^\omega$ such that for all $p \in \mathbb{N}$

0. $|d_p| \subseteq K^p m$.
1. $n \in |d_p| \Rightarrow b_{3^\circ} \sqsubseteq b_{[n]_p} \vee [n]_p \in \{3^1, \ldots, 3^p\}$.
2. d_p is lopsided.
3. For all $a \in T^\omega$ such that $|a| \cap K^p m = \emptyset$, one has

$$\forall c \in T^\omega \quad (a \sqcup d_p)c = \begin{cases} (ac) \sqcup d_{p-1} & \text{if } p > 0 \\ ac & \text{if } p = 0 \end{cases}$$

Proof. Define (depending on m)

$A_0 = K^\circ_{q_0} m$, $A_{p+1} = \{(\underline{+}n_0;\ldots;\underline{+}n_p;\underline{+}b_{3^{p+1}};m) \mid n_0,\ldots n_p \in \mathbb{N}\}$

$d_0 = \langle \emptyset, A_0\rangle$

$d_{p+1} = \underline{d}_p \sqcup \langle \emptyset, A_{p+1}\rangle$.

Note that for all p

(*) $A_p \subseteq K^p m$.

By induction on p we show 0, 1, 2 and $d_p \in T^\omega$.

<u>p = 0</u> : 0. $|d_0| = K^{\circ}_{q_0} m \subseteq K^{\circ} m$

1. $n \in |d_0| \Rightarrow n \in K_{3^0} m \Rightarrow b_{3^0} \sqsubseteq b_{[n]_0}$

2. $(d_0)_- = \emptyset$ is meager

$(d_0)_+ = K^{\circ}_{3^0} m$ is fat.

Clearly $d_0 \in T^{\omega}$.

<u>p + 1</u> : 0. Note that $|a| \subseteq K^p m \Rightarrow |\underline{a}| \subseteq K^{p+1} m$.

Hence by the induction hypothesis and (*) $|d_{p+1}| =$

$= |\underline{d}_p| \cup A_{p+1} \subseteq K^{p+1} m$.

1. Let $n \in |d_{p+1}|$. Case 1. $n \in |\underline{d}_p|$.

Then $n = (\underline{+};n')$ with $n' \in |d_p|$. Now $[n]_{p+1} = [n']_p$.

Hence by the induction hypothesis

(**) $b_{3^0} \sqsubseteq b_{n_{p+1}} \vee [n]_{p+1} \in \{3^1,\ldots,3^p\}$ and we are done.

Case 2. $n \in A_{p+1}$. Then $[n]_{p+1} = 3^{p+1}$, which also gives the result.

2. $(d_{p+1})_- = (\underline{d}_p)_- = (-0;(d_p)_-) \cup (+0;(d_p)_+)$ and this is meager. $(d_{p+1})_+ = (\underline{d}_p)_+ \cup A_{p+1}$, which is fat for $K^{\circ}_0(-n_1;\ldots;-n_p;\underline{+}3^{p+1};m) \subseteq A_{p+1}$. In order to have $d_{p+1} \in T^{\omega}$ we must show that $(\underline{d}_p)_- \cap A_{p+1} = \emptyset$. If $n \in (\underline{d}_p)_-$, then by (**) one has $b_{3^0} \sqsubseteq b_{[n]_{p+1}}$ $\vee [n]_{p+1} \in \{3^1,\ldots,3^p\}$. If $n \in A_p$, then $[n]_p = 3^{p+1}$. Hence by the choice of the q_i both conditions cannot be satisfied simultaneously.

This concludes the inductive argument for 0, 1, 2.

Finally we prove 3 by induction on p. Assume $|a| \cap K^p m = \emptyset$

<u>p = 0</u> : $(a \sqcup d_0)c = (a \sqcup \langle \emptyset, A_0 \rangle)c = ac$, by 4.27.

<u>p > 0</u> : $(a \sqcup d_p)c = (a \sqcup \underline{d}_{p-1} \sqcup \langle \emptyset, A_p \rangle)c$

$= (a \sqcup \underline{d}_{p-1})c$, by 4.27 since $A_p \subseteq K^p m$.

$= (ac) \sqcup d_{p-1}$, by 4.25.

(one has $a \dagger \underline{d}_{p-1}$ since $|\underline{d}_{p-1}| \subseteq K^p m$ and $|a| \cap K^p m = \emptyset$ ⊠

4.29 LEMMA.

Let $a \sqsubseteq \lambda x.ax$. Write $K^{-1}m = \{m\}$. Then for all $p \in \mathbb{N}$
$K^p m \cap |a| = \emptyset$ iff $\forall c \in T^\omega\ K^{p-1} \cap |ac| = \emptyset$.

Proof. $\Rightarrow$: $m_1 \in K^{p-1}m \cap |ac|$ implies that $\exists\, b_{n_0} \sqsubseteq c$

$(-n_0;m_1) \in |a| \cap K^p m$, a contradiction.

$\Leftarrow$: $(\underline{+}n_0;m_1) \in K^p m \cap |a|$ implies that $(\underline{+}n_0;m_1) \in |\lambda x.ax|$,

thus $\exists\, b_{n_0'} \sqsupseteq b_{n_0}\ m_1 \in |ab_{n_0'}| \cap K^{p-1}m$, a contradiction.

⊠

4.30 DEFINITION.

Let $p \in \mathbb{N}$. Then $a \in T^\omega$ is <u>p-fat</u> (<u>p-lopsided</u>) iff $\forall i \leq p$
$\forall \vec{c}^{\,i} \in T^\omega\ a\vec{c}^{\,i}$ is fat (lopsided).

4.31 LEMMA.

Let $a \in T$ and $p \in \mathbb{N}$. Suppose:

1. a is p-meager.
2. $a \sqsubseteq \lambda x_0 \ldots x_p.ax_0 \ldots x_p$.
3. $\exists\, m_0 \in \mathbb{N}\ \ \forall \vec{c}^{\,p+1} \in T^\omega\ \ m_0 \notin |a\vec{c}^{\,p+1}|$.

Then there exists $a^+ \in T^\omega$ such that

4. $a^+ \sim_p a$
5. a^+ is lopsided.

Proof. Let $m_0 \in \mathbb{N}$ be given by 3. Then one has

$$(*)\ \begin{cases} \forall \vec{c}^{\,p} \in T^\omega\ \ K^{-1}m_0 \cap |a\vec{c}^{\,p+1}| = \emptyset; \text{ hence by 4.29} \\ \forall \vec{c}^{\,p-1} \in T^\omega\ \ K^0 m_0 \cap |a\vec{c}^{\,p}| = \emptyset; \text{ hence} \\ \quad\vdots \\ \forall c \in T^\omega\ \ K^{p-1}m_0 \cap |ac_1| = \emptyset; \text{ hence} \end{cases}$$

$$K^p m_0 \cap |a| = \emptyset.$$

Let d_q be the sequence constructed in 4.28 for m_0. Then for all q one has $|d_q| \subseteq K^q m_0$. Define $a^+ = a \sqcup d_p$ (one has $a \uparrow d_p$ since $|a| \cap K^p m_0 = \emptyset$). Now let $i < p+1$, then

$$(a \sqcup d_p)\vec{c}^{\,i} = ((ac_1) \sqcup d_{p-1})\vec{c}^{\,i-1}$$
$$= \ldots$$
(**)
$$\vdots$$
$$= (a\vec{c}^{\,i}) \sqcup d_{p-i}, \text{ by lemma 4.28.3 since by (*)}$$
$\forall j < i$ one has

$\forall \vec{c}^{\,j} \in T^\omega \; |a\vec{c}^{\,j}| \cap K^{p-j} m_0 = \emptyset$.

It follows that

$a^+\vec{c}^{\,i} = \langle a\vec{c}_-^{\,i} \cup (d_{p-1})_-, a\vec{c}_+ \cup (d_{p-1})_+ \rangle$. Since the d_p's are fat $a^+\vec{c}^{\,i}$ is fat too. But $a^+\vec{c}^{\,i}$ is also meager for $a^+\vec{c}_-^{\,i}$ and $(d_{p-i})_-$ are meager and moreover they are disjoint: $|d_{p-i}| \subseteq K^{p-i} m_0$ and by (*) $K^{p-i} m_0 \cap |a\vec{c}^{\,i}| = \emptyset$.

Therefore a^+ is lopsided. Now by (**) for $i = p$, one has

$$a^+\vec{c}^{\,p+1} = (a\vec{c}^{\,p} \sqcup d_0)_{p+1}$$
$$= a\vec{c}^{\,p+1}$$

Since by (*) one has $K^0 m \cap |a\vec{c}^{\,p}| = \emptyset$ and 4.28 applies. Therefore $a^+ \sim_p a$. ⊠

4.32 THEOREM.

For all $p \in \mathbb{N}$ there exists $C^-_p \in T^\omega$ such that

1. $C^-_p \sim_{p+1} C_p$
2. C^-_p is p-lopsided

Proof. Take $a = C^\sim_{p+}$ of 4.16. By 4.17 a satisfies the assumptions of 4.31. Thus $\exists\, a^+ = C^-_p$ in T^ω such that $C^-_p \sim_p C^\sim_p \sim_{p+1} C_p$, by 4.31.4; hence $C^-_p \sim_{p+1} C_p$. Moreover C^-_p is p-lopsided by 4.31.5.

⊠

4.33 Proof of lemma 3.3.

Take C_p^- as in 4.32.

(i) By 4.32.1.

(ii) Let $n < n'$; applying both sides to $x_0,\ldots,x_n$, one has
$\lambda x_0.\ldots.x_n.\ C_p^-\sigma_0.\ldots.\sigma_m * \lambda x_{n+1}.\ldots.x_{n'}.\ C_p^-\tau_0.\ldots.\tau_{m'}.$ iff

(1) $C_p^-\sigma_0.\ldots.\sigma_m * \lambda x_{n+1}.\ldots.x_{n'}.\ C_p^-\tau_0.\ldots.\tau_{m'}.$

In order to prove (1) we check the condition in Corollary 4.23.
First, since C_p^- is p-lopsided and $m < p$ one has $C_p^-\sigma_0.\ldots.\sigma_m$ is lopsided. Moreover $\exists\ c \in T^\omega\ (\lambda x_{n+1}.\ldots.x_{n'}.\ C_p^-\tau_0.\ldots.\tau_{m'})c \neq \bot$.
Otherwise let $k = n' - n$: since $\bot a = \bot$, for all $\vec{c}^k \in T^\omega$ and all $\tau_{m'+1},\ldots,\tau_p$, one has $(\lambda x_{n+1}.\ldots.x_{n'}.\ C_p^-\tau_0.\ldots.\tau_{m'})\vec{c}^k\tau_{m'+1}\ldots\tau_p =$
$= C_p^-\tau_0.\ldots.\tau_p = \bot$.

Thus by 4.32.1 $\forall a \in T^\omega\ \bot = C_p^-\tau_0\ldots\tau_p a =$
$= a\tau_0\ldots\tau_p$ which is impossible. Now 4.23 applies. ⊠

FOOTNOTES

1) $BT(M) \subseteq BT(N)$ iff $BT(M)$ results from $BT(N)$ by replacing some subtrees by Ω.

2) As pointed out in Hindley and Longo [3], the notion of w.e. is stated too weakly in Barendregt [1]. W.e. should have been defined as (ξ) above. Indeed a pre-λ-algebra satisfying (ξ) is automatically a λ-algebra, i.e. satisfies all provable equations, see Hindley et al. [3], th. 3.5.

3) Correction for Barendregt (1977), [1], 7.6: Add the hypothesis "P in $\underline{\Omega}$-nf". The lemma remains true as stated, but the approximation theorem is needed for this.

AKNOWLEDGEMENTS

This research has been partially supported by the Italian Consiglio Nazionale delle Ricerche and by the Netherlands Organization for the advancement of Pure Research (Z.W.O.).

REFERENCES

[1] Barendregt, H. (1977). The type Free Lambda Calculus. *In* "Handbook of Mathematical Logic" (Ed. J. Barwise), pp. 1092-1132. North Holland, Amsterdam.

[2] Barendregt, H. (1980). "The Lambda Calculus, its Syntax and Semantics", North-Holland, Amsterdam. (To appear).

[3] Hindley, R. and Longo, G. (1980). Lambda Calculus Models and Extensionality, *Zeit. Math. Logik Grund.* 26, 2 (To appear).

[4] Hyland, M. (1976). A syntactic characterization of equality in some models of the λ-calculus. *J. London Math. Soc.* 12, 2, pp. 361-370.

[5] Lévy, J.J. (1975). An algebraic intepretation of the λβκ-calculus and a labelled λ-calculus. *In* "λ-Calculus and Computer Science theory", (Ed. C. Böhm) LNCS Vol. 37 pp. 147-165. Springer, Berlin.

[6] Plotkin, G. (1978). T^{ω} as a universale domain, *J. Computer and System Sciences* 17, 2 pp. 209-236.

[7] Scott, D. (1972). Continuous lattices , Lectures Notes in Mathematics Vol. 274, pp. 97-136. Springer Verlag, New York.

[8] Scott, D (1976). Data types as lattices, *SIAM J. Comp.* 5, 3 pp. 522-587.

[9] Volken, H. (1978). Formale Stetigkeit und Modelle des Lambda Kalkuls, Ph.D. Thesis, E.T.H., Zürich.

REFLEXIVE DOMAINS

Luis E. Sanchis

School of Computer and Information Science
Syracuse University, Syracuse, N.Y., 13210, U.S.A.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

0. INTRODUCTION

0.1 Models for Combinatory Logic and Lambda Calculus involve, in general, a domain of objects, and a set of operations that can be represented by elements of the domain. This is a kind of reflexive property, and some authors use the expression reflexive structures to denote collections of this form (see Wagner (1969)). The class of representable operations depends on the particular construction; the only requirement is that all the operations which are definable in the language should be represented.

We study here reflexive structures in which the basic domains are complete lattices, and all continuous operations are representable. This works as a model only if the representation itself is continuous. Models of this form were discovered by D. Scott, and following Stoy we call them reflexive domains.

We are primarily interested in reflexive domains as models of some languages containing lambda abstraction and combinators. But to some extent the study can be pursued quite independently of the particular language. The reason is that the representable operations are independent of the language.

In this paper we study a number of language independent properties of domains. We propose a classification and consider several constructions, in particular the inverse limit model, that we generalize by using representations in place of projections.

0.2 We shall assume familiarity with the notation and results of Sanchis (1977) which from now on will be referred to as DTL. Here again what we call a *lattice* is a complete lattice in the standard literature. If D is a lattice then $\subseteq$, o, u, $\cup$, $\cap$ denote, respectively, the partial order relation, the least element (bottom), the greatest element (top), the least upper bound and the greatest lower bound.

A *continuous representation* between lattices D and D' is a pair (f,g) where

$$f \in D \overset{c}{\to} D', \quad g \in D' \overset{c}{\to} D, \quad I \subseteq g \circ f, \quad f \circ g \subseteq I .$$

The symbol I denotes always the *identity function* on some set which is clear from the context. If (f,g) is a continuous representation between D and D' then f is additive and g is coadditive. Furthermore, if $g \in D' \overset{c}{\to} D$ and g is also coadditive there is a unique f such that (f,g) is a continuous representation between D and D'. If (f,g) is a continuous representation between D and D' and $I = g \circ f$ ($I = f \circ g$) we say that (f,g) is a *continuous projection* (*continuous closure*) between D and D'.

0.3 An *isomorphism* between lattices D and D' is a pair (v,w) where $v \in D \overset{m}{\to} D'$, $w \in D' \overset{m}{\to} D$, $w \circ v = I$ and $v \circ w = I$.

0.4 The notation we use for functions is quite strict. A function has always a domain and a codomain and a change in one of them implies a change in the function. To avoid ambiguities we use as much as possible composition between properly related functions.

0.5 The following notation will be used in several places. Suppose that F is a continuous retraction in a lattice D and define $D' = F(D) =$ the set of fixed points of F. Then we can introduce a function $F' \in D \overset{c}{\to} D'$ such that for $x \in D$, $F'(x) = F(x)$. We can also introduce a function $I' \in D' \overset{c}{\to} D$ such that for $x \in D'$, $I'(x) = x$. If the retraction is denoted by F_n then the new functions are denoted by F'_n and I'_n. The following obvious relations will be used without further explanation:

$$I' \circ F' = F$$
$$F' \circ I' = I$$
$$F' \circ F = F'$$
$$F \circ I' = I'$$

1. REFLEXIVE DOMAINS

1.1 A *reflexive domain* is a triple (D,ϕ,ψ) where D is a lattice, $\phi \in D \overset{c}{\to} (D \overset{c}{\to} D)$ and $\psi \in (D \overset{c}{\to} D) \overset{c}{\to} D$, such that $\phi \circ \psi = I$. In the notation of DTL this means that (ϕ,ψ) is a continuous retraction of $D \overset{c}{\to} D$ into D. A reflexive domain can be used as a model for the λ-calculus. The details of such semantics will not be considered here. The definitions in Wadworth (1976) can be translated verbatim even if the notion introduced here is more general. In fact our definition involves the minimum necessary for the evaluation of terms of the λ-calculus. Rules for α- and β-conversion are preserved in this type of model but extensionality in general fails. Still some weak form of extensionality is satisfied.

A reflexive domain (D,ϕ,ψ) such that $I \subseteq \psi \circ \phi$ is called *additive*, and in case $\psi \circ \phi \subseteq I$ is called *coadditive*. Finally if (ϕ,ψ) is an isomorphism between D and $D \overset{c}{\to} D$ we shall say that the reflexive domain is *strict*.

A non trivial reflexive domain is the so-called graph model of Scott (1976) denoted $(P\omega,\phi_\omega,\psi_\omega)$ where for $x \in P\omega$ and $f \in P\omega \overset{c}{\to} P\omega$ we have

$$\phi_\omega(x) = \lambda y.\ \{n : \exists m.\ \langle n,m\rangle \in x \wedge e_m \subseteq y\}$$

$$\psi_\omega(f) = \{\langle n,m\rangle : n \in f(e_m)\}$$

This domain is additive. (See also the comment in 3.6.5 of DTL.) Other domains can be obtained using the inverse limit construction that we discuss in the next section.

1.2 An *isomorphism* between the reflexive domains (D,ϕ,ψ) and (D',ϕ',ψ') is a pair (v,w) where (v,w) is an isomorphism between D and D' such that

(1) $\phi'(x) = v \circ \phi(w(x)) \circ w$

(2) $\psi'(f) = v(\psi(w \circ f \circ v))$

where $x \in D'$ and $f \in D' \overset{c}{\to} D'$. It follows easily that (w,v) is also an isomorphism between (D',ϕ',ψ') and (D,ϕ,ψ). Furthermore $\psi' \circ \phi' = v \circ \psi \circ \phi \circ w$; hence if (D,ϕ,ψ) is additive (coadditive) (strict) then (D',ϕ',ψ') is also additive (coadditive) (strict).

If (D,ϕ,ψ) and (D',ϕ',ψ') are both additive reflexive domains the proof that (v,w) is an isomorphism can be simplified. In fact it is sufficient to show that either (1) or (2) holds. For suppose (1) holds. Then we can define $\psi''(f) = v(\psi(w \circ f \circ v))$ and it follows easily that (ϕ',ψ'') is a representation between D' and $D' \overset{c}{\to} D'$ hence $\psi' = \psi''$. And similarly we can prove that (1) holds assuming that (2) holds. The same argument can be used in the case where both domains are coadditive.

1.3 Let (D,ϕ,ψ) be some reflexive domain. We introduce an operator $Q \in (D \overset{c}{\to} D) \overset{c}{\to} (D \overset{c}{\to} D)$ by the following definition

$$Q(f) = \lambda x.\psi(f \circ \phi(x) \circ f).$$

First note that in case F is a continuous retraction in D then $Q(F)$ is also a continuous retraction in D. Furthermore if F is a continuous closure in D and the domain is additive the nQ(F) is a continuous closure in D. And if F is continuous projection in D and the domain is coadditive then $Q(F)$ is a continuous projection in D.

1.4 LEMMA. $Q(f) \circ Q(g) \circ Q(f) = Q(f \circ g \circ f)$.

This follows by straight computation using the definitions.

A continuous retraction F in D such that $Q(F) \circ F \circ Q(F) = Q(F)$ is said to be *closed* in the domain (D,ϕ,ψ). Note that the identity retraction I is always closed. Furthermore from Lemma *1.4* it follows that whenever F is closed then Q(F) is also closed. Then $Q(I) = \psi \circ \phi$ is always a closed retraction.

1.5 Let (D,ϕ,ψ) be a reflexive domain, and F a continuous retraction closed in the domain. We put $D' = F(D)$ and introduce the following functions:

$$\phi' \in D' \overset{c}{\to} (D' \overset{c}{\to} D') \quad \text{and} \quad \psi' \in (D' \overset{c}{\to} D') \overset{c}{\to} D' .$$

$$\phi'(x) = F' \circ \phi(x) \circ I'$$

$$\psi'(f) = F(\psi(I' \circ f \circ F'))$$

We use here the notation explained in *0.5*. We shall prove that (D',ϕ',ψ') is also a reflexive domain. We put $F_1 = Q(F)$ and take $f \in D' \overset{c}{\to} D'$. From the definition of Q it follows that

$$F_1(\psi(I' \circ f \circ F')) = \psi(I' \circ f \circ F').$$

Hence from $F_1 \circ F \circ F_1 = F_1$ we get that

$$\psi(F \circ \phi(F(\psi(I' \circ f \circ F'))) \circ F) = \psi(I \circ f \circ F');$$

hence using $\phi \circ \psi = I$ and the equations of *0.5* we get

$$\phi'(\psi'(f)) = F' \circ I' \circ f \circ F' \circ I' = f .$$

We shall say that the reflexive domain (D',ϕ',ψ') is *induced* by the closed retraction F. If (D',ϕ',ψ') is any reflexive domain isomorphic to a reflexive domain induced by retraction F closed in the domain (D,ϕ,ψ) we say that (D',ϕ',ψ') is a *retraction* of (D,ϕ,ψ).

1.6 Using the same notation introduced in *1.5* note that $\psi' \circ \phi' = F' \circ Q(F) \circ I'$, hence (D',ϕ',ψ') is strict if and only if $F \circ Q(F) \circ F = F$. Also in case the domain (D,ϕ,ψ) is additive (coadditive) and the retraction F is a closure (projection) then the domain (D',ϕ',ψ') is additive (coadditive).

1.7 THEOREM. Let (D,ϕ,ψ) be a reflexive domain and F a continuous retraction in D. If one of the following conditions is satisfied then F is closed in the domain.

i) $I \subseteq F \subseteq Q(F)$

ii) $I \subseteq F \subseteq \psi \circ \phi$

iii) $Q(F) \subseteq F \subseteq I$

iv) $\psi \circ \phi \subseteq F \subseteq I$

v) $\phi \circ F = \phi$

vi) $F \circ \psi = \psi$

vii) $F = \lambda x.\psi(F \circ \phi(x))$

viii) $F = \lambda x.\psi(\phi(x) \circ F)$

ix) $Q(F) = F$.

To prove i) note that from $I \subseteq F$ it follows that $Q(F) \subseteq Q(F) \circ F \circ Q(F)$ and from $F \subseteq Q(F)$ it follows that $Q(F) \circ F \circ Q(F) \subseteq Q(F)$. Proof of the other properties can be obtained by similar arguments.

The case $Q(F) = F$ is the most important and here the domain induced by F is strict. Assume that F_0 is a continuous retraction in D such that $F_0 \subseteq Q(F_0)$. Then there exists a least continuous retraction F such that $Q(F) = F$ and $F_0 \subseteq F$. If we define $F_{n+1} = Q(F_n)$ for $n \geq 0$ we know that $F = \cup\{F_n : n \geq 0\}$. We shall give now two examples in which this construction is applied.

1.8 Consider the graph model and define a continuous retraction F_0 in P as follows. If $0 \in x$ then $F_0(x) = \{0\}$. If $0 \notin x$ then $F_0(x) = \emptyset$. Since the pairing used to define the functions is such that $\langle 0,0\rangle = 0$ we have $F_0 \subseteq Q(F_0)$.

1.9 Let (D,ϕ,ψ) be some additive reflexive domain in which the top element u is compact. Define a continuous retraction F_0 as follows. If $x \neq u$ then $F_0(x) = o$. Otherwise $F_0(x) = u$.

Again we have $F_0 \subseteq Q(F_0)$. We shall show later that, if $Q(F) = F$ is the least fixed point of Q extending this F_0, the domain induced by F is independent of the domain (D,ϕ,ψ) and it is isomorphic to the domain induced in Example *1.8*.

2. INVERSE LIMITS

2.1 In this section we present a general theory of the inverse limit construction and prove a representation theorem which has several applications. Essentially we are dealing with the category of lattices with continuous coadditive mappings as morphisms. But we prefer to describe the construction in terms of representations.

A *representation system* is an infinite sequence $R = \{D_n, i_n, j_n\}$, $n \geq 0$, where for each $n \geq 0$, D_n is a lattice and (i_n, j_n) is a continuous representation between D_n and D_{n+1}. In the case where for each n, (i_n, j_n) is a projection (closure) between D_n and D_{n+1} it is called a *projection* (*closure*) *system*.

Let R be a representation system and D' some lattice. A *representation* between R and D' is an infinite sequence $\{v_n, w_n\}$, $n \geq 0$, where for each $n \geq 0$, (v_n, w_n) is a continuous representation between D_n and D', and furthermore the following conditions are satisfied:

i) $w_n = j_n \circ w_{n+1}$

ii) $v_n = v_{n+1} \circ i_n$

The representation is called a *projection* (*closure*) in the case where for each n, (v_n, w_n) is a projection (closure) between D_n and D'.

2.2 Note that it is sufficient that one of the two conditions holds. For suppose i) holds. Then we can define $v'_n = v_{n+1} \circ i_n$ and it follows that (v'_n, w_n) is a representation between D_n and D', hence $v_n = v'_n$.

2.3 PROPOSITION. Let $\{v_n, w_n\}$ be a representation between a system $R = \{D_n, i_n, j_n\}$ and a lattice D'. If $m \geq n$ then:

i) $w_n \circ v_m \circ w_m = w_n$

ii) $v_n \circ w_n \circ v_m \circ w_m = v_n \circ w_n$

iii) $v_m \circ w_m \circ v_n = v_n$

iv) $v_m \circ w_m \circ v_n \circ w_n = v_n \circ w_n$

v) $w_n \circ v_m \circ w_m \circ v_n = w_n \circ v_n$

vi) $v_n \circ w_n \subseteq v_m \circ w_m$.

Properties i) and iii) are proved by induction on m-n. Then ii) follows from i) and iv) follows from iii). Property v) follows from i) and also from iii). To prove vi) note that

$$v_n \circ w_n = v_{n+1} \circ i_n \circ j_n \circ w_{n+1} \subseteq v_{n+1} \circ w_{n+1}$$

Note that in case the system R is a closure system we have $v_n \circ w_n = v_{n+1} \circ w_{n+1}$. On the other hand when $\{v_n, w_n\}$ is a projection between R and D' then $j_n \circ i_n = j_n \circ w_{n+1} \circ v_{n+1} \circ i_n = w_n \circ v_n = I$, so R is a projection system.

Let $R = \{D_n, i_n, j_n\}$ be some representation system. We define the *inverse limit* of R, which is a lattice denoted by IL(R). The elements of IL(R) are all functions $\sigma : \omega \to \bigcup_{n \geq 0} D_n$ such that for $n \geq 0$, $\sigma[n] \in D_n$ and $\sigma[n] = j_n(\sigma[n+1])$. Here $\sigma[n]$ denotes the application of the function σ to the argument n. We call $\sigma[n]$ the nth coordinate of σ.

Let σ and τ be elements of IL(R). Then we define $\sigma \subseteq \tau$ if and only if $\sigma[n] \subseteq_n \tau[n]$ for each $n \geq 0$.

2.4 THEOREM. IL(R) is a lattice. If $\alpha \subseteq$ IL(R) then

$$\cap \dot{\alpha}\, [n] = \cap_n \{\sigma[n] : \sigma \in \alpha\} \quad .$$

Furthermore if α is directed then

$$\cup\, \alpha\, [n] = \cup_n \{\sigma[n] : \sigma \in \alpha\} \quad .$$

This follows immediately from the fact that the functions j_n are coadditive and continuous.

There is a canonical representation between R and IL(R) which is denoted as $\{i_{n\infty}, j_{\infty n}\}$. We define for each $n \geq 0$

$$j_{\infty n}(\sigma) = \sigma[n]$$

so $j_{\infty n} \in IL(R) \to D_n$ is coadditive and continuous. It follows that there are unique functions $i_{n\infty}$ such that for each n the pair $(i_{n\infty}, j_{\infty n})$ is a continuous representation between D_n and IL(R). Since $j_{\infty n} = j_n \circ j_{n+1}$ by definition, we have a representation between R and IL(R).

This is the essential difference from the construction in Scott (1972). Here we cannot assume the representations are projections to define $i_{n\infty}$ by recursion. But $i_{n\infty}$ exists and it is proved in DTL that it is given by the following expression:

$$i_{n\infty}(x) = \cap\{\sigma : x \subseteq \sigma[n]\}$$

2.5 For each $n \geq 0$ we define $P_n = i_{n\infty} \circ j_{\infty n}$. Note that P_n is a continuous projection in IL(R).

The results of *2.3* can be reformulated using the notation for the canonical representation. Hence for $m \geq n$ we have:

i) $j_{\infty n} \circ P_m = j_{\infty n}$

ii) $P_n \circ P_m = P_n$

iii) $P_m \circ i_{n\infty} = i_{n\infty}$

iv) $P_m \circ P_n = P_n$

v) $j_{\infty n} \circ P_m \circ i_{n\infty} = j_{\infty n} \circ i_{n\infty}$

vi) $P_n \subseteq P_m$

Furthermore if R is a closure system then $P_n = P_m$.

2.6 THEOREM. If R is a projection system then $\{i_{n\infty}, j_{\infty n}\}$ is a projection between R and IL(R).

It is sufficient to prove that $j_{\infty n}$ is onto D_n (see DTL, *3.3*). If $x \in D_n$ define $\sigma[n] = x$ and $\sigma[m+1] = i_n(\sigma[m])$ for $m \geq n$. Since R is a projection system this σ is well defined, and clearly $j_{\infty n}(\sigma) = x$.

2.7 THEOREM. If R is a closure system then $\{i_{n\infty}, j_{\infty n}\}$ is a closure between R and IL(R).

Again we need only show that $j_{\infty n}$ is 1-1. If σ and τ are such that $j_{\infty n}(\sigma) = j_{\infty n}(\tau)$ then $\sigma[n] = \tau[n]$, so by induction using the closure property of R it follows that $\sigma[m] = \tau[m]$ for all $m \geq n$, hence $\sigma = \tau$.

2.8 Even at this general level we can obtain some information about the structure of IL(R). First since each $i_{n\infty}$ is additive we have $i_{n\infty}(o_n) = o$ for all $n \geq 0$. And since each $j_{\infty n}$ is coadditive we have $u[n] = u_n$ for all $n \geq 0$.

There are systems in which $i_n(u_n) = u_{n+1}$ for all $n \geq 0$. In this case we can prove that for arbitrary $m \geq n$ we have $j_{\infty m}(i_{n\infty}(u_n) = u_m$. The proof is straightforward using induction on m-n. But this means $i_{n\infty}(u_n) = u$ for all $n \geq 0$. From this it follows that $P_n(u) = u$ for all $n \geq 0$. Under the same assumption note that if $\sigma[n] = u_n$ for some n, $P_n(\sigma) = u$, hence $\sigma = u$.

There are systems in which $j_n(o_{n+1}) = o_n$ for all $n \geq 0$. In this case we have $o[n] = o_n$ for all $n \geq 0$.

2.9 THEOREM. $\cup\{P_n : n \geq 0\} = I$.

Put $P = \cup\{P_n : n \geq 0\}$. Then $P(\sigma) = \cup\{P_m(\sigma) : m \geq 0\}$. But then

$$P(\sigma)[n] = \cup\{P_m(\sigma)[n] : m \geq n\} \text{ (by } 2.4)$$
$$= \sigma[n] \quad \text{(by } 2.5)$$

hence $P(\sigma) = \sigma$.

2.10 THEOREM. If $f \in IL(R) \overset{c}{\to} IL(R)$ then $f = \cup\{P_n \circ f \circ P_n : n \geq 0\}$.

This follows by straightforward computation using *2.9*.

2.11 THEOREM. If σ is compact in IL(R) then $P_n(\sigma) = \sigma$ for some $n \geq 0$.

This follows clearly from *2.9*.

2.12 THEOREM. If x is compact in D_n then $i_{n\infty}(x)$ is compact in IL(R).

Proof. Assume $\alpha \subseteq IL(R)$ is directed and $i_{n\infty}(x) \subseteq \bigcup\alpha$. Then using *2.4* we have $x \subseteq i_{n\infty}(x)[n] \subseteq \cup\alpha[n]$, so for some $\sigma \in \alpha$ we have $x \subseteq \sigma[n]$. Hence $i_{n\infty}(x) \subseteq i_{n\infty}(\sigma[n]) \subseteq \sigma$.

2.13 REPRESENTATION THEOREM. Let $R = \{D_n, i_n, j_n\}$ be a representation system and let $\{v_n, w_n\}$ be a representation between R and a lattice D'. Then there is a unique continuous representation (v,w) between IL(R) and D' such that $w_n = j_{\infty n} \circ w$ and $v_n = v \circ i_{n\infty}$. Furthermore $v \circ w = \cup\{v_n \circ w_n : n \geq 0\}$,

$$w \circ v = \cup\{i_{n\infty} \circ w_n \circ v_n \circ j_{\infty n} : n \geq 0\}.$$

Proof. We define $v(\sigma) = \cup'\{v_n(\sigma[n]) : n \geq 0\}$

$$w(x)[n] = w_n(x).$$

Note first that w is well defined:

$$j_n(w(x)[n+1]) = j_n(w_{n+1}(x)) = w_n(x) = w(x)[n].$$

Now we show that (v,w) is a representation between IL(R) and D'.

$$v(w(x)) = \cup'\{v_n(w_n(x)) : n \geq 0\} \subseteq x .$$

Note that this equation proves $v \circ w = \cup\{v_n \circ w_n : n \geq 0\}$. On the other hand for σ in IL(R) and $n \geq 0$ we have

$$\begin{aligned} \sigma[n] &\subseteq w_n(v_n(\sigma[n])) \\ &\subseteq w_n(\cup'\{v_m(\sigma[m]) : m \geq 0\}) \\ &= w_n(v(\sigma)) = w(v(\sigma))[n] \end{aligned}$$

hence $\sigma \subseteq w(v(\sigma))$.

The continuity of w follows easily from the continuity of the functions w_n. Furthermore $w_n = j_{\infty n} \circ w$ follows from the definition of w. Note that $v_n = v \circ i_{n\infty}$ follows as in *2.2*.

The uniqueness of the function w is straightforward. The evaluation of $v \circ w$ was already discussed. We need only to complete the evaluation of $w \circ v$. Note that by *2.10*

$$w \circ v = \sqcup\{P_n \circ w \circ v \circ P_n : n \geq 0\}$$
$$= \sqcup\{i_{n\infty} \circ w_n \circ v_n \circ j_{\infty n} : n \geq 0\}.$$

This completes the proof.

2.14 COROLLARY. If $\{v_n, w_n\}$ is a closure (projection) between R and D' then (v,w) is a closure (projection) between IL(R) and D'.

This follows from the evaluations of $v \circ w$ and $w \circ v$ in the representation theorem.

2.15 In relation with the definition of $v(\sigma)$ in the proof of the representation theorem notice that

$$v_n(\sigma[n]) = v_n(j_n(\sigma[n+1])$$
$$= v_{n+1}(i_n(j_n(\sigma[n+1])))$$
$$\subseteq v_{n+1}(\sigma[n+1])$$

3. FUNCTIONAL SYSTEMS

3.1 The first application of the representation theorem is a generalization of the construction in Scott (1972). We shall show that the systems considered in this reference, but with representations in place of projections, always produce additive reflexive domains. The domain is strict in case the system is a projection; but also in other cases.

Let D_0 be some lattice, and (i_0, j_0) be some continuous representation between D_0 and $D_0 \overset{c}{\to} D_0$. For $n \geq 0$ define $D_{n+1} = D_n \overset{c}{\to} D_n$. The *functional representation system* induced by (D_0, i_0, j_0) is the system $\{D_n, i_n, j_n\}$ where the functions i_{n+1} and j_{n+1} are defined as follows:

$$i_{n+1}(f) = i_n \circ f \circ j_n$$
$$j_{n+1}(g) = j_n \circ g \circ i_n \ .$$

Here $f \in D_{n+1} = D_n \overset{c}{\to} D_n$ and $g \in D_{n+2} = D_{n+1} \overset{c}{\to} D_{n+1}$. It can easily be verified that in this way we get a representation system. This system will be denoted by $R = [D_0, i_0, j_0]$.

3.2 Several examples of interesting representation between lattices and their spaces of continuous functions are well known. We discuss briefly two possible methods.

First take an additive reflexive domain (D,ϕ,ψ) and put $D_0 = D$, $i_0 = \phi$ and $j_0 = \psi$. The functional system $R = [D_0, i_0, j_0]$ in this case is a closure system.

To get a projection system take a lattice D_0 containing some compact element t (for instance take t = o). Define $i_0 \in D_0 \overset{m}{\to} (D_0 \overset{c}{\to} D_0)$ as follows:

$$i_0(x)(y) = x \text{ if } t \subseteq y$$
$$= o \text{ otherwise.}$$

From the compactness of t it follows that $i_0(x) \in D_0 \overset{c}{\to} D_0$. Now for $f \in D_0 \overset{c}{\to} D_0$ define $j_0(f) = f(t)$ and it can be easily checked that (i_0, j_0) is a continuous projection between D_0 and $D_0 \overset{c}{\to} D_0$.

3.3 Let $R = [D_0, i_0, j_0]$ be some functional representation system. We put $D_\infty = IL(R)$ and introduce $D' = D_\infty \overset{c}{\to} D_\infty$. In order to apply the representation theorem we define a representation $\{v_n, w_n\}$ between R and D' as follows:

$$v_{n+1}(f) = i_{n\infty} \circ f \circ j_{\infty n} \qquad v_0 = v_1 \circ i_0$$
$$w_{n+1}(g) = j_{\infty n} \circ g \circ i_{n\infty} \qquad w_0 = j_0 \circ w_1$$

It is easy to verify that we have a representation between R and D'. By the representation theorem there is a unique continuous representation (v,w) between D_∞ and $D_\infty \overset{c}{\to} D_\infty$. Since $v_{n+1}(w_{n+1}(f)) = P_n \circ f \circ P_n$ for $f \in D_\infty \overset{c}{\to} D_\infty$ and $n \geq 0$, and $v_0 \circ w_0 \subseteq v_1 \circ w_1$, it follows that $v \circ w = I$, so actually we have a closure. We put $\phi_\infty = v$ and $\psi_\infty = w$ and $(D_\infty, \phi_\infty, \psi_\infty)$ is an additive reflexive domain.

In case R is a projection system we have $w_n \circ v_n = I$ for each $n \geq 0$, hence by *2.14* the domain is strict. We shall prove later that the same result is true in case R is a closure system.

If the initial representation is such that $i_0(u_0) = u_1$ then we have $i_n(u_n) = u_{n+1}$ for all n. It follows from *2.8* that $i_{n\infty}(u_n) = u_\infty$ and in case u_0 is compact in D_0 then u_∞ is compact in D_∞.

3.4 A *Scott system* is a functional system $[D_0,i_0,j_0]$ where the function i_0 is defined as follows for $x \in D_0$: $i_0(x) = \lambda y.x$ (or equivalently where the function j_0 is defined as follows for $f \in D_0 \overset{c}{\to} D_0$: $j_0(f) = f(o)$). The domain $(D_\infty,\psi_\infty\psi_\infty)$ induced by a Scott system is called a Scott domain.

Note that in a projection system we have $P_n(\sigma)[n+1] = i_n(\sigma[n])$ for arbitrary $n \geq 0$, since

$$\begin{aligned} j_{\infty n+1} \circ i_{n\infty} \circ j_{\infty n} &= j_{\infty n+1} \circ i_{n+1\infty} \circ i_n \circ j_{\infty n} \\ &= i_n \circ j_{\infty n} \end{aligned}$$

3.5 THEOREM. If $R = [D_0,i_0,j_0]$ is a projection functional system then the following relations hold for arbitrary $\sigma \in IL(R)$ and $m \geq n \geq 0$;

i) $j_{\infty n} \circ i_{m\infty} \circ \sigma[m+1] \circ j_{\infty m} \circ i_{n\infty} = \sigma[n+1]$

ii) $i_{m\infty} \circ P_{n+1}(\sigma)[m+1] \circ j_{\infty m} = i_{n\infty} \circ \sigma[n+1] \circ j_{\infty n}$

iii) $j_{\infty n} \circ \phi_\infty(\sigma) \circ i_{n\infty} = \sigma[n+1]$

iv) $P_n \circ \phi_\infty(\sigma) \circ P_n = i_{n\infty} \circ \sigma[n+1] \circ j_{\infty n}$

v) $\phi_\infty(P_{n+1}(\sigma)) = i_{n\infty} \circ \sigma[n+1] \circ j_{\infty n}$

vi) $P_{n+1}(\sigma) = \psi_\infty(P_n \circ \phi_\infty(\sigma) \circ P_n)$

vii) $\phi_\infty(P_n(\sigma)) = i_{n\infty} \circ i_n(\sigma[n]) \circ j_{\infty n}$.

The proof of i) and ii) is by induction on m-n. Details are omitted but note that $P_{n+1}(\sigma) = P_{n+2}(P_{n+1}(\sigma))$. Now using i) we get iii) by the following computation:

$$\begin{aligned} j_{\infty n} \circ \phi_\infty(\sigma) \circ i_{n\infty} &= \cup\{j_{\infty n} \circ v_{m+1}(\sigma[m+1]) \circ i_{n\infty} : m \geq n\} \\ &= \sigma[n+1] \end{aligned}$$

From iii) we get iv). To get v) we apply the definition of $\phi_\infty(P_{n+1}(\sigma))$ and use ii). Using iv), v) and the fact that the domain is strict we get vi). Finally vii) follows from v) using $P_n = P_{n+1} \circ P_n$.

3.6 Note that property vi) means that $P_{n+1} = Q(P_n)$ where Q is the operator introduced in Section 1. So the minimal fixed point of Q extending P_0 is the identity.

4. LIMITS OF RETRACTIONS

4.1 The second application of the representation theorem considers fixed points of operators which produce continuous retractions whenever applied to continuous retractions. Such fixed points are limits of increasing sequences of retractions and we shall show that they are isomorphic to non-trivial inverse limits constructions. Here again we generalize Scott's results, in particular the characterization of infinite data structures given in Theorem 4.6 of Scott (1976).

4.2 Let D be some lattice and $F_0, F_1, \ldots, F_n, \ldots$ be a sequence of continuous retractions in D such that $F_n \subseteq F_{n+1}$ for $n \geq 0$. Then $F = \cup\{F_n : n \geq 0\}$ is also a continuous retraction. We put $D'_n = F_n(D)$ and $D' = F(D)$. Next we introduce a representation system $R' = \{D'_n, i'_n, j'_n\}$ where $i'_n = F'_{n+1} \circ I'_n$ and $j'_n = F'_n \circ I'_{n+1}$. It follows that $j'_n \circ i'_n = F'_n \circ F_{n+1} \circ I'_n$ and $i'_n \circ j'_n = F'_{n+1} \circ F_n \circ I'_{n+1}$. Hence using $F_n \subseteq F_{n+1}$ and noting that $F'_n \circ F_n \circ I'_n = I$ and $F'_{n+1} \circ F_{n+1} \circ I'_{n+1} = I$ we get $I \subseteq j'_n \circ i'_n$ and $i'_n \circ j'_n \subseteq I$. We denote by $\{i'_{n\infty}, j'_{\infty n}\}$ the canonical representation between R' and IL(R').

Next we introduce a representation $\{v'_n, w'_n\}$ between R' and D'. We put $v'_n = F' \circ I'_n$ and $w'_n = F'_n \circ I'$. Again using $F_n \subseteq F$ we prove the conditions $I \subseteq w'_n \circ v'_n$ and $v'_n \circ w'_n \subseteq I$. The conditions $w'_n = j'_n \circ w'_{n+1}$ follows by a simple computation.

4.3 By the representation theorem there is a unique continuous representation (v',w') between IL(R') and D' such that

$w'_n = j'_{\infty n} \circ w'$ and $v'_n = v' \circ i'_{n\infty}$. We shall prove this representation is actually an isomorphism.

First note that when $\sigma \in IL(R')$ and $n \geq 0$ then $\sigma[n] = F_n(\sigma[n+1]) \subseteq F_{n+1}(\sigma[n+1]) = \sigma[n+1]$ hence $\sigma[n] \subseteq \sigma[m]$ for $n \leq m$.

In the following $\cup$ denotes the l.u.b. operation in D and $\cup'$ the l.u.b. operation in D'. Since D' is of finite character in D, both operations agree on a directed subset of D'

$$\begin{aligned} v'(\sigma) &= \cup'\{v'_n(\sigma[n]) : n \geq 0\} \\ &= \cup'\{F(\sigma[n]) : n \geq 0\} \\ &= \cup\{F(\sigma[n]) : n \geq 0\} \\ &= \cup\{F_m(\sigma[n]) : m \geq 0,\ n \geq 0\} \\ &= \cup\{F_n(\sigma[n]) : n \geq 0\} \\ &= \cup\{\alpha[n] : \geq 0\}\ . \end{aligned}$$

For $n \leq m$ we have $F_n(\sigma[m]) = \sigma[n]$. First from $\sigma[n] \subseteq \sigma[m]$ we get $\sigma[n] \subseteq F_n(\sigma[m])$. To prove the converse we use induction on m-n. The case $m = n$ is trivial. If $m > n$ we use $F_n \subseteq F_n \circ F_{n+1}$, hence $F_n(\sigma[m]) \subseteq F_n(F_{n+1}(\sigma[m])) \subseteq F_n(\sigma[n+1]) = \sigma[n]$.

Now take $y \in D'$; using the representation theorem we have

$$\begin{aligned} v'(w'(y)) &= \cup\{w'(y)[n] : n \geq 0\} \\ &= \cup\{w'_n(y) : n \geq 0\} \\ &= \cup\{F_n(y) : n \geq 0\} \\ &= F(y) = y\ . \end{aligned}$$

Next we take $\sigma \in IL(R')$:

$$\begin{aligned} w'(v'(\sigma))[n] &= F_n(\ \cup\{\sigma[m] : m \geq n\}) \\ &= \cup\{F_n(\sigma[m]) : m \geq n\} \\ &= \sigma[n]\ , \end{aligned}$$

$$w'(v'(\sigma)) = \sigma\ .$$

5. APPLICATIONS TO RETRACTION DOMAINS

5.1 We return now to the construction of reflexive domains using retractions in a given domain. We fix some reflexive domain (D,ϕ,ψ) and again denote by Q the operator introduced in *1.3*. Let F_0 be a continuous retraction in D such that $F_0 \subseteq Q(F_0)$. We put $F_{n+1} = Q(F_n)$ so $F_0, F_1, \ldots, F_n, \ldots$ is an increasing sequence of continuous retractions. If $F = \cup\{F_n : n \geq 0\}$ then $Q(F) = F$ and F induces a reflexive domain $(D',\phi'\psi')$ as defined in *1.5*. This domain is strict (see the remark *1.6*).

Since we can apply the definitions in *4.2* we have a representation system $R' = \{D'_n, i'_n, j'_n\}$ and an isomorphism (v',w') between $IL(R')$ and D'.

5.2 We shall introduce next a functional system which is isomorphic to the system R'. To simplify the notation we introduce two new operators

$$J_{n+1} \in D \overset{c}{\to} (D'_n \overset{c}{\to} D'_n)$$

$$L_{n+1} \in (D'_n \overset{c}{\to} D'_n) \overset{c}{\to} D$$

where $n \geq 0$. If $y \in D$ and $f \in D'_n \overset{c}{\to} D'_n$ we put

$$J_{n+1}(y) = F'_n \circ \phi(y) \circ I'_n$$

$$L_{n+1}(f) = \psi(I'_n \circ f \circ F'_n).$$

By simple computations it follows that

$$J_{n+1} \circ L_{n+1} = I$$

$$L_{n+1} \circ J_{n+1} = F_{n+1}$$

and from these equations it follows that

$$J_{n+1} \circ F_{n+1} = J_{n+1}$$

$$F_{n+1} \circ L_{n+1} = L_{n+1} \; .$$

5.3 We can define now the functional system $R = [D_0, i_0, j_0]$ where $D_0 = D'_0$, $i_0 = J_1 \circ I'_0$, $j_0 = F'_0 \circ L_1$. Using $F_0 \subseteq F_1$ it can be shown that $I \subseteq j_0 \circ i_0$ and $i_0 \circ j_0 \subseteq I$.

The system R induces a reflexive domain $(D_\infty, \phi_\infty, \psi_\infty)$. This domain is additive.

5.4 We want to show that the two domains (D', ϕ', ψ') and $(D_\infty, \phi_\infty, \psi_\infty)$ are isomorphic. For this we must show first that R' and R are isomorphic systems. For this purpose we introduce isomorphism (r_n, s_n) between D'_n and D_n, $n \geq 0$. Since $D'_0 = D_0$ we put $r_0 = s_0 = I$. Assume that the isomorphism (r_n, s_n) has been defined. For $y \in D'_{n+1}$ and $f \in D_{n+1}$ we put

$$r_{n+1}(y) = r_n \circ J_{n+1}(y) \circ s_n$$

$$s_{n+1}(f) = L_{n+1}(s_n \circ f \circ r_n).$$

We must check $s_{n+1}(f) \in D'_{n+1}$; this follows from $F_{n+1} \circ L_{n+1} = L_{n+1}$. The computations to show (r_{n+1}, s_{n+1}) is an isomorphism are straightforward.

It is not sufficient to have isomorphisms between D'_n and D_n. They must preserve the representations in such a way that the following relations are satisfied:

$$i_n = r_{n+1} \circ i'_n \circ s_n$$

$$j_n = r_n \circ j'_n \circ s_{n+1} .$$

We prove these relations by induction on n. We need to prove only one of them; the other follows because we are dealing with representations. For the case $n = 0$ note that $r_1 = J_1 \circ I'_1$, hence

$$\begin{aligned} r_1 \circ i'_0 \circ s_0 &= J_1 \circ I'_1 \circ F'_1 \circ I'_0 \\ &= J_1 \circ F_1 \circ I'_0 \\ &= J_1 \circ I'_0 \quad = i_0 . \end{aligned}$$

Assume now the relations hold for n. Then by direct evaluation we get

$$r_{n+2}(i'_{n+1}(s_{n+1}(f))) = r_{n+1} \circ i'_n \circ s_n \circ f \circ r_n \circ j'_n \circ s_{n+1}$$

$$r_{n+2}(i'_{n+1}(s_{n+1}(f))) = i_n \circ f \circ j_n = i_{n+1}(f)$$

This isomorphism between the systems R' and R induces a natural isomorphism (r,s) between IL(R') and IL(R) = D. In fact if $\tau \in$ IL(R') and $\sigma \in D_\infty$ we put

$$r(\tau)[n] = r_n(\tau[n])$$

$$s(\sigma)[n] = s_n(\sigma[n]) .$$

Note that the definition of r can be reformulated in the following relation

$$j_{\infty n} \circ r = r_n \circ j'_{\infty n} .$$

From this and noting that $(s \circ i_{n\infty}, j_{\infty n} \circ r)$ and $(i'_{n\infty} \circ s_n, r_n \circ j'_{\infty n})$ are representations between D_n and IL(R') we get the relation

$$i'_{n\infty} \circ s_n = s \circ i_{n\infty} .$$

5.5 Finally we shall show that $(r \circ w', v' \circ s)$ is an isomorphism between the reflexive domains (D',ϕ',ψ') and $(D_\infty,\phi_\infty,\psi_\infty)$. Since both domains are additive we need to prove only one of the conditions in *1.2*. We show that for $f \in D_\infty \overset{c}{\to} D_\infty$ we have

$$\psi_\infty(f) = r(w'(\psi(v' \circ s \circ f \circ r \circ w'))).$$

First note that

$$\begin{aligned} j_{\infty n} &= j_{\infty n} \circ r \circ w' \circ v' \circ s \\ &= r_n \circ j'_{\infty n} \circ w' \circ v' \circ s \\ &= r_n \circ w'_n \circ v' \circ s \\ &= r_n \circ F'_n \circ F_n \circ I' \circ v' \circ s ; \end{aligned}$$

$$\begin{aligned} i_{n\infty} &= r \circ w' \circ v' \circ s \circ i_{n\infty} \\ &= r \circ w' \circ v' \circ i'_{n\infty} \circ s_n \\ &= r \circ w' \circ v'_n \circ s_n \\ &= r \circ w' \circ F' \circ F'_n \circ I'_n \circ s_n ; \end{aligned}$$

hence

$$\begin{aligned}\psi_\infty(f)[n+1] &= j_{\infty n} \circ f \circ i_{n\infty} \\ &= r_{n+1}(\psi(F_n \circ I' \circ v' \circ s \circ f \circ r \circ w' \circ F' \circ F'_n)) \\ &= r_{n+1}(w'_{n+1}(\psi(I' \circ v' \circ s \circ f \circ r \circ w' \circ F'))) \\ &= r(w'(\psi'(v' \circ s \circ f \circ r \circ w')))[n+1] \ .\end{aligned}$$

5.6 We can use this result to prove that, if in a functional system $[D_0,i_0,j_0]$ the initial representation is a closure then the induced reflexive domain $(D_\infty,\phi_\infty,\psi_\infty)$ is strict. We take here $(D,\phi,\psi) = (D_0,i_0,j_0)$ and $F_0 = I$. We have $F_0 \subseteq Q(F_0)$ so we apply the construction of *5.3*. It is easy to see that in this case the functional system is $[D_0,J_1,L_1]$ but $J_1 = \phi = i_0$ and $L_1 = \psi = j_0$ so it is actually the original system $[D_0,i_0,j_0]$ and $(D_\infty,\phi_\infty,\psi_\infty)$ is strict because it is isomorphic to the system (D',ϕ',ψ') induced by the retraction F (actually a closure) which extends I and $Q(F) = F$.

5.7 The construction from *5.1* to *5.5* assumes a given reflexive domain (D,ϕ,ψ). If F_0 is a retraction such that $F_0 \subseteq Q(F_0)$, a functional system $[D_0,i_0,j_0]$ is obtained, such that the induced domain $(D_\infty,\phi_\infty,\psi_\infty)$ is isomorphic to the domain (D',ϕ',ψ) induced by the minimal fixed of the operator Q that extends F_0. We ask now in which conditions the system $[D_0,i_0,j_0]$ is a Scott system. A nice characterization can be given in the case F_0 is a projection. Recall that in a Scott system the function j_0 is defined $j_0(f) = f(o)$ for $f \in D_0 \overset{c}{\to} D_0$; here o is the bottom of D_0, but actually is also the bottom of D in case F_0 is a projection. Note that from *5.3* it follows that

$$j_0(f) = F_0(\psi(I'_0 \circ f \circ F'_0)) \ .$$

We say that a continuous projection F_0 in a reflexive domain (D,ϕ,ψ) is *reduced* in case the following two conditions are satisfied for arbitrary $x \in D$:

a) $F_0(x) \subseteq \psi(\lambda y \cdot F_0(x))$

b) $\lambda y \cdot F_0(x) \subseteq \phi(F_0(x))$

The trivial projection $\lambda y \cdot o$ is always reduced in any domain. The example in *1.8* is reduced and so is the example in *1.9* (note that the domain is additive so $\psi(\lambda y \cdot u) = u$). The identity I is reduced only in a trivial domain containing one element.

Consider now a domain induced by a Scott system $[D_0, i_0, j_0]$. From *3.5* iii), using $P_0 = P_1 \circ P_0$, *3.5* ii), and the definition of i_0 (see *3.4*) we get $\phi(P_0(x)) = \lambda y \cdot P_0(x)$. Since the domain is strict, it follows that the projection P_0 is reduced.

5.8 THEOREM. Let F_0 be a reduced projection in the domain (D, ϕ, ψ). The following relations hold for arbitrary $x \in D$ and $f \in D_0 \overset{c}{\to} D_0$:

c) $\phi(F_0(x)) = \lambda y \cdot F_0(x)$

d) $F_0(x) \subseteq F_0(\phi(x)(o))$

e) $F_0 \subseteq Q(F_0)$

f) $F_0(\psi(I_0' \circ f \circ F_0')) = f(o)$.

Clearly c) follows from a) and b); d) follows from c) noting that F is a projection. To prove e) note that d) can be expressed in the form

$$\lambda y \cdot F_0(x) \subseteq (F_0 \circ \phi(x) \circ \lambda y \cdot o) \subseteq (F_0 \circ \phi(x) \circ F_0)$$

hence using a) we get

$$F_0(x) \subseteq \psi(F_0 \circ \phi(x) \circ F_0) \ .$$

Finally to prove f) we apply d), noting that $F_0(o) = o$, so we have

$$F_0(\psi(I_0' \circ f \circ F_0')) \subseteq F_0(f(o)) \subseteq f(o)$$

and on the other hand since $f(o) \in D_0$ we have by a) that

$$f(o) \subseteq F_0(\psi(\lambda y \cdot f(o))) \subseteq F_0(\psi(I_0' \circ f \circ F_0')) \ .$$

5.9 COROLLARY. If F_0 is a reduced projection in the domain (D,ϕ,ψ) then $F_0 \subseteq Q(F_0)$ and the associated functional system $[D_0,i_0,j_0]$ is a Scott system.

THEOREM. Let F_0 be a continuous projection in the domain (D,ϕ,ψ) such that $F_0 \subseteq Q(F_0)$ and the associated functional system $[D_0,i_0,j_0]$ is a Scott system. Then F_0 is reduced.

Proof. For $x \in D$ define $f \in D_0 \xrightarrow{c} D_0$ as $f = \lambda y \cdot F_0(x)$. Then since $j_0(f) = f(o) = F_0(x)$ we have

$$F_0(x) = F_0(\psi(\lambda y \cdot F_0(x))) \subseteq \psi(\lambda y \cdot F_0(x)) \ .$$

Also we have

$$\begin{aligned} F_0(x) &= F_0(F_0(F_0(x))) \subseteq F_0(\psi(F_0 \circ \phi(F_0(x)) \circ F_0)) \\ &= F_0(\psi(I_0' \circ F_0' \circ \phi(F_0(x)) \circ I_0' \circ F_0')) \\ &= j_0(F_0' \circ \phi(F_0(x)) \circ I_0') \\ &\subseteq F_0(\phi(F_0(x))(o)) \\ &\subseteq \phi(F_0(x))(o), \end{aligned}$$

hence $\lambda y \cdot F_0(x) \subseteq \phi(F_0(x))$.

5.10 It follows from *5.9* that the domains induced by the retractions F_0 in examples *1.8* and *1.9* are both Scott domains. They are isomorphic because in both cases the initial lattice is $\{o,u\}$.

6. ACKNOWLEDGEMENTS

The research reported in this paper has been supported by the U.S. Air Force under contract RACD F 30602-76-C-0325.

7. REFERENCES

Sanchis, L.E. (1977). Data Types as Lattices: Retractions, Closures and Projections, *R.A.I.R.O. Informatique Théorique* 11, 329-344.

Scott, D. (1972). Continuous Lattices, Proc. 1971 Dalhousie Conference, *In* Lecture Notes in Mathematics 274, Springer Verlag, New York.

Scott, D. (1976). Data Types as Lattices, *SIAM Journal of Computing* 5, 522-587.

Stoy, J.E. (1977). "Denotational Semantics", The M.I.T. Press, Cambridge, Mass., U.S.A.

Wadsworth, C.P. (1976). The Relation between Computational and Denotational Properties for Scott's D_∞ models of the Lambda-Calculus, *SIAM Journal of Computing* 5, 448-521.

Wagner, E.G. (1969). Uniformly Reflexive Structures: On the Nature of Godelizations and Relative Computability, *Transactions of the American Mathematical Society* 14, 1-41.

LAMBDA-DEFINABILITY IN THE FULL TYPE HIERARCHY

Gordon D. Plotkin

Department of Computer Science,
University of Edinburgh,
Edinburgh EH9 3JZ,
Scotland.

Dedicated to H.B. Curry on the occasion of his 80th Birthday

The completeness theorem for the first-order predicate calculus characterises provability by a semantic means which demonstrates the logical nature (validity) of theorems. Our aim here is to attempt something similar for definability in the full type hierarchy by terms of the typed λ-calculus. The obvious first try is invariance under permutations, but this fails. A first extension using hereditarily defined relations characterises λ-definability up to type level 2 (theorem 1); we do not know what happens at higher types. A second extension using a generalised kind of relation succeeds in characterising λ-definability at all types when the ground set is infinite (theorem 2). Along the way (theorem 3) we obtain a completeness theorem for $\beta\eta$-conversion. It would be interesting to investigate relative definability, to look at other models of the typed λ-calculus and to consider the untyped λ-calculus. Since the present work was completed, Statman has obtained other interesting results in the same area; see, especially, [Sta] where, among many other things, a stronger version of our theorem 3 is proved.

For information on the typed λ-calculus, consult [Hin]; here we briefly consider the necessary background material. The set of *types* is the least set containing ι and containing $(\sigma \to \tau)$ if it contains σ and τ ; $(\sigma_1,\ldots,\sigma_m,\tau)$ abbreviates $(\sigma_1 \to (\ldots.(\sigma_m \to \tau)\ldots))$ (for $m\geq 0$). The *rank* (= *order* = *level*) of a type is defined by induction on types: $r(\iota) = 0$ and $r(\sigma \to \tau) = \max(r(\tau), r(\sigma) + 1)$. We assume a denumerable set, Var_σ, of *variables* x^σ of each type σ, and put $\mathrm{Var} = \bigcup_\sigma \mathrm{Var}_\sigma$ (and often omit the superscripts on variables). The set of *terms* of the typed λ-calculus (as considered here) is the least set such that:

1. Each variable x^σ is a term of type σ.
2. If M, N are terms of types $(\sigma \to \tau),\sigma$ respectively then (MN) is a term of type τ (called a *combination*).
3. If M is a term of type τ then $(\lambda x^\sigma.M)$ is a term of type $(\sigma \to \tau)$ (called an *abstraction*).

The set of free variables of a term M is denoted by FV(M); we do not distinguish α-equivalent terms and often drop brackets (understood as associated to the left); we use $M =_{\beta,\eta} N$ to mean M and N are β,η-interconvertible.

We consider a fixed non-empty ground set D throughout and the *full type hierarchy* $\{D_\sigma\}$ is defined over D by: $D_\iota = D$ and $D_{\sigma\to\tau} = (D_\sigma \to D_\tau)$ the set of all functions from D_σ to D_τ. The set of *environments* is $\mathrm{Env} = \{\rho\colon \mathrm{Var} \to \bigcup D_\sigma \mid \forall x^\sigma.\rho x^\sigma \in D_\sigma\}$; $\rho[d/x^\sigma]$, where d is in D^σ has value ρy when $y \neq x$ and d if $y = x$. The *valuation* $[\![M]\!](\rho)$ of a term is defined by induction on terms:

1. $[\![x^\sigma]\!](\rho) = \rho x^\sigma$
2. $[\![MN]\!](\rho) = [\![M]\!](\rho)([\![N]\!](\rho))$
3. $[\![\lambda x^\sigma.M]\!](\rho)(d) = [\![M]\!](\rho[d/x^\sigma])$

If M has type σ then $[\![M]\!](\rho)$ is in D_σ. The value of $[\![M]\!](\rho)$ depends only on what values ρ assigns to the free variables of M; if M is closed we often omit reference to ρ. If $M =_{\beta,\eta} N$ then for all ρ, $[\![M]\!](\rho) = [\![N]\!](\rho)$. An element d in $\bigcup D_\sigma$ is *λ-definable* if there is a closed term M (one without free variables) such that $d = [\![M]\!]$; it is *λ-definable* from $X \subseteq \bigcup_\sigma D_\sigma$

if there is a closed term M and elements $d_1,\ldots,d_n$ of X so that $d = [\![M]\!]\,(d_1)\ldots.(d_n)$.

Because of the "logical" nature of the λ-definable elements, they should be invariant under permutations of D. Precisely, let $\pi\colon D \to D$ be a permutation and define $\pi_\sigma\colon D_\sigma \to D_\sigma$ by induction on types putting $\pi_\iota = \pi$ and for f in $D_{(\sigma \to \tau)}$,

$$\pi_{(\sigma \to \tau)}(f) = \pi_\tau \circ f \circ \pi_\sigma^{-1}.$$

Then we say an element d of D_σ is *invariant* if $\pi_\sigma(d) = d$ for all such permutations, π. It is easily shown [Läu] that all λ-definable elements are invariant but, as remarked by Läuchli, there are uncountably many invariant elements when D is infinite (even in $D_{((\iota \to \iota),\iota,\iota)}$). For example taking $\circ =_{\text{def}} (\iota,\iota,\iota)$ as a truthvalue type let tt and ff be, respectively, the terms $\lambda x.\,\lambda y.x$ and $\lambda x.\lambda y.y$. The *ground equality* EQ: $D_{(\iota,\iota,\circ)}$ is invariant but not λ-definable if $|D| > 1$, where EQ(d)(d') is $[\![tt]\!]$ if $d = d'$ and $[\![ff]\!]$ otherwise.

M. Gordon proposed, as a possible remedy, that relations rather than just permutations should be extended to higher types; this idea was also used by Howard for defining his hereditarily majorisable functionals [Tro]. Specifically suppose $R \subseteq D^\kappa$ (κ any ordinal) and define $R_\sigma \subseteq D_\sigma^\kappa$ by induction on types putting $R_\iota = R$ and for f in $D^\kappa_{(\sigma \to \tau)}$,

$$R_{(\sigma \to \tau)}(f) \equiv \forall\, d \in D_\sigma^\kappa\,.(R_\sigma(d) \supset R_\tau(f(d))).$$

Here f(d) is $\langle f_\lambda(d_\lambda)\rangle_{\lambda<\kappa}$. Then an element d of D_σ *satisfies* R if $R_\sigma(\langle d\rangle_{\lambda<\kappa})$ holds.

PROPOSITION 1. Suppose $R \subseteq D^\kappa$. Then every λ-definable element satisfies R and every element λ-definable from a set of elements satisfying R itself satisfies R.

Proof. We demonstrate by induction on terms M that:

$$\forall \rho \in \mathrm{Env}^\kappa.(\forall x^\tau \in FV(M).R_\tau(\rho(x^\tau))) \supset R_\sigma([\![M]\!](\rho))$$

where σ is the type of M. Here $\rho(x^\tau)$ is $\langle \rho_\lambda(x^\tau)\rangle_{\lambda<\kappa}$ and $[\![M]\!](\rho)$

is $<[\![M]\!](\rho_\lambda)>_{\lambda<\kappa}$.

In case M is a variable, x^τ, $[\![M]\!](\rho) = \rho(x^\tau)$ which satisfies R_σ by assumption. In case M is a combination (M_1M_2), $[\![M_1M_2]\!](\rho) = [\![M_1]\!](\rho)([\![M_2]\!](\rho))$ and this satisfies R_σ by the definition of $R_{(\sigma\to\tau)}$ using the induction hypothesis for M_1 and M_2. In case M is an abstraction $(\lambda x^\sigma.M_1)$ let d satisfy R_σ. Then $[\![\lambda x^\sigma.M_1]\!](\rho)(d) = [\![M_1]\!](\rho')$, where $\rho' = <\rho_\lambda[d_\lambda/x^\sigma]>$, and we can apply the induction hypothesis to M_1, concluding the inductive proof. The first part of the proposition then follows applying the above to closed M. The second part is then immediate. ⊠

As an example of non-definability suppose 0,1 are distinct elements of D and take R = {<0,0>,<0,1>,<1,0>}. Then $R_o([\![tt]\!],[\![ff]\!])$ does not hold as $R_\iota(1,0)$ and $R_\iota(0,1)$ but not $R_\iota([\![tt]\!](1)(0),[\![ff]\!](0)(1))$; so EQ does not satisfy R as $R_\iota(0,0)$ and $R_\iota(0,1)$ but not $R_\iota(EQ(0)(0),EQ(0)(1))$. This shows EQ is not λ-definable when $|D| > 1$.

As an example of non-relative definability consider the "universal quantification" functional, $F: D_{(\iota\to o)}\to D_o$ where:

$$F(f) = \begin{cases} [\![tt]\!] \text{ (if } f(d) = [\![tt]\!] \text{ for all d in D)} \\ \\ [\![ff]\!] \text{ (otherwise)} \end{cases}$$

Now F is invariant but not λ-definable from EQ if $|D| > 2$. For let R = {<0,0>,<1,1>} where $0 \neq 1$. Then EQ satisfies R but with $f = [\![\lambda x^\iota.tt]\!]$ and $g(d) = [\![tt]\!]$ if d is 0 or 1 and $g(d) = [\![ff]\!]$ otherwise we have $R_{(\iota\to o)}(f,g)$ but not $R_o(Ff,Fg)$.

(Incidentally F is λ-definable from EQ if $|D| \leq 2$.)

THEOREM 1. Suppose $r(\sigma) \leq 2$. Then if D is infinite and $f \in D_\sigma$ satisfies every $R \subseteq D^2$, f is λ-definable.

Proof. We just consider two cases to give the idea without too much detail. The first case is $\sigma = (\iota,\iota,\iota)$. Let d,e,0,1 be elements of D with $0 \neq 1$ and put R = {<d,0>,<e,1>}. Then

R (fde,f01) and so for all d,e in D either fde = d and f01 = 0 or else fde = e and f01 = 1. So either f01 = 0 or f01 = 1 ; in the first case f = [[tt]] , in the second case f = [[ff]].

The second case is $\sigma = ((\iota,\iota),\iota,\iota)$. We can suppose $\omega \subseteq D_\iota$ and choose $s\colon D_\iota \to D_\iota$ to act as the successor on the integers. For g in $D_{(\iota\to\iota)}$ and d in D let $R = \{\langle g^n d, s^n 0\rangle \mid n \geq 0\}$. As $R_\iota(d,0)$ and $R_{(\iota\to\iota)}(g,s)$ and f satisfies R there is an n such that $fgd = g^n d$ and $fs0 = s^n 0$. As the $s^n(0)$ are all different we see that for some $n, f = [[\lambda x.\ \lambda y.x^n(y)]]$, in an obvious notation. ⊠

We believe this theorem holds without the restriction on D; we know nothing about what happens at higher types.

To proceed further we try to interpret the implication sign in the definition of the $R_{\sigma\to\tau}$ in an intuitionistic way, hoping thereby to make any f satisfying $R_{\sigma\to\tau}$ more likely to be constructive and therefore λ-definable. In order to do this we use Kripke's ideas [Kri] on the interpretation of intuitionistic logic.

Specifically suppose $\langle W,\leq\rangle$ is a quasiorder (i.e. a reflexive transitive relation), where we interpret W as a set of worlds and $\leqslant$ as an alternativeness relation over W and suppose too that $R \subseteq D^\kappa \times W$ is a relation such that for all d in D^κ, w in W:

$$R(d,w) \supset \forall w' \geq w.\ R(d,w')$$

We call such an R an I-*relation* and now define $R_\sigma \subseteq D^\kappa_\sigma \times W$ by putting $R_\iota = R$ and for any f in $D^\kappa_{\sigma\to\tau}$ and w in W:

$$R_{\sigma\to\tau}(f,w) \equiv \forall w' \geq w.\ \forall d \in D^\kappa_\sigma.(R_\sigma(d,w') \supset R_\sigma(fd,w')).$$

Then an element d of D_σ I-*satisfies* R if $R(\langle d\rangle_{\lambda<\kappa},w)$ holds for all w in W. It is clear (taking W to be a singleton) how this generalises the previous idea of satisfaction.

LEMMA 1. With R_σ as above and for any d in D^κ_σ, w in W:

$$R(d,w) \supset \forall\ w' \geq w.R(d,w')$$

Proof. The proof is an easy induction on σ , using the transitivity of $\leq$. ⊠

PROPOSITION 2 Suppose $R \subseteq D^\kappa \times W$ is an I-relation. Then every λ-definable element I-satisfies R and every element λ-definable from a set of elements I-satisfying R itself I-satisfies R.

Proof. We demonstrate by induction on terms M that:

$$\forall w \in W . \forall \rho \in Env^\kappa_\cdot((\forall x^\tau \in FV(M) . R_\tau(\rho\, x^\tau, w)) \supset R_\sigma([\![M]\!](\rho), w))$$

where σ is the type of M. The proposition follows.

The cases where M is a variable or a combination are easy - the latter uses the reflexivity of $\leq$. In case M is an abstraction $(\lambda x^\sigma . M_1)$ suppose $R_\sigma(d, w')$ where $w' \geq w$. Then $[\![\lambda x^\sigma . M_1]\!](\rho)(d) = [\![M_1]\!](\rho')$ where $\rho' = \langle \rho_\lambda [d_\lambda / x^\sigma] \rangle_{\lambda < \kappa}$. Now by lemma 1 and the assumption on d, $R_\tau(\rho' x^\tau, w')$ holds for all x^τ in $FV(M_1)$ and we can apply the induction hypothesis to M_1. ⊠

THEOREM 2 (Completeness Theorem) Suppose D is infinite. Then an element d of D_σ is λ-definable iff it I-satisfies every I-relation $R \subseteq D^3 \times W$.

We do not know if the restriction on D can be dropped or if 3 can be reduced to 2 - it cannot be reduced to 1 because, for example, if $D = \omega$ and $F: D_{((\iota \to \iota), \iota, \iota)}$ is defined by:

$$F(g)(d) = g^{g(d)}(d)$$

then it I-satisfies every I-relation $R \subseteq D \times W$ but is not λ-definable.

The consistency half (definability implies I-satisfaction) of theorem 2 is given by proposition 2; the rest of this paper is devoted to proving the other half. The intention is to construct a suitable W and R. We begin with some notation for vectors. If $d = \langle d_1, \ldots, d_m \rangle$ in $\prod_{1 \leq i \leq m} D_{\sigma_i}$ is a vector (= finite sequence) of elements and f is in $D_{(\sigma_1, \ldots, \sigma_m, \tau)}$ then fd is $fd_1 \ldots d_m$ $(m \geq 0)$; if $v = \langle x_1, \ldots, x_m \rangle$ is a vector of variables then v is *non-repeating* if the x_i are all different;

for ρ in Env, ρv is $\langle \rho x_1, \ldots, \rho x_m \rangle$; for a term M, Mv is $Mx_1 \ldots x_m$ and $\lambda v.M$ is $\lambda x_1 \ldots \lambda x_m.M$. A property P holds for *essentially all* vectors of variables if there is a finite set $F \subseteq$ Var such that whenever no component of v is in F then P(v) holds. Concatenation of vectors is indicated by juxtaposition; note the essential unambiguity of the notation Mvv'.

From now on we assume D is infinite. Let $||\cdot||$ be a map from terms of type ι to D such that:

$$|| M || = || N || \text{ iff } M =_{\beta,\eta} N.$$

Define $d \underset{\rho}{\sim} M$ for any environment ρ, element d of D_σ and term M of type σ (where $\sigma = (\sigma_1, \ldots, \sigma_m, \iota))$ by:

$$d \underset{\rho}{\sim} M \equiv \text{For essentially all non-repeating } v \text{ in } \prod_{1\le i\le m} \mathrm{Var}_{\sigma_i}, \quad d(\rho v) = ||Mv||.$$

Note that this relation depends only on the value of ρ at variables of types strictly smaller than σ. Also if $d \underset{\rho}{\sim} M_i$ $(i = 0,1)$ then $M_0 =_{\beta,\eta} M_1$. Now for d in D_σ let $M(d,\rho)$ be a term of type σ such that $d \sim_\rho M(d,\rho)$ if one exists, and an arbitrary term (say x^σ) of that type otherwise.

Now we can define an environment ρ_s by putting for x^σ where $\sigma = (\sigma_1, \ldots, \sigma_m, \iota)$, and d in $\prod_{1\le i\le m} D_{\sigma_i}$

$$\rho_s(x^\sigma)(d) = ||x^\sigma M(d_1,\rho_s) \ldots M(d_m,\rho_s)||$$

The above remarks show, by structural induction on σ, that this is a good definition.

LEMMA 2. For all terms M $[\![M]\!](\rho_s) \underset{\rho_s}{\sim} M$.

Proof. Without loss of generality we can just prove the proposition by induction on terms in long $\beta\eta$-normal form, see [Jen]. So assume M has the form $\lambda x_1 \ldots \lambda x_m . xM_1 \ldots M_n$ where $xM_1 \ldots M_n$ has type ι and the M_j $(1\le j\le n)$ are in long $\beta\eta$-normal form. To show $[\![M]\!](\rho_s) \underset{\rho_s}{\sim} M$ it is enough to consider only vectors, v, in $\prod_{1\leq i\leq m} Var_{\sigma_i}$ none of whose components are free variables of M. By taking α-conversions of M we see that the case $v = \langle x_1,\ldots,x_m\rangle$ is typical. So we calculate:

$$\begin{aligned}
[\![M]\!](\rho_s)(\rho_s x_1)..(\rho_s x_m) &= [\![M_1\ldots.x_m]\!]\,(\rho_s)\\
&= [\![xM_1\ldots..M_n]\!](\rho_s)\\
&= \rho_s(x)([\![M_1]\!](\rho_s)\ldots([\![M_n]\!](\rho_s)\\
&= ||xM([\![M_1]\!](\rho_s),\rho_s)\ldots M([\![M_n]\!](\rho_s),\rho_s)||\\
&= ||xM_1\ldots M_n||\ \text{(by induction}
\end{aligned}$$

hypothesis, the definition of $||\cdot||$ and the above remark on $\underset{\rho}{\sim}$)

$$= ||Mx_1\ldots..x_m||. \boxtimes$$

This gives a completeness theorem for βη-conversion (cf. [Böh]). From now on we generally omit the reference to ρ_s in $[\![M]\!](\rho_s)$.

THEOREM 3. For any term M of type ι , $[\![M]\!] = ||M||$. Further for any terms M and N of the same type σ:

$$M =_{\beta,\eta} N \text{ iff } \forall\rho \in Env.[\![M]\!](\rho) = [\![N]\!]\,(\rho) \text{ iff } [\![M]\!]\,(\rho_s) = [\![N]\!](\rho_s)$$

Proof. The first part is immediate from lemma 2. For the second part the implications from left to right are well-known; for the converses suppose $[\![M]\!](\rho_s) = [\![N]\!](\rho_s)$, $\sigma = (\sigma_1,\ldots.,\sigma_m,\iota)$ and let x_i be a variable of type σ_i not free in either M or N ($1\leq i\leq m$). Then

$$\begin{aligned}
||Mx_1\ldots.x_m|| &= [\![Mx_1\ldots x_m]\!]\ \text{(by the first part)}\\
&= [\![Nx_1\ldots.x_m]\!]\ \text{(by assumption)}\\
&= ||Nx_1\ldots.x_m||\text{(by the first part).}
\end{aligned}$$

So $Mx_1\ldots x_m =_{\beta,\eta} Nx_1\ldots.x_m$ and so taking the x_i to be all different we find that $M =_{\beta,\eta} N$. ⊠

The second part of this theorem fails if D is finite; for example, there are only finitely many elements in $D_{((\iota\to\iota),\iota,\iota)}$ but infinitely many closed normal terms of that type.

We are now in a position to define $\langle W,\leq\rangle$ and R. First:

$W = \{\langle d,v,\bar{v}\rangle \mid d$ is a vector of elements of $\bigcup_\sigma D_\sigma$, v and $\bar{v}$ are non-repeating vectors of variables, all three have the same length and corresponding components have the same type$\}$.
For i = 1,2,3 the ith component of any w in W is written as w_i; for any w,w^+ in w the concatenation ww', is defined to be the componentwise concatenation $\langle w_1w_1^+,w_2w_2^+,w_3w_3^+\rangle$ and we define $w\leq w'$ to mean $w' = ww^+$ for some w^+. Finally, $R \subseteq D^3 \times W$ is defined by:

$R(d,w) \equiv$ There is a closed term M such that $d_1 = [\![M]\!](w_1)$, and $d_i = [\![Mw_i]\!]$ for i=2,3.

Clearly $\leq$ is reflexive and transitive and R is an I-relation.

A term, M, is <u>head-λ -free</u> iff it has the form $xM_1\ldots M_k$.

LEMMA 3. Let M be a head-λ-free term and let d,d' be elements of D_σ. Then if $d\ \rho_s(v) = d'\ \rho_s(v)$ for essentially all non-repeating vectors, v, of variables of the appropriate types such that $d_\rho(v)$ has type ι, then $[\![M]\!](d) = [\![M]\!](d')$.

Proof. As $M(d,\rho_s) =_{\beta,\eta} M(d',\rho_s)$ by assumption, the conclusion is immediate from the definition of ρ_s. ⊠

LEMMA 4. 1. Suppose $R_\sigma(f,g,h,w)$ holds where $w = \langle d,v,\bar{v}\rangle$. Then there is a closed term M such that $f = [\![M]\!]d$, $g(\rho v^+) = [\![Mvv^+]\!]$ and $h(\rho\bar{v}^+) = [\![M\bar{v}\bar{v}^+]\!]$ whenever vv^+, $\bar{v}\bar{v}^+$ are non-repeating vectors of variables of the appropriate type such that $[\![Mvv^+]\!]$ is of type ι.

2. Suppose f,g,h are of type σ and w is a world. If g,h are denotations of head-λ-free terms and there is a closed term M such that $f = [\![M]\!]w_1$, $g = [\![Mw_2]\!]$ and $h = [\![Mw_3]\!]$, then $R_\sigma(f,g,h,w)$ holds.

Proof. Both parts are proved together by induction on σ.
1. For ι the result is immediate from the definition of R.
For the case $\sigma \to \tau$ suppose $R_{\sigma\to\tau}(f,g,h,w)$ holds where $w = \langle d,v,\bar{v}\rangle$. Let $w' = w\langle e,x,\bar{x}\rangle$ be a world with e in D_σ. Then by induction hypothesis, using part 2 we see that $R_\sigma(e,\rho_s(x),\rho_s(\bar{x}),w')$ (take

$M = \lambda v.\lambda x.x$). Therefore, by the definition of $R_{\sigma \to \tau}$ we have $R_\tau(fe, g\rho_s(x), h\rho_s(\bar{x}), w')$. Therefore by induction hypothesis, using part 1, there is a closed term M such that $fe = [\![M]\!]de, (g\rho_s(x))\rho_s(v^+) = [\![Mvxv^+]\!]$ and $(h\rho_s(\bar{x}))\rho_s(\bar{v}^+) = [\![M\overline{vxv}^+]\!]$ whenever vxv^+, $\overline{vxv}^+$ are non-repeating vectors of variables of the appropriate type such that $[\![Mvxv^+]\!]$ is of type ι.

Clearly M may depend on e, x and $\bar{x}$. As $g\rho_s(x)\rho_s(v^+) = [\![Mvxv^+]\!]$ and neither side of the equation mentions e or $\bar{x}$ and as vxv^+ is non-repeating and M is closed it follows by Theorem 3 that M is independent of e or $\bar{x}$; similarly, using the equation for h it is independent of x. Therefore we have a closed term M such that $fe = [\![M]\!]\, de$, $g(\rho_s x)(\rho_s v^+) = [\![Mvxv^+]\!]$, $h(\rho_s\bar{x})(\rho_s\bar{v}^+) = [\![M\overline{vxv}^+]\!]$ whenever e is in D_σ and vxv^+ and $\overline{vxv}^+$ are non-repeating vectors of variables of the appropriate type such that $[\![Mvxv^+]\!]$ is of type ι, which finishes the proof of part 1.

2. For ι the result is immediate from the definition of R. For the case $(\sigma \to \tau)$ suppose f,g,h are of type $(\sigma \to \tau)$, $w = \langle d, v, \bar{v}\rangle$ is a world, that g,h are denotations of head-λ-free terms and that there is a closed term M such that $f = [\![M]\!]d$, $g = [\![Mv]\!]$ and $h = [\![M\bar{v}]\!]$. Let $w' = w\langle d^+, v^+, \bar{v}^+\rangle$ be a world and suppose that $R_\sigma(e,a,b,w')$. Then by induction hypothesis using part 1, there is a closed term M_1 such that $e = [\![M_1]\!]dd^+$, $a(\rho(v^{++})) = [\![M_1 vv^+v^{++}]\!]$ and $b(\rho(\bar{v}^{++})) = [\![M_1\overline{vv^+v^{++}}]\!]$ whenever vv^+v^{++}, $\overline{vv^+v^{++}}$ are non-repeating vectors of variables of the appropriate type such that $[\![M_1vv^+v^{++}]\!]$ is of type ι.

Now we have, $f(e) = [\![M]\!]d([\![M_1]\!]dd^+) = [\![M_2]\!]dd^+$, where $M_2 = \lambda v.\lambda v^+.Mv(M_1vv^+)$. Since $a(\rho_s(v^{++})) = [\![M_1vv^+]\!]\rho_s(v^{++})$ for essentially all non-repeating vectors of variables of the appropriate types such that $a(\rho_s(v^{++}))$ has type ι and since g is the denotation of a head-λ-free term, we can apply lemma 3 to see that $g(a) = g[\![M_1vv^+]\!]$. Therefore $g(a) = [\![Mv]\!]\,[\![M_1vv^+]\!] = [\![M_2vv^+]\!]$ and similarly $h(b) = [\![M_2vv^+]\!]$. As $g(d)$ and $h(b)$ are clearly, therefore, denotations of head-λ-free terms and as we have already shown that $f(e) = [[M_2]]\, dd^+$ it follows by the induction hypotheses, using part 2, that $R_\tau(fe, ga, hb, w')$, showing $R_{\sigma \to \tau}(f,g,h,w)$ and concluding the inductive proof. ☒

The proof of the rest of theorem 2 is now immediate. For suppose an element d in D_σ I-satisfies every I-relation $R \subseteq D^3 \times W$. Then with R as defined above and taking w_o as the world all of whose components are empty we have $R_\sigma(d,d,d,w_o)$. Then by lemma 4.1 there is a closed term M such that $d = [[M]]$.

ACKNOWLEDGEMENTS

I would like to thank Mike Gordon for his help and encouragement. The work was carried out with the aid of an SRC research grant.

REFERENCES

Böhm, C. (1968). Alcune proprieta della forme λ-κ-normali del β-η-calcolo, *Publicazioni dell'instituto per le applicazioni del calcolo (CNR-Roma)* No. 696.

Hindley, J.R., Lercher, B. and Seldin, J.P. (1972). Introduction to Combinatory Logic. London Math. Soc. Lecture Note Series No. 7 . Cambridge University Press, Cambridge.

Jensen, D.C. and Pietrzykowski, T. (1976). Mechanising ω-order type theory through unification, *Theoretical Computer Science* Vol. 3, 123-171.

Kripke, S.A. (1965). Semantical analysis of intuitionistic logic I. *In* "Formal systems and recursive functions." Proc. of the 8th Logic Colloquium, Oxford ,1963. North-Holland, Amsterdam.

Läuchli, H. (1970). An abstract notion of realizability for which intuitionistic predicate calculus is complete. *In* "Intuitionism and Proof Theory". North-Holland, Amsterdam.

Statman, R. (1978). Completeness, Invariance and λ-definability.

Troelstra, A.S. (1973). Metamathematical Investigation of Intuitionistic Arithmetic and Analysis. Lecture Notes in Mathematics Vol. 344, pp. 123-124. Springer-Verlag, Berlin.

FROM λ-CALCULUS TO

CARTESIAN CLOSED CATEGORIES

J. Lambek

*Mathematics Department, McGill University,
Montreal, P.Q. H3A 2K6, Canada.*

Dedicated to H.B. Curry on the occasion of his 80th Birthday.

Haskell Curry may be surprised to hear that he has spent a lifetime doing fundamental work in category theory. The purpose of this account is to convince categorists that Cartesian closed categories (Eilenberg and Kelly, 1966) have been anticipated by logicians (Curry, 1930) by many years and, conversely, to persuade logicians that combinatory logic may benefit from being phrased in categorical language.

I have attempted to tell this story twice before (1972, 1974), but am not entirely satisfied with these earlier accounts. The present exposition is essentially my unscheduled talk at the 1977 Durham Symposium on applications of sheaf theory to logic, algebra and analysis.

I regret that limitations of space do not permit a discussion of illative combinatory logic (Curry and Feys, 1958) or combinatory type theory (Church, 1940) and applications thereof to the construction of free toposes.

Let me confess at once that I am not a historical scholar and that I have taken some liberties with the original material. Thus, I have taken the opportunity to present the early discoveries of combinatory logic in the language of universal algebra.

Our story begins in 1924, when Schönfinkel studied what would now be called an albegra $A = (|A|, {}^{\ulcorner}, I, S, K)$ consisting of a set $|A|$ equipped with a binary operation ${}^{\ulcorner}$ and constants I, S and K. These were to satisfy the following identities:

(1) $I^{\ulcorner}a = a,$

(2) $(K^{\ulcorner}a)^{\ulcorner}b = a,$

(3) $((S^{\ulcorner}f)^{\ulcorner}g)^{\ulcorner}c = (f^{\ulcorner}c)^{\ulcorner}(g^{\ulcorner}c),$

for all elements a, b, c, f and g of $|A|$. Actually, Schönfinkel did not employ the language of universal algebra, and he defined I in terms of K and S, but of this we shall speak later. His main result would now be stated as follows.

PROPOSITION 1. Every polynomial $\varphi(x)$ over a Schönfinkel algebra A can be written in the form $f^{\ulcorner}x$, where $f \in |A|$.

Polynomials are of course formed as words in an indeterminate x and are subject to the same three identities. More precisely, equality $\underset{x}{=}$ between polynomials is the smallest equivalence relation $\equiv$ between words in x which has the substitution property

$$(0_x) \qquad \frac{\varphi(x) \equiv \psi(x) \quad \alpha(x) \equiv \beta(x)}{\varphi(x)^{\ulcorner}\alpha(x) \equiv \psi(x)^{\ulcorner}\beta(x)}$$

and which satisfies

$$(1_x) \qquad I{}^{\ulcorner}\alpha(x) \equiv \alpha(x),$$

$$(2_x) \qquad (K{}^{\ulcorner}\alpha(x)){}^{\ulcorner}\beta(x) \equiv \alpha(x),$$

$$(3_x) \qquad ((S{}^{\ulcorner}\varphi(x)){}^{\ulcorner}\psi(x)){}^{\ulcorner}\gamma(x) \equiv (\varphi(x){}^{\ulcorner}\gamma(x)){}^{\ulcorner}(\psi(x){}^{\ulcorner}\gamma(x)).$$

Alternatively, one may regard the polynomial $\varphi(x)$ as an element of the Schönfinkel algebra $\mathcal{A}[x]$, which comes equipped with an element x and with a homomorphism $h_x\colon \mathcal{A} \to \mathcal{A}[x]$ with the usual universal property: for every algebra $\mathcal{B}$, every homomorphism $f\colon \mathcal{A} \to \mathcal{B}$ and every element $b \in |\mathcal{B}|$, there exists a unique homomorphism $f'\colon \mathcal{A}[x] \to \mathcal{B}$ such that $f'h_x = f$ and $f'(x) = b$. In the special case when $\mathcal{B} = \mathcal{A}$ and f is the identity homomorphism $\mathcal{A} \to \mathcal{A}$, f' is the <u>substitution</u> homomorphism which allows us to replace x in any polynomial by b. Similarly, when $\mathcal{B} = \mathcal{A}[x]$ and $f = h_x$, f' allows us to replace x by a polynomial $\beta(x)$.

The proof of Proposition 1 is remarkably simple. It proceeds by induction on the length of the word $\varphi(x)$, which must be either x, or a constant or of the form $\psi(x){}^{\ulcorner}\chi(x)$ not constant, in which last case we may assume by inductional assumption that $\psi(x) \underset{x}{=} g{}^{\ulcorner}x$ and $\chi(x) \underset{x}{=} h{}^{\ulcorner}x$. In the three cases we have

$$\varphi(x) \underset{x}{=} I{}^{\ulcorner}x,$$

$$\varphi(x) \underset{x}{=} (K{}^{\ulcorner}a){}^{\ulcorner}x,$$

$$\varphi(x) \underset{x}{=} (g{}^{\ulcorner}x){}^{\ulcorner}(h{}^{\ulcorner}x) \underset{x}{=} ((S{}^{\ulcorner}g){}^{\ulcorner}h){}^{\ulcorner}x,$$

respectively.

This proof also yields an algorithm for converting every polynomial into the form $f{}^{\prime}x$. For example, one easily calculates

$$x{}^{\prime}x \underset{x}{=} ((S{}^{\prime}I){}^{\prime}I){}^{\prime}x.$$

In 1930, Schönfinkel's result was rediscovered by Curry. However, Curry was interested in imposing an additional requirement:

(4) If $f{}^{\prime}x \underset{x}{=} g{}^{\prime}x$ in $A[x]$ then $f = g$ in A.

For example, from

$$((S{}^{\prime}K){}^{\prime}I){}^{\prime}x \underset{x}{=} (K{}^{\prime}x){}^{\prime}(I{}^{\prime}x) \underset{x}{=} x \underset{x}{=} I{}^{\prime}x$$

one could use (4) to obtain $(S{}^{\prime}K){}^{\prime}I = I$, which equation cannot be derived from (1) to (3) alone. In the same way, one could deduce that $(S{}^{\prime}K){}^{\prime}K = I$. In fact, Schönfinkel originally defined I by this equation. For reasons that will become clear later, we shall not follow him in this application of Occam's razor.

While (4) does not have the form of an identity, by which I mean an equation prefixed by universal quantifiers, Curry discovered that it could be replaced by a finite number of equations. These were later simplified, and Rosenbloom lists four, one of which reads

$$(S{}^{\prime}((S{}^{\prime}(K{}^{\prime}S)){}^{\prime}K)){}^{\prime}(K{}^{\prime}I) = I.$$

The reader will forgive us for not copying out the other three!

By a <u>Curry algebra</u> we shall mean a Schönfinkel algebra subject to certain additional equations or identities whose conjunction is equivalent to (4). Curry's result may then be formulated thus:

PROPOSITION 2. Over a Curry algebra A every polynomial $\varphi(x)$ may be uniquely written in the form $f^{\ulcorner}x$ with $f \in |A|$.

This property of Curry algebras is called <u>functional completeness</u>. It is an immediate consequence of the equivalence of (4) with the conjunction of the above mentioned four equations. For a proof we could refer the reader to the book by Rosenbloom. However, we prefer to give another proof, in the course of which we shall discover five equations whose conjunction is equivalent to (4).

Proof. Let $\lambda_x\varphi(x)$ be defined by induction on the length of the word $\varphi(x)$ thus:

(i) $\lambda_x x = I$;

(ii) $\lambda_x a = K^{\ulcorner}a$, when a is a constant;

(iii) $\lambda_x(\psi(x)^{\ulcorner}\chi(x)) = (S^{\ulcorner}\lambda_x\psi(x))^{\ulcorner}\lambda_x\chi(x)$, when $\psi(x)$ and $\chi(x)$ are not both constant.

We shall prove below that the restriction on (iii) is not necessary. In view of the above proof of Proposition 1, we have

$$\varphi(x) \underset{x}{=} \lambda_x\varphi(x)^{\ulcorner}x,$$

so the existence of f with $\varphi(x) \underset{x}{=} f^{\ulcorner}x$ is assured. It remains to prove its uniqueness. First, we claim that

(*) $\varphi(x) \underset{x}{=} \psi(x)$ implies $\lambda_x\varphi(x) = \lambda_x\psi(x)$,

so that $\lambda_x\varphi(x)$ depends not just on the word $\varphi(x)$ but on the polynomial $\varphi(x)$, that is, the word modulo the equivalence relation $\underset{x}{=}$.

To prove (*), we write $\varphi(x) \equiv \psi(x)$ for $\lambda_x\varphi(x) = \lambda_x\psi(x)$. It is easily seen that $\equiv$ is an equivalence relation between words which satisfies (0_x), that is, a congruence relation. If we make sure that $\equiv$ also satisfies (1_x) to (3_x), it will follow that $\equiv$ contains $\underset{x}{=}$, and this is what (*) asserts.

To say that $\equiv$ satisfies (1_x) means that

$$\lambda_x(I^{\ulcorner}\alpha(x)) = \lambda_x\alpha(x).$$

Writing a for $\lambda_x\alpha(x)$, we may rewrite this, in view of the unrestricted (iii), as

$$(S^{\ulcorner}(K^{\ulcorner}I))^{\ulcorner}a = a,$$

an easy consequence of

$$(4.1) \qquad S^{\ulcorner}(K^{\ulcorner}I) = I,$$

which itself has already been derived from (4).

In the same manner we may obtain consequences (4.2) and (4.3) of (4) which imply (2_x) and (3_x) respectively. We shall not bother to spell them out.

We now turn to the uniqueness of f in Proposition 2. Suppose also $g^{\ulcorner}x \underset{x}{=} \varphi(x)$, we claim that $g = \lambda_x\varphi(x)$. Now, by (*), $\lambda_x(g^{\ulcorner}x) = \lambda_x\varphi(x)$, so it suffices to prove that $\lambda_x(g^{\ulcorner}x) = g$.

Now a small calculation shows that $\lambda_x(g^{\ulcorner}x) = (S^{\ulcorner}(K^{\ulcorner}g))^{\ulcorner}I$, so we require the identity

$$(S^{\ulcorner}(K^{\ulcorner}g))^{\ulcorner}I = g$$

for all g, which is an easy consequence of (4).

This identity must remain valid if we adjoin an indeterminate y to the algebra, so we have

$$(S^{\ulcorner}(K^{\ulcorner}y))^{\ulcorner}I \underset{y}{=} y.$$

We may therefore replace the required identity by the equation

$$\lambda_y((S^{\ulcorner}(K^{\ulcorner}y))^{\ulcorner}I = I. \tag{4.4}$$

Of course the λ may be eliminated from this using (i) to (iii).

It remains to show the validity of (iii) when $\psi(x)^{\ulcorner}\chi(x)$ is constant, say $b^{\ulcorner}c$. So we want to show that

$$\lambda_x(b^{\ulcorner}c) = (S^{\ulcorner}(K^{\ulcorner}b))^{\ulcorner}(K^{\ulcorner}c).$$

But, by (ii), $\lambda_x(b^{\ulcorner}c) = K^{\ulcorner}(b^{\ulcorner}c)$, so we are led to stipulate the identity

$$(S^{\ulcorner}(K^{\ulcorner}b))^{\ulcorner}(K^{\ulcorner}c) = K^{\ulcorner}(b^{\ulcorner}c)$$

for all b and c. Again, this is an easy consequence of (4). By the same argument as above, we may replace the stipulated identity by the equation

$$\lambda_x\lambda_y((S^{\ulcorner}(K^{\ulcorner}x))^{\ulcorner}(K^{\ulcorner}y)) = \lambda_x\lambda_y(K^{\ulcorner}(x^{\ulcorner}y)). \tag{4.5}$$

The proof of Proposition 2 is now complete, provided we adopt (4.1) to (4.5) as the five equations which a Curry algebra must satisfy in addition to the identities (1) to (3).

From now on we shall write $\lambda_x\varphi(x)$ for the unique f corresponding to $\varphi(x)$ by Proposition 2, as we did in the proof. The properties of the new symbol λ_x are embodied in the

λ-calculus of Church (1932). The equivalence of the systems of Curry and Church (λK-calculus, 1941) are summed up as follows.

PROPOSITION 3. The identities and equations of Curry algebras imply and are implied by the following:[1)]

(1) $\quad I = \lambda_x x,$

(2) $\quad K = \lambda_x \lambda_y x,$

(3) $\quad S = \lambda_u \lambda_v \lambda_z ((u^{\ulcorner} z)^{\ulcorner}(v^{\ulcorner} z)),$

(4) $\quad \lambda_x (f^{\ulcorner} x) = f,$

(5) $\quad (\lambda_x \varphi(x))^{\ulcorner} a = \varphi(a).$

The proof is almost straightforward. We shall only explain why (5) holds for Curry algebras. By Proposition 2, we have $f^{\ulcorner} x \underset{x}{=} \varphi(x)$, and we want to deduce from this that $f^{\ulcorner} a = \varphi(a)$. By the universal property of $A[x]$, there exists a unique homomorphism $h' : A[x] \to A$ such that $h'h_x = h_x$ and $h'(x) = a$. This is of course the substitution homomorphism which replaces x by a, hence yields the required equation from $f^{\ulcorner} x \underset{x}{=} \varphi(x)$.

To recapture the traditional terminology, let us mention that the theory of Curry algebras is called <u>combinatory logic</u>. Proposition 2 may then be compressed into the slogan:

$$\text{combinatory logic} = \lambda\text{-calculus}.$$

Incidentally, <u>combinators</u> are the canonical elements of Curry (or Schönfinkel) algebras, that is, the elements of the free Curry algebra generated by the empty set.

Both Schönfinkel and Curry had intended to use combinatory logic for the foundations of mathematics. An obstacle arose in the following result.

PROPOSITION 4. In a Curry algebra every element f has a "fixpoint" a such that $f^{\ulcorner}a = a$.

Proof. Let $b = \lambda_x(f^{\ulcorner}(x^{\ulcorner}x))$ and put $a = b^{\ulcorner}b$. Then

$$f^{\ulcorner}a = f^{\ulcorner}(b^{\ulcorner}b) = b^{\ulcorner}b = a.$$

If f is negation, usually denoted by $\neg$, we have $\neg a = a$, so a cannot be a proposition. We must therefore distinguish between propositions and other entities; but even this distinction does not prevent Russell's paradox from raising its head (e.g. Curry and Feys, 1958).

It is of course well-known that Russell's paradox may be avoided by introducing types. In the following exposition of typed combinatory logic I shall follow Curry and Feys in principle, even though I shall reject their permissiveness in allowing symbols with ambiguous types.

First of all we replace a Curry algebra by a many-sorted algebra, whose elements, which we call <u>entities</u>, may belong to different sorts, which we call <u>types</u>. If A and B are types, so is B^A, the type of all "functions" from B to A. For typographical reasons, we write $B \Leftarrow A$ in place of B^A. The binary operation symbol $^{\ulcorner}$ may not be placed between entities indiscriminately, but is subject to the following rule:

$$(0) \qquad \frac{a \in A \qquad\qquad f \in B \Leftarrow A}{f^{\ulcorner}a \in B},$$

meaning that, if a is of type A and f of type $B \Leftarrow A$, then $f^{\ulcorner}a$ is of type B.[2)]

In place of the three constants I, K and S we adopt three families of constants:

(1) $I_A \in A \Leftarrow A$ such that $I_A{}^{\ulcorner}a = a$;

(2) $K_{A,B} \in (A \Leftarrow B) \Leftarrow A$ such that $(K_{A,B}{}^{\ulcorner}a)^{\ulcorner}b = a$;

(3) $S_{A,B,C} \in ((A \Leftarrow C) \Leftarrow (B \Leftarrow C)) \Leftarrow ((A \Leftarrow B) \Leftarrow C)$

such that $((S_{A,B,C}{}^{\ulcorner}f)^{\ulcorner}g)^{\ulcorner}c = (f^{\ulcorner}c)^{\ulcorner}(g^{\ulcorner}c)$.

It is assumed here that $a \in A$, $b \in B$, $c \in C$, $f \in (A \Leftarrow B) \Leftarrow C$ and $g \in B \Leftarrow C$.

A many-sorted algebra of the kind constructed above may be called a <u>typed Schönfinkel algebra</u>, or a <u>typed Curry algebra</u> if the appropriate equations are postulated.

Curry and Feys have pointed out that, if $\Leftarrow$ is read as "if", the types of I_A, $K_{A,B}$ and $S_{A,B,C}$ are precisely the axioms of the intuitionist implicational calculus, while (0) bears an obvious relation to the usual rule of modus ponens.

Two remarks are in order. The same axioms also appear in the classical propositional calculus accompanied by an additional axiom involving negation. Nonetheless, the negationless formula $A \Leftarrow (A \Leftarrow (B \Leftarrow A))$ is a theorem classically but not in the system without negation. This is why we call the present system intuitionistic.

Secondly, it should be pointed out that $A \Leftarrow A$ is usually not taken as an axiom, but is deduced from the other two axioms. Nevertheless, we prefer to regard it as an axiom here.

Implicit in the the definition of a typed Schönfinkel algebra is a way of regarding each entity of type A as a proof of the formula A.

For example, $S_{A,B,A}$ is by definition a proof of the axiom $((A \Leftarrow A) \Leftarrow (B \Leftarrow A)) \Leftarrow ((A \Leftarrow B) \Leftarrow A)$ and $K_{A,B}$ is by definition a proof of the axiom $(A \Leftarrow B) \Leftarrow A$. Hence $S_{A,B,A}{}^{\ulcorner}K_{A,B}$ is a proof of the theorem $(A \Leftarrow A) \Leftarrow (B \Leftarrow A)$ by modus ponens. Again, $K_{A,C}$ is a proof of the axiom $(A \Leftarrow C) \Leftarrow A$. Take $B = A \Leftarrow C$, then $(S_{A,B,A}{}^{\ulcorner}K_{A,B}){}^{\ulcorner}K_{A,C}$ is a proof of the theorem $A \Leftarrow A$. Since we regard $A \Leftarrow A$ as an axiom, another proof is I_A. Incidentally, we see here that the derivation of $A \Leftarrow A$ from the other axioms of the implicational calculus is nothing else than Schönfinkel's definition of I as $(S{}^{\ulcorner}K){}^{\ulcorner}K$.

The association of entities with proofs becomes even more striking when we compare the free typed Schönfinkel algebra (generated by a set of letters) with pure intuitionistic implicational logic. Then

$$\text{combinators} = \text{proofs}.$$

The reader should note that throughout we distinguish between an axiom such as $(A \Leftarrow B) \Leftarrow A$ and its proof $K_{A,B}$, a pedantic but necessary distinction.

Let us now look at Proposition 1 for typed Schönfinkel algebras. The reader will easily convince himself that the proposition and its proof remain valid, provided x is an indeterminate of type A, where A is any type of $\mathcal{A}$. It asserts that, if $\varphi(x)$ is a polynomial of type B in the indeterminate x of type A, then there is a constant f of type $B \Leftarrow A$ such that $\varphi(x) \underset{x}{=} f{}^{\prime}x$.

In the proof-theoretic interpretation we should regard x as an assumption that A holds. Proposition 1 then becomes the usual deduction theorem with an extra punch at the end:

if $\varphi(x)$ is a proof of B from the assumption x that A holds, then there is a proof of $B \Leftarrow A$ without any assumption such that $\varphi(x) \underset{x}{=} f{}^{\prime}x$.

The extra bit at the end asserts the "equality" of two proofs. Perhaps it would have been better to speak only of "equivalence" of proofs.

As far as I know, this association between combinators and proofs is due to Curry and Feys (1958). It was developed further by Howard (see Stenlund, 1972) and the author (1969, 1972).[3)]

Proposition 2 will also remain valid for typed Curry algebras. It asserts that the proof f of $B \Leftarrow A$, whose existence has been established in Proposition 1, is unique up to equivalence of proofs (which we called equality here).

Proposition 3 will remain valid for typed Curry algebras provided we use the typed λ-calculus and write $f = \lambda_{x\in A}\varphi(x)$.

Proposition 4 will not remain valid for typed Curry algebras without very strong restrictions. After all, the whole purpose of introducing types was to render expressions such as $b^{\ulcorner}b$ meaningless in general. If $b \in B$, it does have a meaning only if $B = B \Leftarrow B$. Proposition 4 remains valid for typed algebras in the following sense:

if A is a type such that $A \Leftarrow A = A$, then every entity f of type A has a fixpoint a such that $f^{\ulcorner}a = a$.

The fragment of propositional logic investigated up to now studiously avoids conjunction, which classically was usually defined in terms of implication and negation. Let us now turn to a fragment of logic, the <u>positive intuitionist propositional calculus</u>, which deals with implication $\Leftarrow$, conjunction $\wedge$ and truth T. We shall present this as a deductive system, making use of the symbol $\rightarrow$ for entailment, in the spirit of Gentzen. Here are our axioms and rules of inference, suitably labelled:

$$(1) \qquad A \xrightarrow{1_A} A, \qquad \frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{gf} C} \;;$$

$$(2) \qquad A \xrightarrow{0_A} T \;;$$

$$(3) \qquad A \wedge B \xrightarrow{\pi_{A,B}} A, \quad A \wedge B \xrightarrow{\pi'_{A,B}} B, \quad \frac{C \xrightarrow{f} A \quad C \xrightarrow{g} B}{C \xrightarrow{\langle f,g\rangle} A \wedge B} \;;$$

$$(4) \qquad (A \Leftarrow B) \wedge B \xrightarrow{\varepsilon_{A,B}} A, \quad \frac{C \wedge B \xrightarrow{h} A}{C \xrightarrow{h^*} A \Leftarrow B}$$

The labels are useful in naming proofs. For example, the commutative law is proved thus:

$$\frac{A \wedge B \xrightarrow{\pi'_{A,B}} B \quad A \wedge B \xrightarrow{\pi_{A,B}} A}{A \wedge B \xrightarrow{\langle \pi'_{A,B}, \pi_{A,B} \rangle} B \wedge A}$$

The label $\langle \pi'_{A,B}, \pi_{A,B} \rangle$ appearing in the last line may be used to denote the whole proof.

We are also interested in an equivalence relation between proofs, which we may as well denote by the equality symbol $=$. We do not bother to write down the usual reflexive, symmetric, transitive and substitution laws for equality. However, we list the following equations, where in $f = g$ it is assumed that f and g have the same source and target.

(1') $f1_A = f$, $1_B f = f$, $(hg)f = h(gf)$, for all $f\colon A \to B$, $g\colon B \to C$ and $h\colon C \to D$;

(2') $f = 0_A$, for all $f\colon A \to 1$;

(3') $\pi_{A,B}\langle f,g\rangle = f$, $\pi'_{A,B}\langle f,g\rangle = g$, $\langle \pi_{A,B}h, \pi'_{A,B}h\rangle = h$, for all $f\colon C \to A$, $g\colon C \to B$ and $h\colon C \to A \wedge B$;

(4') $\varepsilon_{A,B}\langle h^* \pi_{C,B}, \pi'_{C,B}\rangle = h$, $(\varepsilon_{A,B}\langle k\pi_{C,B}, \pi'_{C,B}\rangle)^* = k$, for all $h\colon C \wedge B \to A$ and $k\colon C \to A \Leftarrow B$.

The reader will recognize that (1) and (1') define a category. Moreover, (2) and (2') assert that T is a terminal object, usually denoted by 1. Furthermore, (3) and (3') assert that $A \wedge B$ is the Cartesian product of A and B, usually written $A \times B$. Finally, (4) and (4') express the fact that $(-) \Leftarrow B$ is right adjoint to $(-) \wedge B\colon \mathcal{A} \to \mathcal{A}$, which makes $\Leftarrow$ an internal hom-functor, the usual notation for $A \Leftarrow B$ being A^B. (1) to

(3) and (1') to (3') define a <u>Cartesian category</u>, (1) to (4) and (1') to (4') define a <u>Cartesian closed category</u>.

Cartesian closed categories were introduced under this name by Eilenberg and Kelly (1966). Lawvere (1969) also pointed out that $\times$ is right adjoint to the diagonal $A \to A \times A$ and emphasized the analogy with propositional logic. Our definition of Cartesian closed categories is slightly different, in as much as 1, $\times$ and exponentiation are not only said to exist, but are regarded as part of the structure. In fact, our notion of Cartesian closed category is algebraic over the category of categories, or the category of graphs for that matter, in much the same way that the notion of group is algebraic over the category of sets.

For future reference, we also note the following consequence of (1') to (4'):

(5') $$\langle f,g\rangle h = \langle fh,gh\rangle.$$

This is proved thus: let $k = \langle f,g\rangle h$, then, omitting subscripts, we have

$$\pi k = fh, \quad \pi' k = gh$$

by (3'), hence

$$\langle fh,gh\rangle = \langle \pi k,\pi' k\rangle = k,$$

by (3') again.

In the author's opinion, it is a tour de force to present propositional logic without conjunction. Curiously, the same tour de force is found in the paper by Eilenberg and Kelly, who went to some trouble to eliminate the Cartesian product from Cartesian closed categories. One could argue that closed categories without products are essentially the same as typed Curry algebras. It therefore seems reasonable to proceed from the study of typed Curry algebras to the study of Cartesian closed categories.

I had proved in 1972 that functional completeness holds for Cartesian closed categories, provided that these satisfied a finite set of equations, like those due to Curry, and conjectured that these equations are already a consequence of (1') to (4') above, that is, they hold in any Cartesian closed category. I proved this later (1974) and shall give another exposition of the proof here. The new versions of Propositions 1 and 2 will be called Theorems 1 and 2.

First, we must explain what it means to adjoin an indeterminate morphism $x: 1 \to A$ to a Cartesian closed category $\mathcal{A}$, A being an object of $\mathcal{A}$, which we also regard as a formula. In the same spirit, we may regard x as an assumption that T entails A. The objects of $\mathcal{A}[x]$ are the same as those of $\mathcal{A}$, but the morphisms $\varphi(x): B \to C$ in $\mathcal{A}[x]$ may be regarded as proofs that B entails C on the assumption x. Equality between proofs or polynomials is determined by (1') to (4') and is denoted by $\underset{x}{=}$. One must check of course that $\mathcal{A}[x]$ thus constructed has the expected universal property.

THEOREM 1. (Deduction theorem). With every proof $\varphi(x): B \to C$ from the assumption $x: 1 \to A$ there is associated a proof $f: A \wedge B \to C$ not depending on this assumption. Moreover $f\langle x0_B, 1_B\rangle \underset{x}{=} \varphi(x)$.

It should be pointed out that we have presented the positive intuitionist propositional calculus as a deductive system, so that the usual deduction theorem becomes absorbed in the rules governing the deduction symbol $\to$, thus:

$$\frac{A \wedge B \to C}{A \to C \Leftarrow B} .$$

However, at a higher level the horizontal bar functions as a deduction symbol and Theorem 1 plays the role of a new deduction theorem.

Proof. Clearly, every polynomial $\varphi(x)$ must have one of the five forms: k, x, $\langle\psi(x), \chi(x)\rangle$, $\chi(x)\psi(x)$, $\psi(x)^*$, where k is a constant and where $\psi(x)$ and $\chi(x)$ are shorter polynomials. By inductional assumption we have, omitting subscripts:

$$\psi(x) \underset{x}{=} g\langle x0,1\rangle,\ \chi(x) \underset{x}{=} h\langle x0,1\rangle.$$

The result now follows by verifying the following equations:

$$k\pi'_{A,B}\langle x0_B, 1_B\rangle \underset{x}{=} k ,$$

$$\pi_{A,1}\langle x0_1, 1_1\rangle \underset{x}{=} x ,$$

$$\langle g,h\rangle\langle x0_B, 1_B\rangle \underset{x}{=} \langle\psi(x),\chi(x)\rangle ,$$

$$h<\pi_{A,B},g><x0_B,1_B> \underset{x}{=} \chi(x)\psi(x) ,$$

$$(g\alpha_{A,B,D})^*<x0_B,1_B> \underset{x}{=} \psi(x)^* ,$$

where $\alpha_{A,B,D}$: $(A\wedge B) \wedge D \to A \wedge (B\wedge D)$ is given by

$$\alpha_{A,B,D} = <\pi_{A,B}\pi_{A\wedge B,D},<\pi'_{A,B}\pi_{A\wedge B,D},\pi'_{A\wedge B,D}>> .$$

The last equation is proved by showing (omitting subscripts) that

$$\varepsilon<(g\alpha)^*<x0,1>\pi,\pi'> \underset{x}{=} g<x0,1> ,$$

which follows from a routine calculation, as in (Lambek 1974, p. 272).

THEOREM 2. (Functional completeness of Cartesian closed categories). For every polynomial $\varphi(x)$: $B \to C$ in an indeterminate x: $1 \to A$ there is a unique constant f: $A\times B \to C$ such that $f<x0_B,1_B> \underset{x}{=} \varphi(x)$.

Proof. Let us write $\kappa_{x\in A}\varphi(x)$ for the constant f which is assigned to the proof $\varphi(x)$ in the proof of Theorem 1. Thus, when k is a constant, we have:

(i) $\kappa_{x\in A}k = k\,\pi'_{A,B}$;

(ii) $\kappa_{x\in A}x = \pi_{A,1}$;

(iii) $\kappa_{x\in A}<\psi(x),\chi(x)> = <\kappa_{x\in A}\psi(x),\kappa_{x\in A}\chi(x)>$;

(iv) $\kappa_{x\in A}(\chi(x)\psi(x)) = \kappa_{x\in A}\chi(x)<\pi_{A,B},\kappa_{x\in A}\psi(x)>$;

(v) $\kappa_{x\in A}(\psi(x)^*) = (\kappa_{x\in A}\psi(x)\alpha_{A,B,D})^*$, where (iii), (iv) and (v) are subject to the restriction that $\varphi(x)$ is not constant.

Note, in particular, the following special case of (iv), in view of (i):

(vi) $\quad \kappa_{x\in A}(k\ \psi(x)) = k\ \kappa_{x\in A}\psi(x)$,

where again it is assumed that $\psi(x)$ is not constant.

We first show that the above restrictions on (iii) to (v), and therefore on (vi), may be removed. For example, to take the most difficult case, let us check that (v) holds when $\psi(x)$ is a constant g. Note that $\pi'\alpha = \langle\pi'\pi,\pi'\rangle$ by (3'). Then

$$\begin{aligned}
(\kappa_{x\in A}\ g\ \alpha)^* &= (g\ \pi'\alpha)^* && \text{by (i)}\\
&= (g\ \beta)^* && \text{if } \beta = \langle\pi'\pi,\pi'\rangle\\
&= (\varepsilon\ \langle g^*\pi,\pi'\rangle\beta)^* && \text{by (4')}\\
&= (\varepsilon\ \langle g^*\pi\beta,\pi'\beta\rangle)^* && \text{by (5')}\\
&= (\varepsilon\ \langle g^*\pi'\pi,\pi'\rangle)^* && \text{by (3')}\\
&= g^*\pi' && \text{by (4')}\\
&= \kappa_{x\in A}(g^*) && \text{by (i)} .
\end{aligned}$$

We next show that $\kappa_{x\in A}\varphi(x)$ depends only on the polynomial $\varphi(x)$, which may be regarded as the proof $\varphi(x)$ modulo the equivalence relation $\underset{x}{=}$. Let us write $\varphi(x) \equiv \psi(x)$ for $\kappa_{x\in A}\varphi(x) = \kappa_{x\in A}\psi(x)$. Then it is easily checked that $\equiv$ has the substitution property and satisfies all the conditions which equality in $A[x]$ must satisfy (see the sample calculation below). Since $\underset{x}{=}$ is by definition the smallest such equivalence relation, it follows that $\underset{x}{=}$ is contained in $\equiv$, that is,

$$(*) \qquad \varphi(x) \underset{x}{=} \psi(x) \text{ implies } \kappa_{x\in A}\varphi(x) = \kappa_{x\in A}\psi(x),$$

as claimed.

As promised, we shall prove, for example, that $\varepsilon < \chi(x)^{*}\pi,\pi'> \equiv \chi(x)$, to take the worst case. Indeed, writing $\kappa_{x\in A}\chi(x) = h$, we have

$$\begin{aligned}
\kappa_{x\in A}(\varepsilon<\chi(x)^{*}\pi,\pi'>) &= \varepsilon\ \kappa_{x\in A}<\chi(x)^{*}\pi,\pi'> && \text{by (vi)}\\
&= \varepsilon<\kappa_{x\in A}(\chi(x)^{*}\pi),\kappa_{x\in A}\pi'> && \text{by (iii)}\\
&= \varepsilon<\kappa_{x\in A}\chi(x)^{*}<\pi,\kappa_{x\in A}\pi>,\pi'\pi'> && \text{by (iv) and (i)}\\
&= \varepsilon<(h\alpha)^{*}<\pi,\pi\pi'>,\pi'\pi'> && \text{by (v) and (i).}
\end{aligned}$$

Now the associativity morphism $\alpha = <\pi\pi,<\pi'\pi,\pi'>>$ clearly has an inverse α^{-1}, so the above

$$\begin{aligned}
&= \varepsilon<(h\alpha)^{*}<\pi\alpha,\pi\pi'\alpha>,\pi'\pi'\alpha>\alpha^{-1} && \text{by (5')}\\
&= \varepsilon<(h\alpha)^{*}<\pi\pi,\pi'\pi>,\pi'>\alpha^{-1} && \text{by (3')}\\
&= \varepsilon<(h\alpha)^{*}\pi,\pi'>\alpha^{-1} && \text{by (3')}\\
&= h\alpha\alpha^{-1} = h && \text{by (4').}
\end{aligned}$$

We are finally ready to prove the uniqueness of f.

Suppose $g<x0_B,1_B> \underset{x}{=} \varphi(x)$, then

$$\begin{aligned}
\kappa_{x\in A}\varphi(x) &= \kappa_{x\in A}(g\langle x0_B, 1_B\rangle) && \text{by } (*) \\
&= g\,\kappa_{x\in A}\langle x0_B, 1_B\rangle && \text{by (vi)} \\
&= g\,\langle \kappa_{x\in A}(x0_B), \kappa_{x\in A}1_B\rangle && \text{by (iii)} \\
&= g\,\langle \kappa_{x\in A}x\langle \pi_{A,B}, \kappa_{x\in A}0_B\rangle, 1_B\pi'_{A,B}\rangle && \text{by (iv) and (i)} \\
&= g\,\langle \pi_{A,1}\langle \pi_{A,B}, 0_B\pi'_{A,B}\rangle, \pi'_{A,B}\rangle && \text{by (ii) and (i)} \\
&= g\,\langle \pi_{A,B}, \pi'_{A,B}\rangle && \text{by (3')} \\
&= g\,1_{A\times B} && \text{by (3')} \\
&= g && \text{by (1')}\ .
\end{aligned}$$

REMARK 1. Theorems 1 and 2 remain valid for Cartesian categories, that is, if exponentiation is absent.

REMARK 2 (for categorists). Theorem 2 may be interpreted as saying that $\mathcal{A}[x]$ is the Kleisli category of the obvious cotriple on $\mathcal{A}$ associated with the function $A \times (-)\colon \mathcal{A} \to \mathcal{A}$, or of the triple on $\mathcal{A}$ associated with the functor $(-)^A$, see (Lambek 1974) for details. It is also not difficult to prove directly that the Kleisli category has the universal property of $\mathcal{A}[x]$. This alternative approach (loc. cit. § 5) bypasses the historical development discussed above, but also loses the algorithm for calculating $\kappa_{x\in A}\varphi(x)$. Such a direct approach, without mention of Kleisli categories by that name, was also used in (Volger, 1975).

As a special case of Theorem 2 we obtain the following.

COROLLARY. For every polynomial $\varphi(x)\colon 1 \to C$ in an indeterminate $x\colon 1 \to A$ there is a unique constant $g\colon A \to C$ such that $gx \underset{x}{=} \varphi(x)$ and, equivalently, a unique constant $h\colon 1 \to C^A$ such that $\varepsilon_{C,A}\langle h,x\rangle \underset{x}{=} \varphi(x)$.

Proof. To obtain this from Theorem 2, merely put

$$g = \kappa_{x\in A}\varphi(x)\ \langle 1_A, 0_A\rangle\ ,$$

$$h = (g\pi'_{1,A})^* = (\kappa_{x\in A}\varphi(x)\ \langle \pi'_{1,A}, \pi_{1,A}\rangle)^*.$$

Actually, the corollary is no weaker than the theorem, since polynomials $B \to C$ are in one-to-one correspondence with polynomials $1 \to C^B$. To compare the corollary with the λ-calculus, we write

$$h^{\ulcorner} = \varepsilon_{C,A}\ \langle h0_A, 1_A\rangle\ ,$$

so that

$$h^{\ulcorner}x \underset{x}{=} \varepsilon_{C,A}\ \langle h,x\rangle \underset{x}{=} \varphi(x)\ ,$$

and define

$$\lambda_{x\in A}\varphi(x) = (\kappa_{x\in A}\varphi(x)\langle\pi'_{1,A}, \pi_{1,A}\rangle)^* = h.$$

There are two traditional applications of the λ-calculus. The first is to arithmetic, in particular, the theory of recursive functions. The second is to the foundations of mathematics via Curry'a illative combinatory logic or the type theory of Church. We shall briefly discuss the first application and regret that space does not permit discussion of the second and the use which has recently been made of it in the construction of free toposes.

Writing

$$f \circ g = \lambda_x (f^{\ulcorner}(g^{\ulcorner}x)) \ ,$$

one may introduce natural numbers

$$0 = \lambda_x \, I, \ 1 = \lambda_x \, x = I, \ 2 = \lambda_x \, (x \circ x), \ \ldots..$$

and successor, addition, multiplication and exponentiation by

$$S^{\ulcorner}n = \lambda_y (y \circ (n^{\ulcorner}y) \ ,$$

$$m + n = \lambda_y \, ((m^{\ulcorner}y) \circ (n^{\ulcorner}y)) \ ,$$

$$m \, n = m \circ n \ ,$$

$$m^n = n^{\ulcorner}m \ .$$

These are recursive functions, and Kleene proved in 1936 that recursive (or Turing computable) functions are precisely those recursive functions which are definable in the type-free λ-calculus.

Unfortunately there are difficulties when one introduces types. If a has type A, then f and g in $(f \circ g)^{\ulcorner}a$ have types $A^A = B$, say. For $n^{\ulcorner}f$ to make sense, n will then have to be of type B^B, and for $n^{\ulcorner}m$ to make sense, m will have to be of type B. If m and n are both natural numbers, we are led to $B^B = B$, and this would be a consequence of $A^A = A$.

Now, it is certainly possible to postulate a type A such that $A^A = A$ (Scott, 1972). However, it seems more natural to postulate a type N with entities $0 \in N$, $S \in N \Leftarrow N$ and

$R_A \in ((A \Leftarrow A) \Leftarrow (A \Leftarrow A)) \Leftarrow N$ satisfying certain equations. In the language of Cartesian closed categories, we thus require an object N and morphisms $0\colon 1 \to N$, $\sigma\colon N \to N$ and $\rho_A\colon N \to (A^A)^{(A^A)}$ satisfying certain identities, to wit:

$$(\rho_A\ 0)\ulcorner f = I_A\ ,$$

$$(\rho_A(\sigma n))\ulcorner f = f \circ ((\rho_A\ n)\ulcorner f),$$

for all $n\colon 1 \to N$ and $f\colon 1 \to A^A$.

These identities are related to the Peano-Lawvere axiom, which deals with a morphism $A \times A^A \to A^N$ instead. They imply the existence (but not the uniqueness) of a morphism $N \to A$ such that the following diagram commutes for given morphisms $1 \to A$ and $A \to A$:

$$\begin{tikzcd}
1 \arrow[r, "0"] \arrow[dr] & N \arrow[r, "\sigma"] \arrow[d, dotted] & N \arrow[d, dotted] \\
& N \arrow[r] & N
\end{tikzcd}$$

The Peano-Lawvere axiom also requires uniqueness.

Marie-France Thibault called a Cartesian closed category with N, 0, σ, ρ satisfying the above identities a <u>prerecursive category</u>. She proved the following:

(a) The set of primitive recursive functions is properly contained in the set F of functions represented by morphisms $N^k \to N$ in the free prerecursive category generated by the empty category ;

(b) $\mathcal{F}$ is properly contained in the set of all recursive functions ;

(c) $\mathcal{F}$ coincides with the set of type-recursive functions discussed by Grzegorczyk.

It seems clear from (Hindley, Lercher and Seldin, 1972, Chapter 11) that $\mathcal{F}$ is also essentially the same as the set of Gödel's functions of finite type.

FOOTNOTES

1) Here $=$ denotes equality in $\mathcal{A}$, $\mathcal{A}[x]$, $\mathcal{A}[x,y]$, etc. simultaneously, subscripts having been suppressed. It is an equivalence relation subject to the following rules:

$$\frac{f = g \qquad a = b}{f^{\ulcorner}a = g^{\ulcorner}b}\ , \quad \frac{\varphi(x) = \psi(x)}{\lambda_x\varphi(x) = \lambda_x\psi(x)}\ .$$

2) Actually, the application symbol $^{\ulcorner}$ should carry subscripts A and B; but we omit these whenever they are clear from the context, as they are throughout this paper.

3) For related ideas and further references see also the thought provoking paper by Scott (1970).

ACKNOWLEDGEMENTS.

The author wishes to acknowledge support from the Natural Sciences and Engineering Research Council of Canada and partial support from the Quebec Department of Education. He is indebted to W.S. Hatcher for drawing his attention to an error in the original manuscript.

REFERENCES

Church, A. (1940). A formulation of the simple theory of types, J. Symbolic Logic 5, 56-58.

Church, A. (1941). "The calculi of lambda conversion", Princeton University Press, Princeton.

Curry, H.B. (1930). Grundlagen der Kombinatorischen Logik, Amer. J. Math. 52, 789-834.

Curry, H.B. (1932). Some additions to the theory of combinators, Amer. J. Math. 54, 551-558.

Curry, H.B. (1942). The inconsistency of certain formal logics, J. Symbolic Logic 7, 115-117.

Curry, H.B. (1968). Recent advances in combinatory logic, Bull. Soc. Math. Belg. 20, 288-298.

Curry, H.B. and Feys, R. (1958). "Combinatory logic", Vol. 1, North-Holland, Amsterdam.

Curry, H.B., Hindley, J.R. and Seldin, J.P. (1972). "Combinatory logic", Vol. 2, North-Holland, Amsterdam.

Eilenberg, S. and Kelly, G.M. (1966). Closed categories. In "Proc. Conf. on Categorical Algebra, LaJolla 1965" (Eds. S. Eilenberg et. al.), 421-562.

Grzegorczyk, A. (1964). Recursive objects in all finite types, Fundamenta Mathematicae 54, 73-93.

Hindley, J.R., Lercher, B. and Seldin, J.P. (1972). "Combinatory logic", Cambridge University Press, Cambridge.

Kleene, S.C. (1936). λ-definability and recursiveness, Duke Math. J. 2, 340-353.

Kleene, S.C. and Rosser, J.B. (1935). The inconsistency of certain formal logics, Ann. Math. 36, 630-636.

Kleisli, H. (1965). Every standard construction is induced by a pair of adjoint functors, Proc. Amer. Math. Soc. 16, 544-546.

Lambek, J. (1969). Deductive systems and categories II, Lecture Notes in Mathematics 86, 76-122.

Lambek, J. (1972). Deductive systems and categories III, Lecture Notes in Mathematics 274, 57-82.

Lambek, J. (1974). Functional completeness of Cartesian categories, Annals of Math. Logic 6, 259-292.

Lambek, J. (1980). From types to sets, Advances in Mathematics 35, (to appear).

Lawvere, F.W. (1969). Adjointness in foundations, Dialectica 23, 281-296.

MacLane, S. (1971). "Categories for the working mathematician", Springer, New York.

Rosenbloom, P.C. (1950). "The elements of mathematical logic", Dover, New York.

Rosser, J.B. (1942). New sets of postulates for combinatory logic, J. Symbolic Logic 7, 18-27.

Schönfinkel, M. (1924). Über die Bausteine der mathematischen Logik, Math. Ann. 92, 305-316. Translated in "From Frege to Gödel" (Ed. J. Van Heijenoort).

Scott, D. (1970). Constructive validity, Lecture Notes in Mathematics 125, 237-275.

Scott, D. (1972). Continuous lattices, Lecture Notes in Mathematics 274, 97-136.

Stenlund, S. (1972). "Combinators, λ-terms and proof theory", Reidel, Dordrecht-Holland.

Szabo, M.E. (1974). A categorical equivalence of proofs, Notre Dame J. Formal Logic 15, 177-191.

Szabo, M.E. (1978). "Algebra of proofs", North-Holland, Amsterdam.

Thibault, M.-F. (1977). "Représentations des fonctions récursives dans les catégories", Thesis, McGill University, Montreal.

Thibault, M.-F. (1978). Prerecursive categories, Manuscript.

van Heijenoort, J. (1967). "From Frege to Gödel", Harvard University Press, Cambridge-Massachussets.

Volger, H. (1975). Completeness theorem for logical categories, Lecture Notes in Mathematics 445, 51-86.

RELATING THEORIES OF THE λ-CALCULUS

Dana S. Scott

Merton College
Oxford

Dedicated to Professor H. B. Curry on the occasion of his 80th Birthday

Mathematical theories arise for many different reasons, sometimes in connection with specific applications and often owing to accidental inspiration. From time to time we ought to ask ourselves concerning our theories where *should* they have come from; usually the answer will have little to do with the exact historical development. The λ-calculus is, I feel, a case in point. In Scott (1980), in the Kleene Festschrift, I made up a story of where the theory of type-free λ-calculus *could* have come from. Any number of people who heard my lecture and read the manuscript were cross with me. They said "But it didn't develop that way! And besides we doubt it ever would have." But this reaction misses the point of my story. I shall not, however, repeat the earlier story here, for the point of the present paper is different. For those people who do not like to discuss philosophy - even Philosophy of Mathematics - my remarks here can be taken as a suggestion of how to group diverse models of λ-calculus rather uniformly under a general scheme. The scheme is by now rather well known and not at all original with me. What I hope can be regarded as a useful contribution is my putting of the ideas in a certain order. As I consider the order to be a natural one, I feel there is philosophical significance to my activity; but I should not want to force this view on anyone.

1. THEORIES OF FUNCTIONS.

Everyone agrees that λ-calculus is a theory of functions. But we must ask: "What kind of a theory?" And also: "Have we got the best theory?" Personally, I think we should also inquire: "How does it relate to other theories?" I certainly find many discussions far too silent on this last issue.

Well, what other theories are there? Certainly *set theory* comes to mind at once, and no set theory would be worth its salt if it did not provide a theory of functions. Let us not try to catalogue the various known theories here but look at a theory in the style of Zermelo - and we do not have even to be too specific, since in any case such a theory is very standard. What is "unsatisfactory" about Zermelo's theory is the limitization-of-size view of sets: any *one* set A is extremely small compared to the size of V, the class or universe of *all* sets. Thus, functions $f : A \to B$ mapping one set A into another set B tell us very little about operations on all sets, maps on V into V. We therefore have an urge to "improve" our set theory by constructing a *class theory*. Sets are elements $A \in V$; while classes are subcollections $B \subseteq V$. As V is (by the usual assumptions) so highly closed under so many operations, we have no difficulty in construing certain classes as maps $F : V \to V$. For example for all $X \in V$ we could have $F(X) = \{X\}$ or $F(X) = A \times X$ (where A is a fixed set).

The passage from sets to classes is a familiar and useful move in the formalization of the theory: many things can be done generally for classes and then specialized to sets. And having a notation for functions defined on all sets is in many cases a great advantage. But wait. What about operations on classes? What should we say about them? Given any two classes A and B, we can form their union, $A \cup B$. The operation,

$\cup : V \times V \to V$, of union of *sets* does not directly apply to *classes* even though there is a connection. Do we also want a theory of class operations? Do we have to go to hyperclasses (classes of classes)? Is there any end to this expansion?

[*An Aside*: The story of Scott (1980) was meant to suggest one answer - the one known to Plotkin (1972). Namely, we consider only "continuous" class operations. These are objects F such that F(X) is defined for every class $X \subseteq V$ and F(X) is a class, too. Moreover F should satisfy:

(1) $X \subseteq Y$ always implies $F(X) \subseteq F(Y)$;
(2) Whenever $A \subseteq F(X)$ and A is a set, then $A \subseteq F(B)$ for some set $B \subseteq X$.

We do not have time to discuss the justification of the word "continuous" here; suffice it to say that conditions (1) and (2) are not as strict as they at first might seem. Every *ordinary* map $f : V \to V$ determines a continuous class operator by the definition:

$$F(X) = \{f(x) \mid x \in X\}.$$

Furthermore, F determines f, for we have:

$$y = f(x) \text{ iff } \{y\} = F(\{x\}),$$

for all $x,y \in V$. In a suitable sense, then, nothing has been lost; but what has been gained?

The reply is that continuous class operators can be *identified* with classes. We could write, for instance:

$$F = \{(A,B) \mid A,B \in V \wedge A \subseteq F(B)\},$$

where, say:

$$(A,B) = \{\{A\}, \{A,B\}\}.$$

More in harmony with Scott (1980) would be:

$$F = \{(a,B) \mid a,B \in V \wedge a \in F(B)\}.$$

Either trick reduces operator theory to class theory - in the continuous case. And the same trick could be carried over to other kinds of set theory (e.g. Quine's). What we know is that operator theory gives a model for λ-calculus; it is a quite elementary model, too.

Nice as this connection is, it is not the topic of the present paper: we do not want to make λ-calculus depend on set theory, since then we have still to explain where set theory comes from. But the connection should be borne in mind.]

Perhaps set theory brings in too many extraneous issues. V, after all, is a massive object closed under all manner of strange operations. What we are probably seeking is a "purer" view of functions: a theory of functions in themselves, not a theory of functions derived from sets. What, then, is a pure theory of functions? Answer: category theory.

General category theory is a *very* pure theory: it is the milk-and-water theory of functions under *composition*. This composition operation is associative and possesses neutral elements (compositions of zero terms). That is about all you can say about it except to stress that it is also a rather bland theory of *types*. Every function f has a (unique) *domain* and *codomain*, and we write:

$$f : \mathrm{dom}\ f \to \mathrm{cod}\ f$$

Every possible domain is a codomain (and conversely), because if A is such, then

$$\mathrm{dom}\ 1_A = A = \mathrm{cod}\ 1_A,$$

where 1_A is the neutral element of type A.

[If we want to be especially parsimonious in entities, we can even write $1_A = A$, because each of 1_A and A uniquely determines the other.]

The point of distinguishing domains and codomains is not only do they specify the type of f, but a composition $g \circ f$ is defined if, and only if, dom g = cod f. And then $\mathrm{dom}(g \circ f) = \mathrm{dom}\, f$ and $\mathrm{cod}\,(g \circ f) = \mathrm{cod}\,(g)$. We usually write this as a "rule of inference":

$$\frac{f : A \to B \qquad g : B \to C}{g \circ f : A \to C}$$

with the understanding that the typing of $f \circ g$ can only be obtained by such an application of the rule. The types, then, are invoked just to type functions, and the only theory involved is that of the "transition" of types under composition.

Sets (and set-theoretical mappings) do of course form a category; category theory is *meant* to be more general than set theory. We should construe the function entities here as triples of sets (A,f,B) where

$$f \subseteq A \times B \wedge \forall x \in A\ \exists! y \in B.\ (x,y) \in f.$$

The definition of composition is obvious. Sets, in this way, give us only one special example of a category.

I beg forgiveness of the reader for boring him. All of this is well known to the moderately awake undergraduate in mathematics. Indeed, that is the point: there is plenty of evidence now that category theory is a natural *and* useful theory of functions. I do not have to rehearse the examples as they can be found in any number of books (e.g. Mac Lane [1971]). There is a rather important logical point to stress, however - important for anyone who has thought about λ-calculus models. Category theory is very *extensional*. We assume as axioms the *equations*:

$$1_A \circ f = f \circ 1_B = f \quad \text{and}$$

$$h \circ (g \circ f) = (h \circ g) \circ f,$$

provided $A = \mathrm{dom}\, f$ and $B = \mathrm{cod}\, f$ and the double compositions are defined. These are *functional* equations, and they say that two functions defined in *different* ways are in fact *identical*.

Furthermore, identical things can everywhere replace one another.

This point about extensionality may not seem exciting or important, but the logician should remember that, in certain *intensional* theories of functions, "obvious" definitions will *not* provide categories. We shall return to this point later.

But is category theory the long-sought answer? No, no, not at all. Category theory *pure* provides nothing explicitly aside from identity functions - and they occur only *if* we have some possible domain. We do get compositions *if* we have the necessary terms. Thus, as it stands, category theory has no existential import. (It was not meant to.) Set theory has "too much" existential import. (It was meant to.) What we seek is the middle way - and an argument that the middle way is natural and general.

There is no need to build up unnecessary suspense: the middle way is the theory of the (so called) *cartesian closed categories*. Fortunately Lambek has written extensively about the theory, and I can refer the reader to his papers for further details; I also am happy to acknowledge his writings as helping me understand what is going on. If we remark that his paper in *this* volume is called "From λ-calculus to cartesian closed categories", we might say that my present paper ought to be called "From cartesian closed categories to λ-calculus." I am trying to find out where λ-calculus *should* come from, and the fact that the notion of a cartesian closed category (c.c.c) is a late developing one (Eilenberg & Kelly (1966)), is not relevant to the argument: I shall try to explain in my own words in the next section why we should look to it *first*.

2. A THEORY OF TYPES

I say "a theory", because there are many possible theories; indeed pure category theory is one of the theories. Its weakness lies in the fact that we are given no construction principles, no way of making new types from old. From the point of view of logic what should we expect? What more do we want to say beyond relations between types which hold when a mapping statement $f : A \to B$ obtains.

An immediate question that must come to anyone's mind concerns the *arity* of functions. The usual way of reading a mapping statement is to take it as a statement about *one-place* functions, and the $\circ$ of composition is the composition of one-place functions. This seems very restricted.

People have suggested generalizing categories to *multi-place* functions with concomitant compositions (cf. e.g. the book Szabo (1978)), but it does not seem the neatest solution. Much easier is to assume that the category has cartesian products - and more specifically particular representatives of the product domains are chosen. As a special case we will know what the cartesian power A^n is for each $n=0,1,2,\ldots$, and n-ary functions are then maps $f : A^n \to B$. Not much of a surprise.

We have to take care, however. In the first place, a given category may not have cartesian products (it fails to have enough types). Even if it does, the maps allowed may be too restricted - for logical purposes. Take the category of groups and homomorphisms, for example. The required products exist. A map $f : A^2 \to B$ in this category has to be a group homomorphism, naturally. Suppose two maps $u,v : A \to B$ were given. Intuitively we think in terms of elements and that we are mapping $x \mapsto u(x)$ and $y \mapsto v(y)$. The pointwise group product of the maps, namely, $x,y \mapsto u(x) \cdot v(y)$ is a very nice map $g : A^2 \to B$ in the ordinary sense - but unless the group B is abelian, g is not a homomorphism. It is a "logical" map but not an "algebraic" map. Pure category theory applies to many algebraic situations (as everyone knows that is why it is a good theory), but not all categories are "logical" even if they have products. In the example of groups, what was "missing" was the group multiplication $\mu : B^2 \to B$ as a map in the category. (Inverse is missing as well, since it reverses order.) There is an interesting theory of algebraic theories that address the question of the proper categorial construction of categories of algebras, but I do not think we should invoke that theory here.

The precise description of products is as follows. We assume our category has a special domain $\mathbb{1}$ (*the* empty product, so $A^0 = B^0 = \mathbb{1}$), and for each domain A a special map $0_A : A \to \mathbb{1}$. (The domain $\mathbb{1}$, intuitively, has just one "element".) Moreover the rule about maps is that 0_A is unique; that is whenever $f : A \to \mathbb{1}$, then we have the equation

$$f = 0_A.$$

Concerning *binary* products, we have for any two domains A and B a special choice of a domain $A \times B$ (and so $A^{n+1} = A^n \times A$), and special maps

$$p_{AB} : A \times B \to A$$
$$q_{AB} : A \times B \to B$$

But the mere existence of "projections" does not characterize $A \times B$ as a product. We have to assume that there is a chosen pairing operation $\langle f,g \rangle$ on maps such that types are assigned by the rule:

$$\frac{f : C \to A \qquad g : C \to B}{\langle f,g \rangle : C \to A \times B}$$

Moreover, provided f and g are as above and $h : C \to A \times B$ we have to assume

$$p_{AB} \circ \langle f,g \rangle = f$$

$$q_{AB} \circ \langle f,g \rangle = g$$

$$\langle p_{AB} \circ h, q_{AB} \circ h \rangle = h.$$

that is to say, there is an explicit one-one correspondence between the pairs of maps f,g and the maps h into the product. This all now makes $A \times B$ well behaved within the category.

So much for a theory of tuples and multiary maps. But we still want a *theory* of functions: a category allows us to talk of *selected* functions, while we would want various equations to have a force relating to *arbitrary* functions. The answer to this desire is *function spaces* as explicit domains in the category. Given A and B we want to form $(A \to B)$ as a domain in its own right; if so, there are many maps that have to be set down to make the function space behave. (And here we must definitely leave the category of groups.)

In the first place there has to be an *evaluation map* $\varepsilon_{BC} : (B \to C) \times B \to C$ with the intuitive interpretation that it is the map $f,x \mapsto f(x)$. In the second place there has to be a map for shifting around variables; more precisely, suppose $h : A \times B \to C$ is a map with two arguments. In an evaluation $h(x,y)$, we can think of holding x constant and regarding $h(x,y)$ as a function of y. We need a name for this function - and for the correspondence with possible values of x. We write

$$\Lambda_{ABC}\, h : A \to (B \to C)$$

so that the function we were thinking of - given x - was $(\Lambda_{ABC}\, h)(x)$. But all this function-value notation is not categorical notation; what we have to say is that there is a one-one correspondence via Λ between maps $h : A \times B \to C$ and maps $k : A \to (B \to C)$. This comes down to these two equations:

$$\varepsilon \circ \langle (\Lambda h) \circ p,\ q \rangle = h, \quad \text{and}$$

$$\Lambda(\varepsilon \circ \langle k \circ p,\ q \rangle) = k.$$

(It is necessary to have subscripts here: Λ_{ABC}, p_{AB}, q_{AB}, and ε_{BC}; but we leave them off when there is no ambiguity.)

The notation is now wholly categorical (and mostly unreadable). Category theorists put the whole thing (that is, the definition of a c.c.c., which is what we have just given) into the language of functors, which has a lot of sense. But if you have never seen any abstract category theory before, it is really rather too abstract. The idea of a c.c.c. as a system of types is, I think, reasonably simple. *Each c.c.c. represents a theory of functions.* The maps in the category are certain special functions that are used to express the relations between the types (the domains of the category). In order to be able to deal with multiary functions, we assume we can form (and analyze) products. In order to be able to work with transformations of arbitrary functions ("arbitrary" within the theory), we assume we can form function spaces: this is where "higher" types enter the theory, as in the sequence of domains:

$$A,\ (A \to A),\ ((A \to A) \to A),\ (((A \to A) \to A) \to A),\ldots$$

To be able really to view these domains as function spaces, certain operations, ε and Λ, with characteristic equations have to be laid down.

If a c.c.c. is a theory of functions (and we include here higher-type functions), then the theory of c.c.c's is the *theory of types* of the title of this section. It is only one such theory. "Bigger" theories could be obtained by demanding more types: for example we could axiomatize *coproducts* (disjoint sums), $\mathbb{0}$, and $A + B$. We could demand *infinite* products and coproducts. We could throw in a type Ω of "propositions" so that higher types like $(A^n \to \Omega)$ correspond to n-ary *predicates*. This gets into *topos theory* (as in Johnstone (1977) or Goldblatt (1979) - just to name two recent texts). But the bigger the theory, the more involved, and full definitions at this point would not help this discussion very much.

We could also look for "smaller" theories. Some examples - of a rather highly formal nature - can be found in Szabo (1978) with an indication of the algebraic interest of these other type theories. However, a c.c.c. is rather more "logical" and good as a middle ground; further Lambek has explained the logical interest; *there is a perfect correspondence between c.c.c.'s and (extensional) typed λ-calculi.* The reader can turn to Lambek's paper for details and references.

Roughly put, when we formally define the typed λ-language (with types in the given c.c.c.), then if τ is a typed λ-expression with free variables of types $A_0, A_1, \ldots, A_{n-1}$, we can define the "meaning" of τ as a map

$$[[\tau]] : A_0 \times A_1 \times \ldots \times A_{n-1} \to B,$$

where B is the type of τ. For example if u is a variable of type $(B \to C)$ and v a variable of type B, then

$$[[u(v)]] : (B \to C) \times B \to C,$$

and in fact $[[u(v)]] = \varepsilon_{BC}$. Also if x is the variable in τ of type A_{n-1}, then

$$[[\lambda x \,.\, \tau]] : A_0 \times \ldots \times A_{n-2} \to (A_{n-1} \to B),$$

and in fact $[[\lambda x \,.\, \tau]] = \Lambda[[\tau]]$. (Warning: for other of the variables that are not the last mentioned, it is not so easy to write down the answer: some permutations of the products have to be introduced.)

The two characteristic equations for ε and Λ in the axioms for a c.c.c. have very familiar translations:

$$(\lambda y \,.\, h(x,y))(y') = h(x,y') \text{ and}$$

$$\lambda y \,.\, k(x)(y) = k(x),$$

where the type of x is A, the type of y and of y′ is B. (That is to say, the $[[\cdot]]$-meaning of the two sides of the equation is the same map in the category.) Of course this all has to be defined more rigorously, but I hope I have conveyed the main part of the idea of Lambek's correspondence. A typed λ-calculus (with pairs, products, and function spaces) is just another notation for a c.c.c.

No, we have to be more specific than that. Take a c.c.c. How does it correspond to a theory (of functions)? The domains of the category are the *types* of the theory, and they are structured by the $\mathbb{1}$, $(A \times B)$, $(A \to B)$ operations on types. Things like $\circ$, $0, p, q$, $\langle \cdot, \cdot \rangle$, ε, Λ stand for *logical constants* (or operators) - with type subscripts as needed. Maps $f : A \to B$ of the category stand for the *non-logical constants* of the theory. The equations $f = g$ between maps are the *assertions* of the theory. The logical axioms are those special equations common to all c.c.c.'s - the other equations are those that just happen to work out in the category. From this point of view the theory has *no free variables:* all assertions are written with constant terms. Equations with free variables can be construed as functional equations (by a heavy use of Λ).

Conversely, a more conventional typed λ-calculus is an equational theory with both the familiar logical axioms as well as with non-logical axioms as desired. The equations can involve free variables. Aside from the usual deduction rules for equality, we *must* employ the extensionality rule

$$\frac{\tau = \sigma}{\lambda x \,.\, \tau = \lambda x \,.\sigma}$$

A category is formed from the types (which are *given* as closed under $\mathbb{1}$, $(A \times B)$, $(A \to B)$). The terms all have unambiguous types, and they are divided into equivalence classes by the theory. As the maps of the category we take the equivalence classes $[\lambda x \,.\, \tau]$ where the term τ of type B has at most the variable x of type A free, and we write $[\lambda x \,.\, \tau] : A \to B$. Of course

$$1_A = [\lambda x \,.\, x], \text{ and}$$

$$[\lambda y \,.\, \sigma] \circ [\lambda x \,.\, \tau] = [\lambda x \,.\, \sigma(\tau/y)]$$

where τ is substituted for y in σ. We must verify that a category is obtained - using the laws of λ-calculus. And we must see that if we go back again to a λ-calculus from the category we have essentially the same theory.

A c.c.c. (or typed λ-calculus - with non-logical axioms) is a satisfactory (extensional) theory of functions because all we have built into the theory is the idea of the product and the function space. The axioms set down are just those needed to make this structuring explicit.

The reason that category theory is a convenient way to formalize this definition is that starting from the especially elementary concept of maps under composition, we can see that we have done *nothing more* than close up under products and function spaces. λ-calculus, then, becomes mostly a notational device for setting down our functional equations. At least for typed λ-calculus, we can see in this way that it is harmless.

The typed λ-calculus is even more harmless than these last remarks suggested. By the well-known Yoneda embedding, one can prove that an arbitrary (small) category has a full and faithful embedding into a c.c.c. This means that starting with a given category and its maps, there is a precise sense in which it is consistent to close up under products and function spaces. No new maps are added co the given category; no new equations between the given maps are imposed by the adjunction of higher types. One can say even more than this about relative consistency, but the remark is best deferred to Section 4, where references to the proof are provided.

A final remark must be added to this section to clear up a possible confusion between *theories* and *models*.

Up to this point we have been talking about *theories*. In many systems of logic models can be described by theories: every model has a "diagram" involving constants for all the "elements" of the model and taking as axioms all statements in the language "true" about the model. It depends on the nature of the logic how hard it is to show that every "consistent" theory has a model.

In the case of a c.c.c., a domain A could be said to have an "element" if there is a map $a : \mathbb{1} \to A$. The question is: *are there enough elements*? Suppose $f,g : A \to B$ are two maps. If $a : \mathbb{1} \to A$, then $f \circ a : \mathbb{1} \to B$; so in a certain sense maps in the category behave as functions on elements. (This is not an original suggestion but is one well known in category theory.) It is natural to ask whether, if $f \circ a = g \circ a$ for all $a : \mathbb{1} \to A$, then $f = g$. If this is true in a c.c.c., then it is said to have "enough" elements or to be *concrete*. In case it is concrete, domains can be identified with sets, maps with functions, products $A \times B$ in the category with the corresponding cartesian product of the sets (ask: which $c : \mathbb{1} \to A \times B$?), and function spaces $(A \to B)$ with spaces of actual functions (because there is a one-one correspondence between maps $f : A \to B$ and elements $e : \mathbb{1} \to (A \to B)$).

For a theory in the form of a c.c.c., to ask whether it has a (non-trivial) model is to ask whether it can be expanded to a concrete c.c.c. by the adjunction of elements (and other maps and additional equations, but no new domains) which is non-trivial in the sense of not making all domains isomorphic to $\mathbb{1}$.

An answer - though perhaps a rather formal one - is supplied by the method of adjunction of indeterminates $x : \mathbb{1} \to A$ presented in Lambek's paper (this volume). We just have to adjoin infinitely many for each domain, one after the other. Each polynomial involves only finitely many indeterminates. But the results stated by Lambek (esp. Corollary to Theorem 2) show us at once that this expanded category is concrete. The idea is really just like the idea of having "free algebras" for any equational theory. (In λ-calculus an algebraic equation that is regarded as universally quantified, say $x + y = y + x$, is replaced by the functional equation

$$\lambda x \lambda y \,.\, x + y = \lambda x \lambda y \,.\, y + x.)$$

More thoughts on concreteness will be brought out in Section 4.

That is the (easy) passage from theories to (certain) models. But remember, a theory is *not* a model: the maps in a given c.c.c. are not concrete maps, they are just the *definable* maps in the language of the theory, and the equations between them are the "theorems" of the theory. It is no surprise that a given theory may not have enough *definable* elements: we may need to expand the stock of elements in order to have a model. For a c.c.c. we find we can. So far, so good; and this is the (known) story of typed λ-calculus.

3. "TYPE-FREE" DOMAINS

In the paper of Lambek, the *analogy* between typed and type-free is illustrated (in the obvious way), but no real *connection* or *relation* is established. This we shall now do, and the relationship will be deepened in the next section.

In the first place, we shall only consider the λ-calculus (or λK-calculus) and not the λη-calculus; the latter can be regarded as a special case. What is needed is a notion of domain appropriate to the interpretation of the "type-free" calculus.

In a category, a *retraction* between two domains A and B is a pair of maps $i : A \to B$ and $j : B \to A$ where $j \circ i = 1_A$. Regard A as the "smaller" domain; it is injected into B, and B is surjectively mapped onto A. The notion shares qualities, then, of A being both a *subspace* of B and at the same time a *quotient*. But the injection and surjection have to be related.

Now, suppose that in a cartesian closed category a domain U satisfies the condition that the function space $(U \to U)$ is a retract of the domain U itself. (This is always so for $U = \mathbb{1}$, but we seek non-trivial examples.) Let the retraction maps be $i : (U \to U) \to U$ and $j : U \to (U \to U)$. Then U (as it sits in its category) gives us an interpretation of the type-free calculus, which we now explain.

Let the type-free terms be constructed in the usual way from variables $x,y,z,\ldots$ by means of application and λ-abstraction. Think of all variables as being of type U and define a translation τ^* from untyped terms to typed terms so that

$$x^* = x,$$
$$(\tau(\sigma))^* = j(\tau^*)(\sigma^*),$$
$$(\lambda x \,.\, \tau)^* = i(\lambda x \,.\, \tau^*).$$

We intend this in such a way that τ^* is always of type U. The type-free theory (determined by the category, the domain U, and by the choice of i and j) has as its assertions exactly those equations $\tau = \sigma$ where $\tau^* = \sigma^*$ in the category. The theory satisfies (α), (β)-conversion, all the rules of equality, and the rule (ξ): from $\tau = \sigma$ to deduce $\lambda x \,.\, \tau = \lambda x \,.\, \sigma$. This much is surely obvious to anyone reading Lambek's paper.

What I would like to point out here is the *converse*: given any type-free theory, there is a c.c.c. and a domain U (with a suitable retraction pair i,j) so that the above interpretation gives *exactly the same* type-free theory. Consequently, nothing is lost in considering type-free theories just as *special parts* of typed theories. I do not find this result mentioned by Lambek.

The proof is elementary. Let the domains for the category be the λ-terms A, without free variables, for which we can prove in the theory:

$$A = \lambda x \,.\, A(A(x)).$$

The maps $f : A \to B$ are terms f without free variables for which we can prove

$$f = \lambda x . B(f(A(x))).$$

The equations between maps are the equations we can prove in the theory. [Actually, it might be better to construe maps as triples (A,f,B), but never mind.] It is not hard to show that this is a category where

$$1_A = A, \text{ and}$$

$$f \circ g = \lambda x . f(g(x)).$$

[More properly spoken, the maps should be equivalence classes of terms based on the equations of theory, but never mind.]

To show this construction gives a c.c.c. we need to define:

$$A \times B = \lambda u \lambda z . z(A(u(\lambda x \lambda y . x)))(B(u(\lambda x \lambda y . y))),$$

where

$$p_{AB} = \lambda u . (A \times B)(u)(\lambda x \lambda y . x),$$

$$q_{AB} = \lambda u . (A \times B)(u)(\lambda x \lambda y . y),$$

and if $f : C \to A$ and $g : C \to B$, then

$$\langle f,g \rangle = \lambda t \lambda z . z(f(t))(g(t)).$$

All of this is based on the familiar pairing functions of λ-calculus.

For function spaces, we define:

$$(A \to B) = \lambda f. B \circ f \circ A$$

where

$$\varepsilon_{BC} = \lambda u. C(u(\lambda x \lambda y.x)(B(u(\lambda x \lambda y.y)))),$$

and

$$\Lambda_{ABC}\, h = \lambda x \lambda y.h(\lambda z.z(x)(y)),$$

provided $h : (A \times B) \to C$.

There are a jolly lot of equations to verify, but the work is all straight-forward conversion. The method of retracts as a c.c.c. has in any case been exposed before with respect to the $P\omega$ model in Scott (1976). Note here, however, we are to verify the required equations in a theory (not a model) making use of nothing but the "logical" axioms of λ-calculus.

It remains to identify the domain U in the constructed category. We define:

$$U = \lambda x.\ x$$

Clearly

$$U = \lambda x.\ U(U(x)) = U \circ U .$$

Note that every A in the category is a retract of U; indeed, for retractions define:

$$A : A \to U \quad \text{and} \quad A : U \to A \quad \text{and}$$

$$A \circ A = A = 1_A .$$

We thus speak of these A's also as retractions. We can write:

$$(U \to U) = \lambda f \lambda x.\ f(x),$$

and it is thus easy to verify now that $(U \to U)$ is a retraction of U. As U is in fact the identity function, the reinterpretation *via* U of the type-free calculus will obviously translate every term into itself.

I just note in writing down the definition of the c.c.c., I forgot to define $\mathbf{1}$ - because it is so dull, I suppose! For this we have to map everything onto a constant:

$$\mathbf{1} = \lambda u\ \lambda x.x, \quad \text{and}$$

$$0 = \mathbf{1}.$$

I think the calculations suggested provide an argument that type-free λ-calculus takes *second place* to typed λ-calculus - foundationally speaking. Type-free domains are *special kinds* of types. As I have said before in other writings, to get

$(U \to U)$ inside U, we have to pass to an *infinite type*. I thought this was made very clear in the so-called D_∞-construction. The category of *continuous lattices and continuous functions* is a c.c.c. Starting with any domain D_0, in that category the sequence of types D_n where $D_{n+1} = (D_n \to D_n)$ has a certain limit D_∞ with D_0 (and all the D_n's) as retracts, and with $(D_\infty \to D_\infty)$ not only a retract but an isomorph of D_∞. That is one choice of an U, and I showed many variations are possible for other type-free domains U in this one category.

We hasten to note that in the c.c.c. of *sets and arbitrary functions*, a non-trivial domain U with $(U \to U)$ a retract is *impossible* (by cardinality considerations). This means that not all c.c.c. lead directly to interpretations of the type-free theory. Hence, we must conclude, *the typed theory is the more general one, and the prior one*.

Such a conclusion will not be welcome, however. The type-free theory from our experience *seems* general enough. Even though we have shown two good ways of relating the two kinds of theories, we would like something more. We do not want just *some* c.c.c. related to a given type-free theory, but we would like to find a relation that achieved *any desired* c.c.c., provided we cook up the type-free one properly. This problem is the topic of the next section.

Before we turn to this new relationship, a word about *models* of the type-free calculus would be to the point. There is considerable discussion of the notion of a model in Hindley and Longo (1980) and Barendregt (1980) (where other references are given). We should state how this all fits in with the present view.

When presenting a *theory* in the usual λ-notation, free variables are permitted as well as full use of the rule (ξ). *But*, when thinking of elements (relative to a theory) only terms (better: equivalence classes of terms) *without free variables* should be considered. As is known from many examples, there may not be enough of them. This can of course be so even if we allow in our language many non-logical constants. What does "enough" mean? Well, if f and g are closed terms, it may be that f(a) = g(a) is provable for *all* closed a, but f = g is not provable. The fact that this happens for some theories should come as no surprise. (For the explicit examples consult Barendregt (1980).)

The remedy is to adjoin indeterminates (constants without new axioms) until "enough" is reached. (A proof is also found in Barendregt (1980).) As with the typed calculus, every theory has a model which satisfies exactly the same equations as are provable in the theory (one might call it a conservative model).

The notion of a λ-model has not struck people as quite satisfactory because the extensionality principle in the "enough" clause is not very algebraic. A suggestion of mine is mentioned in the cited references, but I think it would be useful to recast the idea in the light of the present discussion.

In typed λ-calculus, the categorical formulation is one way of eliminating all use of variables. In type-free λ-calculus, the usual plan is to use the combinators - and the plan leads to awfully long formulae. Let us not try to give a variable-free formulation, but talk in terms of first-order models. What is unalgebraic in the model definition is the λ-operator, since a bound variable is of the essence of the use of λ. So let us replace λ by the combinators in the usual way. We take S and K as primitive, and a λ-model is (at least), a structure of the form $\langle U, \cdot(\cdot), S, K\rangle$, with a domain, a binary operation, and two distinguished constants. The problem is: *what are the axioms?* Clearly we want:

$$(*)\begin{cases} K(x)(y) = x, \text{ and} \\ S(u)(v)(x) = u(x)(v(x)), \end{cases}$$

as usual. But these are not sufficient to express extensionality, which in λ-notation reads:

$$\forall x.\ \tau = \sigma \rightarrow \lambda x.\tau = \lambda x.\sigma$$

If we convert out the variable x, we are tempted to write:

$$\forall x.\ f(x) = g(x) \rightarrow f = g.$$

But this is too strong. (It corresponds to $U = (U \rightarrow U)$ rather than the weaker: $(U \rightarrow U)$ is a retract of U.) If we wrote:

$$\forall x.\ f(x) = g(x) \rightarrow \lambda x.f(x) = \lambda x.g(x),$$

the statement would at least be correct - even if containing the unwanted λ. Well, we just have to define this λ in terms of S and K. Introduce the standard definitions:

$$I = S(K)(K)$$

$$B = S(K(S))(K).$$

Then (with λ-notation)

$$\lambda x.\ f(x) = B(I)(f).$$

So the desired axiom now reads

$$(**)\quad \forall x.\ f(x) = g(x) \rightarrow B(I)(f) = B(I)(g)$$

We are not quite done, however. We want S and K to correspond to λ-expressions (eventually), so we need an axiom which makes them suitably unique. Now we note intuitively that

$$\lambda x_0 \lambda x_1 \ldots \lambda x_{n-1}.f(x_0)(x_1)\ldots(x_{n-1}) = B^n(I)(f).$$

Thus, what we need to say is:

$$(***)\begin{cases} S = B(B(B(I)))(S), \text{ and} \\ K = B(B(I))(K) \end{cases}$$

To see that (*), (**), and (***) are adequate, we note first that

$$B(I)(f)(x) = f(x)$$

by (*). From (**) it then follows that

$$B(I)(B(I)(f)) = B(I)(f).$$

This means we can reformulate (**) as:

$$(**_1)\quad f = B(I)(f) \wedge g = B(I)(g) \wedge \forall x. f(x) = g(x) \rightarrow f = g.$$

[This does not seem to be equivalent to (**) unless we have the equation about $B(I)(B(I)(f))$ just noted - the retraction equation.] We now generalize $(**_1)$ to n variables:

$$(**_n)\left\{\begin{array}{l} f = B^n(I)(f) \wedge g = B^n(I)(g) \wedge \\ \forall x_0, x_1, \ldots, x_{n-1}. f(x_0)(x_1)\ldots(x_{n-1}) = g(x_0)(x_1)\ldots(x_{n-1}) \\ \rightarrow f = g. \end{array}\right.$$

If we prove this, then by (***) we see that the original axioms (*) uniquely determine S and K; further we have the uniqueness required to define $\lambda x.\tau$ for any term (cf. the references cited).

To establish $(**_n)$, we need some lemmas. From (*) and (***) and the definitions, we can easily prove:

$$S(u) = B^2(I)(S(u))$$
$$S(u)(v) = B(I)(S(u)(v))$$
$$B(u) = S(K(u)).$$

We then establish for $n \geq 1$:

$$B(I)(B^n(I)(f)) = B^n(I)(f),$$

because $B^n(I)(f)$ has the form $S(u)(v)$. Suppose then that, e.g.

$$\forall x,y,z.\ f(x)(y)(z) = g(x)(y)(z).$$

By (**) we find:

$$\forall x,y.\ B(I)(f(x)(y)) = B(I)(g(x)(y)).$$

This can be rewritten as

$$\forall x,y.\ B^2(I)(f(x))(y) = B^2(I)(g(x))(y)\,.$$

But again by (**) we find:

$$\forall x.\ B(I)(B^2(I)(f(x))) = B(I)(B^2(I)(f(x))).$$

By the lemma, drop the B(I). Throw on another B, use (**), drop off the B(I), and get:

$$B^3(I)(f) = B^3(I)(g).$$

The method is perfectly general and proves ($**_n$).

The import of this axiomatization is that B(I) is the retraction of the universe U onto (U → U) and $B^n(I)$ retracts onto

$$\underbrace{(U \to (U \to (U \to \ldots\ (U}_{\text{n times}} \to U)\ldots))).$$

We need (***) to show, e.g.:

$$S : U \to (U \to (U \to U)).$$

We need ($**_n$) to show that the maps in these function spaces are *uniquely* determined by their values.

We have just been speaking in terms of models; but the calculations just carried out were formal. The axiomatic question, then, is: what is the relationship between the equational theories and the first-order theories? We shall now see the relation is a close one - even if the logic is allowed to go beyond the first order.

4. A RÔLE FOR INTUITIONISTIC LOGIC.

The (rather cheap) method of adjoining indeterminates proves that every typed or untyped theory of λ-calculus has an extensional model. This can also be put as a conservative extension result for theories: a λ-theory is an equational theory, and every such equational theory can be expanded to a first-order theory *without* forcing any new equations on us. In the untyped case, the style of first-order theory is that of axioms (*), (**), (***) of the previous section. These are the "logical" axioms (i.e. common to all such theories); the non-logical axioms would be all the equations between closed λ-terms demanded by whatever equational theory we started with - and these special equations could involve special "non-logical" constants.

In the typed case, we would get a many-sorted theory with a sort for each domain in the given category. As we have already pointed out, an untyped theory can always be reformulated as a typed theory by the method of retracts. So we now concentrate on typed theories - that is to say, cartesian closed categories.

But the writing down of *first-order* theories is not all that interesting: we clearly have nice axioms for a theory of functions, but first-order theories do not impress us as being very categorical. Such theories do not really capture the idea of the "arbitrary" function. We began our discussion with set theory, where the intention was that function spaces did really contain "all" functions - they did not just appear as an "algebra" of functions. Leaving aside for the moment the (philosophical) question of whether the desire for the ALL is a rational one, we can ask the (formal) question of whether there is a conservative extension result for *higher-order* theories. Surprisingly, category theorists have known the answer for some time.

Now we cannot hope to embed the theory of a typed λ-calculus into a *classical* higher-order theory with a full comprehension axiom of the form

$$\forall x : A\ \exists! y : B.\varphi(x,y) \rightarrow \exists f : A \rightarrow B \forall x : A.\ \varphi(x,f(x)).$$

Because in higher-order logic we can prove Cantor's Theorem which implies that the only type U which has a surjective map $j : U \rightarrow (U \rightarrow U)$ is the one-element type. Thus, if a typed theory had such a type (and we know many), then the adding of the standard higher-order axioms (where we construe $(A \to B)$ as the *total* function space of *all* functions) would not at all be conservative. Something else has to be tried, and the answer is higher-order *intuitionistic* logic.

As we shall now have to consider more than one category, let me call our given c.c.c. the category C. To fix ideas, the constructions to be carried out will be done in ordinary set theory - with classical logic! The models obtained, however, will only satisfy intuitionistic logic. The obvious lack of harmony can be repaired, but it would take too much explanation here. Moreover, we are also going to assume that the given category C is a *set*. This is not much of a restriction, since we were thinking of C as a *theory* and usually a theory has a limited number of symbols in any case.

Before saying where the intuitionistic logic comes from, let me give the *construction*. Let S be the category of all sets and arbitrary functions; we know it is a c.c.c. The construction we need here is the well-known one of the *functor category* $S^{C^{op}}$ of all contravariant functors from C into S with the natural transformations as the maps - full definitions follow. The *result* is that the functor category is a model for higher-order intuitionistic logic, in particular it is a c.c.c.; moreover the original category C has a full and faithful embedding in $S^{C^{op}}$, and this shows the conservative extension property. So much for the outline of the method, now for the details. Needless to say this represents a *very early* chapter in topos theory; it should be more widely known.

What is a *contravariant functor?* It is a mapping $F : C \to S$ that associates to every domain A of C a set $F(A)$ of S and to every map $f : B \to A$ of C a function $F(f) : F(A) \to F(B)$ (and note the change of order!) so that:

$F(1_A) = 1_{F(A)}$, and

$F(f \circ g) = F(g) \circ F(f)$,

provided $f : B \to A$ and $g : C \to B$ in $\mathcal{C}$. It was one of the major early insights of category theory to see that the functors form a category in themselves. What is needed is a definition of transformation between functors. We call such maps $\nu : F \to G$ "natural transformations" for reasons explained in category theory books.

What is a *natural transformation* $\nu : F \to G$? It is an association with every domain A of $\mathcal{C}$ of a function $\nu_A : F(A) \to G(A)$ so that whenever $f : B \to A$ in $\mathcal{C}$, then the following diagram commutes in $\mathcal{S}$:

$$\begin{tikzcd} F(A) \arrow[r, "F(f)"] \arrow[d, "\nu_A"'] & F(B) \arrow[d, "\nu_B"] \\ G(A) \arrow[r, "G(f)"] & G(B) \end{tikzcd}$$

This means $\nu_B \circ F(f) = G(f) \circ \nu_A$. An example will help explain this.

For each C of $\mathcal{C}$, let

$H_C(A) = \{h \mid h : A \to C\}$

and if $f : B \to A$ in $\mathcal{C}$, let $H_C(f)$ be the map taking $h \in H_C(A)$ into $h \circ f \in H_C(B)$. It is easy to show H_C is a (contravariant) functor. It is often called the *representable functor* (corresponding to C), and we shall see that it is very "representative".

Now let $g : C \to D$ in $\mathcal{C}$. There is a natural transformation $H_g : H_C \to H_D$; because, for each $h \in H_C(A)$ we can map it to $g \circ h \in H_D(A)$, naturally. The composite map for $f : B \to A$ takes $h \in H_C(A)$ into $g \circ h \circ f \in H_D(B)$, and there are two equal ways to calculate it owing to the associativity of composition (in C); that is why the necessary diagram commutes.

Not only are the H_C pleasant functors with cooperative natural transformations between them, but the by now classic *Yoneda Lemma* proves for us that the *only* natural transformations $\nu : H_C \to H_D$ are those of the form H_g for some $g : C \to D$. If we remark that if $k : D \to E$, then

$$H_k \circ H_g = H_{k \circ g} ,$$

this shows us that $H : C \to S^{C^{op}}$ is a (*covariant*) functor between these categories. Of course H_C uniquely determines C and H_g determines g, so we conclude from this and the *Yoneda Lemma* that H is a full and faithful embedding of C into the functor category (all of this on p.2 of Johnstone (1977)!).

All of this discussion is "abstract nonsense" in the sense that its validity is perfectly general for *any* category C. If we assume that C is a c.c.c., then we can say more. The point is that $S^{C^{op}}$ is a very powerful category. For example, it is always a c.c.c. even if C is not. The cartesian closed structure of the functor category is obtained through the following definitions.

Before getting down to details, however, some more-vivid terminology might help. Think of a functor U in $S^{C^{op}}$, following Lawvere, as a "variable domain". That is to say, for each $A \in C$ we have an associated domain (set) $U_A = U(A)$. The maps $f : B \to A$ in C give us transitions between "stages" A and "later" stages B; and each such transition "restricts" elements in U_A to elements in U_B "along" the map f. To save writing, let us set $a \restriction f = (Uf)(a)$ when the functor U is understood. That U is indeed a functor comes down to these equations:

$$a \restriction 1_A = a \quad \text{and} \quad (a \restriction f) \restriction g = a \restriction (f \circ g),$$

where $f : B \to A$ and $g : C \to B$ in $\mathcal{C}$. To define a functor U, then, we just have to give the domains and the restrictions. For example the unit functor $\mathbb{1}$ has $\mathbb{1}_A = \{0\}$, a one-point set, and all restrictions constant $0 \restriction f = 0$. And all natural transformations into $\mathbb{1}$ are constant.

Now suppose U and V are two functors. We define $U \times V$ so that for all A in $\mathcal{C}$

$$(U \times V)_A = U_A \times V_A$$

and whenever $a \in U_A$ and $b \in U_A$ and $f : B \to A$, then

$$(a,b) \restriction f = (a \restriction f,\ b \restriction f).$$

(Note that the restriction symbol above is used in three different senses.) The natural transformations $p : U \times V \to U$ and $q : U \times V \to V$ have obvious pointwise definitions (e.g. $p_A = p_{U_A V_A} : U_A \times V_A \to U_A$) and they clearly commute with restrictions. Similarly, if $\mu : W \to U$ and $\nu : W \to V$ are given natural transformations, then $<\mu,\nu> : W \to U \times V$ is also defined pointwise:

$$<\mu,\nu>_A = <\mu_A,\nu_A> : W_A \to U_A \times V_A.$$

Again it is obvious that these maps commute with restrictions, so $<\mu,\nu>$ is natural. As all of this is pointwise; the verification of such equations as $p \circ <\mu,\nu> = \mu$ is easy.

Again suppose U and V are given. In defining $(U \to V)$, we cannot be quite as pointwise. That is, $(U \to V)_A$ cannot be taken simply as the set of functions $(U_A \to V_A)$, the function space in sets. The reason, roughly, is that when we have a function at *one* stage, we also have to know how it restricts at *later* stages; a simple mapping from U_A into V_A does not give us enough information for that. When $f : B \to A$, restriction on U_A maps into U_B; this is the wrong direction for us to be able to pass from an arbitrary function defined on U_A to one defined on U_B. So an element of $(U \to V)_A$ has to be a whole *family* of functions

$$\varphi_f : U_B \to V_B ,$$

one for each $f : B \to A$. (Note: A is fixed, f and B are variable.) Moreover, we must assume that all is harmonious with restrictions: $c \restriction g = \phi_{f\circ g}\,(b \restriction g)$ whenever $c = \varphi_f(b)$, for $b,c \in U_B$ and $g : C \to B$ in $\mathcal{C}$. In words, φ_{1_A} is the "present" function; while φ_f is what becomes of it in the "future", supposing time evolves along f. Now families φ of this kind in $(U \to V)_A$ have to be restricted. By what we just said, the following is more or less forced upon us:

$$(\varphi \restriction f)_g = \varphi_{f\circ g}$$

where $f : B \to A$ and $g : C \to B$, so that $\varphi \restriction f$ is a family in $(U \to V)_B$.

That defines $(U \to V)$ as a functor. To have $S^{\mathcal{C}^{op}}$ be a c.c.c., certain maps ε and Λ are, alas, still required. The evaluation map $\varepsilon : ((V \to W) \times V) \to W$ is fortunately rather clear (as a natural transformation). Suppose $\varphi \in (V \to W)_A$ and $a \in V_A$. Then

$$\varepsilon_A(\varphi,a) = \varphi_{1_A}(a),$$

so $\varepsilon_A : ((V \to W)_A \times V_A) \to W_A$. If $f : B \to A$, then

$$\begin{aligned}\varepsilon_A(\varphi,a) \restriction f &= \varphi_{1_A}(a) \restriction f \\ &= \varphi_f(a \restriction f) \\ &= (\varphi \restriction f)_{1_B}(a \restriction f) \\ &= \varepsilon_B(\varphi \restriction f, a \restriction f)\end{aligned}$$

This proves that ε is natural.

Next, suppose $\psi : U \times V \to W$ is natural. Define $\Lambda\psi : U \to (V \to W)$ by

$$(\Lambda\psi)_A : U_A \to (V \to W)_A$$

where for $a \in U_A$, $b \in V_B$, and $f : B \to A$ we have:

$$(\Lambda\psi)_A(a)_f(b) = \psi_B(a \upharpoonright f, b).$$

To show $\Lambda\psi$ is natural, we must calculate:

$$\begin{aligned}((\Lambda\psi)_A(a) \upharpoonright f)_g(c) &= (\Lambda\psi)_A(a)_{f\circ g}(c)\\ &= \psi_C(a \upharpoonright f \upharpoonright g, c)\\ &= (\Lambda\psi)_B(a \upharpoonright f)_g(c)\end{aligned}$$

for $a \in U_A$, $f : B \to A$, $g : C \to B$, $c \in U_C$. It follows that

$$(\Lambda\psi)_A(a) \upharpoonright f = (\Lambda\psi)_B(a \upharpoonright f).$$

We have to leave to the reader the verification of the two basic equations of c.c.c.'s involving ε, Λ, p and q. As there was only one way that the definitions could be written, the verification is quite mechanical, however.

As I said before, the functor category is "powerful", and indeed it is much more than a c.c.c. For instance, we can define the analogue of the *power set* for arbitrary functors. For any U, let $(\mathcal{P}U)_A$ be the collection of all families S_f indexed by $f : B \to A$ where $S_f \subseteq U_B$ and such that $b \upharpoonright g \in S_{f\circ g}$ whenever $b \in S_f$ and $g : C \to B$ in $\mathcal{C}$. Restriction is defined by

$$(S \upharpoonright f)_g = S_{f\circ g}.$$

The significance of the power operator will become clear when we speak about higher-order logic.

Having seen why the functor category is a c.c.c., it is good to pause a moment to appreciate the difference between the elements $a \in U_A$ as sets and the "elements" of U in the categorical sense. If $\alpha : \mathbf{1} \to U$ is natural, it means that $\alpha_A : \mathbf{1}_A \to U_A$ in $\mathcal{S}$. Let $a_A = \alpha_A(0)$, then $a_A \in U_A$. If $f : B \to A$, then because α is natural we find:

$$
\begin{aligned}
a_A \upharpoonright f &= \alpha_A(0) \upharpoonright f \\
&= \alpha_B(0 \upharpoonright f) \\
&= \alpha_B(0) \\
&= a_B
\end{aligned}
$$

This is very strong indeed, since usually if $a \in U_A$ and $f_0, f_1 : B \to A$, there is no reason why $a \upharpoonright f_0 = a \upharpoonright f_1$. So the number of "elements" of U will very likely be rather small. (And, even worse, a_A has to be chosen for *all* A in $\mathcal{C}$.)

In the special case $\sigma : \mathbf{1} \to \mathcal{P}U$ we can simplify the choices out of $(\mathcal{P}U)_A$ even further. Write $S_A = \sigma_A(0)_{1_A}$, then $S_A \subseteq U_A$ for all A in $\mathcal{C}$. Moreover, when $f : B \to A$, then

$$
\begin{aligned}
S_B &= \sigma_B(0)_{1_B} \\
&= \sigma_B(0 \upharpoonright f)_{1_B} \\
&= (\sigma_A(0) \upharpoonright f)_{1_B} \\
&= \sigma_A(0)_f
\end{aligned}
$$

This means that $\sigma_A(0)$ is determined from the S_B's. And if they are chosen so that

$$b \in S_B \text{ implies } b \upharpoonright g \in S_C$$

whenever $g : C \to B$, then the σ_A so defined from them provides a natural transformation. Again, we see the elements of $\mathcal{P}U$ are rather special. We can say that elements of a functor provide information about the "global" nature of the functor; but this is far from determining it, for there can be considerable "local" activity that cannot be sensed globally. For example, the sets U_B can be empty for a long "time", only becoming non-empty in the "future". The functor U is non trivial, but it has no global elements.

We should also pause to see why the functor H maps C into a subcartesian closed category of $S^{C^{op}}$ (up to isomorphism). It is easy to check that the functors $H_A \times H_B$ and $H_{A\times B}$ are naturally isomorphic. We also have to do the same for $(H_A \to H_B)$ and $H_{A\to B}$. Consider an element of $(H_A \to H_B)_C$. It is a family of maps

$$\varphi_f : H_A(D) \to H_B(D),$$

for $f : D \to C$. In particular consider the standard maps $p : (C \times A) \to C$ and $q : (C \times A) \to A$. Then $\varphi_p(q) : (C \times A) \to B$. So, since C is a c.c.c., we find $\Lambda\varphi_p(q) \in (H_{A\to B})_C$. In the other direction, let $t : C \to (A \to B)$. Define τ_f for $f : D \to C$ by

$$(\tau_f)(g) = \varepsilon \circ \langle t \circ f, g \rangle$$

where $g : D \to A$. We see this lies in $H_B(D)$. Now

$$\varepsilon \circ \langle t \circ f, g\rangle \circ k = \varepsilon \circ \langle t \circ f \circ k, g \circ k\rangle$$

whenever $k : E \to D$. Thus,

$$(\tau_f)(g) 1 k = \tau_{f\circ k}(g 1 k).$$

This proves that the family τ_f lies in $(H_A \to B_B)_C$. It has to be left without proof that these two correspondences are inverse to one another and provide a natural isomorphism.

Well, this is a rather heavy construction starting from one little category C. The question: *what does it prove?* Why worry about the functor category? The answer is that the functors give an interpretation of higher-order logic, as we hinted earlier, and now we have to pay up and demonstrate how to construe logical formulae. The idea from topos theory when specialized to the functor category looks very much like Kripke models of intuitionistic logic - except that the "times" form a category C rather than just a partially ordered set, as has often been emphasized by Lawvere (see, e.g. Lawvere (1975)).

To make the logical language more definite, let us think of the domains A in $\mathcal{C}$ as being (in a one-one correspondence with) *type symbols*. Introduce *new* type symbols built from the ones in $\mathcal{C}$ (the "ground" types) by forming $\mathbb{1}$, $(T \times S)$, $(T \to S)$, $\mathcal{P}T$ for all type symbols. (Note: $A \times B$ in $\mathcal{C}$ is being distinguished from the type symbol $A \times B$. But the "meaning" of the symbol $A \times B$ will turn out to be something "isomorphic" to $A \times B$ in $\mathcal{C}$. The trouble is that the domain $A \times B$ does not in itself determine the A and the B; whereas the type symbol does.) We extend the notation H_A to H_T for any type symbol in the obvious way; that is, $H_{T\times S}$ is the product $H_T \times H_S$ in the functor category. This is the first step in treating the functor category as an interpretation of a higher-order theory.

Next we must imagine a logical language with a supply of variables of each type. Atomic formulae will be of these forms: $\perp$; $x = y$, where x and y have the same type; $y = fx$ where f is a constant symbol corresponding to a map $f : A \to B$ in $\mathcal{C}$ and x has type A and y type B; $z = (x,y)$ where x has type T, y type S, z type $T \times S$; $z = x(y)$, where z has type S, y has type T, and x type $(T \to S)$; $y \in x$, where y has type T and x type $\mathcal{P}T$. Atomic formulae are then made into compound formulae by the usual constructs: $\Phi \wedge \Psi$, $\Phi \vee \Psi$, $\Phi \to \Psi$, $\forall x.\ \Phi$, $\exists x.\ \Psi$.

Suppose A is a domain of $\mathcal{C}$, Φ is a formula, and s is a valuation of the free variables of Φ. We are going to define what Joyal-Reyes (1980) call the *forcing-satisfaction* relation $A \Vdash \Phi[s]$. The definition here will be in one respect simpler than theirs since the category $\mathcal{C}$ carries no topology; in another respect it is more complicated because we have the whole higher-order language. But the adaptation is straight forward. Before we can give the clauses, we must say what kind of a creature s is. We must make s *relative* to A in the first place. So if x has type T, then s(x) is to belong to the set $H_T(A)$. Now here are the clauses:

$A \Vdash \bot\,[s]$	iff	*false*
$A \Vdash x = y\,[s]$	iff	$s(x) = s(y)$
$A \Vdash y = fx\,[s]$	iff	$s(y) = f \circ s(x)$
$A \Vdash z = (x,y)\,[s]$	iff	$s(z) = (s(x), s(y))$
$A \Vdash z = x(y)\,[s]$	iff	$s(z) = s(x){\upharpoonright}_A(s(y))$
$A \Vdash y \in x\,[s]$	iff	$s(y) \in s(x){\upharpoonright}_A$

These were for the atomic cases and the reader should stop and think how the types are supposed to match. For the compound cases we have:

$A \Vdash [\Phi \wedge \Psi][s]$	iff	$A \Vdash \Phi[s]$ and $A \Vdash \Psi[s]$
$A \Vdash [\Phi \vee \Psi][s]$	iff	$A \Vdash \Phi[s]$ or $A \Vdash \Psi[s]$
$A \Vdash [\Phi \to \Psi][s]$	iff	whenever $f : B \to A$ and $B \Vdash \Phi[s \upharpoonright f]$, then $B \Vdash \Psi[s \upharpoonright f]$
$A \Vdash \forall x.\,\Phi[s]$	iff	whenever $f : B \to A$ and $b \in H_T(B)$, then $B \Vdash \Phi[s \upharpoonright f\ (b/x)]$
$A \Vdash \exists x.\,\Phi[s]$	iff	there is an $a \in H_T(A)$ such that $A \Vdash \Phi[s(a/x)]$

In the above, the notation $s(a/x)$ means the valuation is fixed so that a matches x; of course the type of x must be T. By $s \upharpoonright f$ we mean the valuation that matches $s(x) \upharpoonright f$ with each of the relevant variables x. In each case the restriction operation must be made appropriate to the functor H_T, where T is the type of x.

This is so much like Kripke models, the reader will have no problem in showing every intuitionistic quantificational validity Φ is such that $A \Vdash \Phi[s]$ for all A and all appropriate s.

We only have to take care that we remember that some ranges of variables can be empty (that a set $H_T(A)$ may be empty), and so the logic is the so-called "free" logic (cf. Scott (1979) for a discussion).

In order to verity the special axioms of higher-order logic, we need to remark first on what Joyal-Reyes call the "functorial" character of $\Vdash$:

$$\text{if } A \Vdash \Phi[s] \text{ and } f : B \to A, \text{ then } B \Vdash \Phi[s \upharpoonright f].$$

This, too, is a property familiar from Kripke models. It plays a direct rôle in the verification of the comprehension axiom:

$$\forall u_0,\ldots,u_{n-1}\ \exists x \forall y\ [y \in x \leftrightarrow \Phi]$$

where the free variables of Φ are among $u_0,\ldots,u_{n-1},y$ and x is a new variable not free in Φ of type PT where T is the type of y.

To show the above valid in the interpretation we only have to show that for every A of $\mathcal{C}$ and for all $b_0,\ldots,b_{n-1}$ in the $H_S(A)$ of the appropriate types S, there is an element $c \in H_{P(T)}(A)$ such that

$$A \Vdash \forall y[y \in x \leftrightarrow \Phi][s].$$

Here s is the valuation where $s(x) = c$ and $s(u_i) = b_i$. We have to define c. For each $f : B \to A$, let

$$c_f = \{t \in H_T(B) \mid B \Vdash \Phi[s \upharpoonright f(t/y)]\}$$

The functorial character of $\Vdash$ proves for us that $c \in H_{P(T)}(A)$. It is now easy to check from the clauses of the definition of $\Vdash$ that at A the above formula is indeed forced.

In a similar way we can verify the functional version of comprehension:

$$\forall x \exists y \forall z[z = y \leftrightarrow \Phi] \to \exists f \forall x,z[z = f(x) \leftrightarrow \Phi],$$

where we have x of type T, y and z of type S; f of type $(T \to S)$ and y not free in Φ. Again, the functorial character of $\Vdash$ connects with the way we had to define $(H_T \to H_S)$.

We also have to verity such extensionality properties as:

$$\forall f,g[\forall x,y[y = f(x) \leftrightarrow y = g(x)] \to f = g],$$

$$\forall x,y[\forall z[z \in x \leftrightarrow z \in y] \to x = y],$$

where the variables have to be given the appropriate types. But in defining the function spaces and the powersets in the functor category, we only put in just enough of a mapping or a set to get an appropriate functorial character. Hence, if two such objects are extensionally equal by the formulae above, they will be equal. This has to be spelled out via $\Vdash$, but it is not surprising.

The higher-order axioms for ordered pairs are obvious, and their satisfaction relates at once to the definition of product of functors. As for the embedding of $\mathcal{C}$ into the higher-order theory we find

$$\forall x,y[y = fx \leftrightarrow y = gx]$$

is valid if and only if $f = g$ in $\mathcal{C}$. Also, when $h = g \circ f$, we have as valid

$$\forall x,y,z[[y = fx \wedge z = gy] \to z = hx].$$

Further, functions like f are well defined:

$$\forall x\, \exists y\, .\, y = fx$$

There are many principles of identity that should be mentioned, but we will not write them down here. Among them we would also find the statements that there is a unique element of type $\mathbf{1}$ and all maps of type $(T \to \mathbf{1})$ are constant. (Perhaps the constant 0 should figure in the language, but it is not all that essential.)

As for questions of uniqueness, if the sentence

$$\forall x \exists y \forall z[z = y \leftrightarrow \Phi]$$

is valid (i.e. forced at all A), then provided x and y have ground types B and C, respectively, there is an $f : B \to C$ in $\mathcal{C}$ such that

$$\forall x,z[z = fx \leftrightarrow \Phi]$$

is valid too. The validities, chen give us an exact picture of $\mathcal{C}$ at the level of ground types: the higher-order theory is conservative over $\mathcal{C}$.

But the higher-order intuitionistic theory of the functor category is much more than just a conservative extension; it is a full-blown *higher-order* theory with full comprehension axioms. That is to say, we started out with a category $\mathcal{C}$ we regarded algebraically as a theory of functions. Well, the construction of the functor category shows us that we can indeed construe $\mathcal{C}$ and its maps as normal, everyday functions in a normal, everyday higher-order logic. This works as long as we agree to keep our logic intuitionistic. But experience with intuitionistic logic really shows that the system is a natural one and that it leads to very, very interesting theories. Even if $\mathcal{C}$ is a c.c.c., we can show that the embedding of $\mathcal{C}$ in the higher-order logic preserves all the cartesian closed structure, so that the function spaces in $\mathcal{C}$ really become spaces of all possible functions in the higher-order theory. The principles of λ-calculus are thus consequences of the standard logical axioms. This seems to me to establish complete harmony between (intuitionistic) logic and (typed) λ-calculus.

The next step in this investigation would be to see what other properties of the higher-order logic could be enforced and still preserve the conservative extension over the given category $\mathcal{C}$. The functor category is just a very first stage of the investigation: In topos theory the categories of *sheaves* result from putting a kind of modal operator into the logic, and making a reinterpretation of the logical connectives and quantifiers. The passage from $S^{C^{op}}$ is one of finding c.c.c. as cartesian closed subcategories of the functor category. There are many of them and many still contain C as a cartesian closed subcategory. So, there is much to look for, and - I am sure - much left to be discovered of definite logical interest.

5. TYPE-FREE DOMAINS REVISITED

Having made *any* given c.c.c. C "honest" as a theory of functions in higher-order logic, we can conclude from the method of retracts of Section 3 that any type-free λ-theory can similarly be made honest. Intuitionistic logic is very tolerant of types U where $(U \to U)$ is a retract; so tolerant in fact that *any* λ-theory can be embedded in a suitable higher-order logic. Self-application is no longer odd: it is something that may very well turn up when we weaken our logic to be intuitionistic but still require that functions spaces like $(U \to U)$ contain *all* functions.

This provides a certain kind of rescue for the type-free calculus, but the move fails to give it a universal rôle: the creators of the type-free theory hoped that such a universe U could be thought of as containing all the functions there were. We shall not try to go so far in the present context, but various constructions can be used to show that not only is it possible to have *one* such type-free domain, but it is always possible to find them being richer and richer and containing more and more functions. Only a sketch of the construction can be given here.

Suppose, for the sake of illustration, we have some types A, B, C that we happen to like, and that we are interested in the functions between them - possibly also in functions of the type $(A^2 \times B) \to C$, and similar multivariable types. We could probably work up a c.c.c. containing A, B, C and these functions, but the straight-forward construction would contain no type-free domains (cf. the category of sets and maps in ordinary logic). We need a new method. My first approach is to use the idea of continuous lattices. I do not want to go into a lot of detail (cf. Gierz et al. (1980) for just such details), but there is an easy definition that can be invoked at least to make the statements precise.

We shall employ what are not technically lattices but "half" lattices without unit elements (top elements). Fortunately we do not have to go into a long list of definitions, since I have been able to characterize them neatly as special *topological spaces*. They are in fact T_0-spaces (i.e. spaces where points are uniquely determined by their neighborhoods) D such that whenever X is a dense subspace of a topological space Y, and $f : X \to D$ is a continuous function, then f has a continuous extension $\bar{f} : Y \to D$. What I proved is that the category of such spaces D together with continuous maps between them is a c.c.c. (There are very many intriguing c.c.c.'s related to the category of topological spaces!) Let us employ the temporary name "injective" for these spaces.

As an example of injective spaces, consider one of our given types A, which for simplicity we construe just as a *set*. The injective space A_* corresponding to A results from adding one new point $*$. Or, if classical logic is not assumed, we take A_* as the space of subsets of A with *at most one element*. The topology is generated by sets of the form $\{x \in A_* \mid a \in x\}$ where $a \in A$. Thus, a function $f : X \to A_*$ is *continuous* iff $\{x \in X \mid a \in f(x)\}$ is *open* in X for each $a \in A$. Now if $X \subseteq Y$ as a dense subspace, we have only to define

$$\bar{f}(y) = \bigcup \{\{a \in A \mid \forall x \in N \cap X.\ a \in f(x)\} \mid y \in N\},$$

where N ranges over the open sets of Y. Because every non-empty open set has a non-empty intersection with X, it follows that $\bar{f} : Y \to A_*$. To prove $\bar{f}$ continuous, we remark that $\{y \in Y \mid a \in \bar{f}(y)\}$ is the largest open subset of Y whose intersection with X gives $\{x \in X \mid a \in f(x)\}$. It is also easy to calculate that $\bar{f}$ extends f. So A_* is injective. Note, too, that A may be regarded as a dense subspace of A_* if we map a to $\{a\}$. Hence, every function $g : A \to B$ has a unique counterpart $g_* : A_* \to B_*$ so that the "restriction" of g_* to A gives g back again (indeed $g_*(\{a\}) = \{g(a)\}$). This really means that the $*$-construction is a faithful functor from the category of our sets A, B, C into the category of injective spaces and continuous functions.

But now we can apply my construction of λ-calculus models to find an injective space U which, in the category of injective spaces, has $(U \to U)$ as a (continuous) retract *and* in addition has A_*, B_*, and C_* as retracts. (In fact, for those who know the method, we solve the domain equation

$$U = A_* \times B_* \times C_* \times (U \to U),$$

where of course a factor is always a retract of a product; because in the category of injective spaces the one-point space is a retract of every space.) This idea could be extended to obtain any given *set* of injective spaces as retracts of a *single* space U.

Next we invoke the plan of the previous section using as the category $\mathcal{C}$ the retracts of U (and continuous functions), which we can regard as a small category (as a set). The functor category has all higher-order logic as well as a full and faithful picture of $\mathcal{C}$. We are definitely going to take advantage of the higher-order structure in looking at subtypes of the functors H_V where V is a domain in $\mathcal{C}$ - more precisely, we will look at the category generated by certain of these subtypes or subfunctors in the functor category.

In the first place, consider H_{A_*}. Let K_A be the functor where $K_A(V)$ is just the set of all continuous functions $f : V \to A_*$ where for some $a \in A$, we have $a \in f(x)$ for all $x \in V$. (The retracts of U are simply being regarded as injective spaces, and we do not distinguish between A_* as a constructed space based on the set A and as a retract of U.) The restriction operation $\restriction f : K_A(V) \to K_A(W)$ is the one for the functor H_{A_*} with the domain cut down: K_A is a *subfunctor* of H_{A_*}. We can think of the maps in $K_A(V)$ as being the *constant* maps with values in A. So what then can we have for natural transformations $\nu : K_A \to K_B$, the maps in the functor category? Well, imagine one. Now if $a \in A$, we can take the appropriate constant map $k_a \in K_A(\mathbb{1})$, where $\mathbb{1}$ is the one-element space. Then $\nu(k_a) \in K_B(\mathbb{1})$. But this too is a constant map and determines a unique $b \in B$; so ν defines a function $|\nu| : A \to B$. And, since every constant map factors through the one-element space $\mathbb{1}$, the map $|\nu|$ uniquely determines ν. But *any* map from A into B can be made to turn up in this way by trivially fooling around with constants. We conclude, therefore, that K_A as a functor - of our given sets A - is a full and faithful embedding.

What have we done? First, starting in sets - or perhaps, better, in higher-order logic - we found (or gave ourselves) a category of types we liked. To be more definite, they could have been unioned together so they were all subtypes (subsets) of a single set, V, say. We then embedded faithfully this category of sets and maps into the category of injective spaces *via* the very elementary A_* construction. Of course, the spaces A_* are very special, so the "universal" space U with $(U \to U)$ as a retract is much more messy than V_*. But the category of retracts of U contains all the maps between the A_*. Finally this category of retracts is fully and faithfully embedded in the

the functor category. The latter has the advantage of subtypes in profusion, so we were able to recapture the *original* category of subsets of V as a full and faithful subcategory of the functor category.

And having done all this, what have we bought? Well, the (pictures of) the A's were subtypes of the A_*'s which are both retracts and subtypes of U, and in the functor category U is a model for the untyped λ-calculus. So that means that starting with our original notion of function, we have - in the logic of the functor category - consistently been able to assume that there are types giving models for the "type-free" λ-calculus, and further, that these types are rich enough to contain our original category in a full and faithful way. In more detail: the new logic allows us to think of A and B as subtypes of U, where $(U \to U)$ is a retract, so that *any* function from A into B is the result of restricting a function in $(U \to U)$ down to the subset A. *Warning*: this does not hold for *all* subtypes of U, the A, B, C were given *in advance* and U was constructed relative to them. Still, this means that even in models for type-free λ-calculus (which can be regarded as ordinary function spaces), we are *not* losing sight of the standard idea of function. To have $(U \to U)$ as a retract of U, the functions have to "bend" a little, but we have kept them "straight" as far as the given A, B, and C go.

We have just shown how a type-free domain U can incorporate given domains as well as the "arbitrary" functions on them. In Engeler (1979) it is shown that λ-calculus models can also incorporate any *algebra*; specifically it is shown that any algebra can be made isomorphic to a subset of the model where the operation is functional application itself. We shall give a proof here using the constructs we have mentioned.

Let A be set. An "algebra" can be regarded as any binary operation $\bullet : A \times A \to A$. A *partial algebra* can be taken to be any *continuous* $\bullet : A_* \times A_* \to A_*$. It is easy to argue that any algebra on A determines a partial algebra on A_*.

Now let the λ-calculus model U be taken so that $U = A_* \times (U \to U)$. We regard elements $x \in U$ as pairs (x_0, x_1). The application operation $f(x)$ on U can be defined as $f_1(x)$ because $f_1 \in (U \to U)$. Now recall that λ-calculus models satisfy the fixed-point theorem; so we can define a map $\rho : A_* \to U$ by the functional equation:

$$\rho(a) = (a, \lambda x{:}U.\ \rho(a \bullet x_0)),$$

where the λ-operator gives an element in $(U \to U)$. This map is continuous and one-one into. Now calculate in U:

$$\begin{aligned}\rho(a)(\rho(b)) &= \rho(a \bullet \rho(b)_0)\\ &= \rho(a \bullet b)\end{aligned}$$

So the image of ρ is closed under application, and the resulting applicative subalgebra is *isomorphic* to the given algebra $\langle A_*, \bullet\rangle$.

This result would seem to have a potentially useful implication for *non-extensional* models of combinatory algebra: any such can be embedded in an application-preserving way into an extensional model. This works even if we regard application as a *partial* operation. *Warning*: we do *not* obtain a combinator-preserving embedding, however. That is, if the algebra $\langle A_*, \bullet\rangle$ has elements S and K, satisfying the usual equations in A_*, we *cannot* conclude that the embedding $\rho : A_* \to U$ will map the S and K of A_* to the "true" S and K of the λ-calculus model U. The "functions" in A_* operate only on A_*, which is quite a limited part of U; clearly ρ does not give elements $\rho(a) \in U$ very broad rôles. But at least we can say that anything that even looks a little like application can be assumed to be application in a suitable domain.

6. SUMMARY AND CONCLUSIONS

The constructions reviewed and outlined here have been rather lengthy, so it would seem best to summarize the principal conclusions we have reached.

1. *A theory in typed λ-calculus is just the same as a cartesian closed category.*

As was stated, this has been known for well over ten years from the work of Lambek. It should be stressed, however, that category theory achieves a greater generality than the usual logical presentations, because in category theory the type constructions are *axiomatized.* Thus, the types form an "algebra" under the operations $T \times U$ and $(T \to U)$. We need not assume that we always have a "free" algebra of types built out of "ground" types.

2. *In a c.c.c. a reflexive domain provides an interpretation of the "type-free" theory.*

We can call U "reflexive" if $(U \to U)$ is a retract. The last statement is of course obvious. What makes it interesting is:

3. *Every type-free theory is the theory of a reflexive domain in a c.c.c.*

The proof of this result was by the author's method of *retracts.* The use of idempotents in a category as forming a category is well known, but the author believes that he was the first to note that in a c.c.c. we really have a calculus of retracts - especially when there are reflexive domains available.

Then some remarks were made about theories and models and the significance of adding indeterminants (also well known). What might not have been clear from other works was the type-free theory in terms of application, S, and K. Up to that point the theories had been equational; and, though the first-order version (with extensionality) was pleasant, it was not of great philosophical interest since it does not relate the idea of λ-calculus to any broad notion of functions. This desire was taken care of by:

4. *Every c.c.c. can be fully and faithfully embedded in an intuitionistic theory of types with the full (impredicative) power-set construct and function spaces (higher-order intuitionistic logic).*

The domains of the c.c.c. become types in the theory. The word "fully" means that the definable maps between the types all come from maps in the category; "faithfully" means that in the higher-order theory no new equations between these maps are introduced over what we already had in the category. In other words, this is a conservative extension result. It has been known for quite a time in category theory, and the functor category we employed in the construction is one of the very first examples of a *topos*; there must be considerable use possible of more interesting examples of topoi.

However, there was already enough philosophical interest in this easy construction. Namely, it was seen that equational λ-calculus is perfectly consistent with higher-order logic where - provided we only employ intuitionistic logic - we can speak of function spaces in the normal way in type theory. Some people can, if they like, stick to λ-terms and equations; but others can use whatever logical means they like for discussing functions. However, if the logician *proves* in his higher-order theory that a certain property picks out a function $f : A \to B$, then, if A and B are from our given category, this definable f must be given by a standard λ-term. So the logic in that sense gives nothing new, but at least we know that the sense for λ-calculus is exactly that it can always be taken to be talking about functions and *full* function spaces in a higher-order theory.

Turning things around the other way, it is interesting to see from what we know about "type-free" theories that intuitionistic logic allows for reflexive domains - and even lots of them. It would be even more interesting if it were possible to strengthen higher-order logic (say, by adding some new primitives), so that we could express *in the logic* the axiom that *every* c.c.c. (a structure satisfying a simple first-order statement) had a full and faithful representation in a category of subsets (better: quotients of subsets) of a reflexive domain. I conjecture that this is possible and that the theory can be taken conservative over any given c.c.c.

Though we did not prove such a sweeping result, we did sketch a proof of:

5. *Every given c.c.c. can be realized fully and faithfully as a category of subtypes of a reflexive type in a higher-order theory.*

We did not work out the cartesian closed details of this assertion but contented ourselves with showing how to accomodate a finite number of types. By the way, it should be remarked that 4 and 5 hold for an arbitrary (small) category, so in particular we have the (known) result that every category can be conservatively extended to a c.c.c. The method of proof for 5 was *via* the author's original construction of λ-calculus models, which can be conveniently carried out in the category of "injective" T_0-spaces. There are many variations on this construction, and it might be interesting to see how different the different categories with reflexive domains really are from the point of view of higher-order logic.

Another remark: the construction has many connections with *Curry's Theory of Functionality* (i.e. the problem of finding other c.c.c.'s inside a λ-calculus model). But as I have indicated several times it is really better to work with equivalence relations (on subtypes of a reflexive domain U) because in typing the functions we have to make them hereditarily extensional in order to be able to have a category. Thus a reflexive domain has (at least) two interesting c.c.c.'s associated with it: the category of retracts and the category of equivalence relations. The second, by the way, contains the first as a sub-c.c.c.

Finally we recalled a result of Engeler which was very appropriate to the present discussion:

6. *Any (partial) algebra can be isomorphically represented as an applicative subalgebra of a reflexive domain.*

I think that this means in particular that non-extensional theories of functions can be subsumed under the extensional theory: the non-extensional function algebras are just subalgebras of normal function algebras. There is certainly a conservative extension result here, but whether it helps to prove any new theorems is another question.

What has to be investigated next, I think, is the problem of how strong the higher-order theories can be made and still have them as conservative extensions of given categories. Much is known in topos theory about constructions of categories of *sheaves* (these are subcategories of the functor category), but much remains to be explained to the logician. Thus, there are several interesting categories made up out of continuous functions or out of computable functions (when we look at them from the outside), but what we would like to know is what logical sentences (internal properties) are satisfied for the various functor or sheaf categories. The so-called Church's Thesis (all number-theoretic functions are recursive) or Brouwer's Theorem (all real functions are continuous) are cases in point.

and they are satisfied in certain topoi. It would be an important next step for λ-calculus to relate these model constructions to interpretations of λ-calculus. The author hopes that the present paper will encourage others to look further.

REFERENCES

Barendregt, H., (1980). "The Lambda Calculus: its Syntax and Semantics", North-Holland, Amsterdam, to appear.

Engeler, E., (1979). Algebras and combinators, *preprint, ETH Zürich.*

Gierz, G., Hofmann, K.H., Keimel, K., Lawson, J.D., Mislove, M., and Scott, D.S., (1980). "A Compendium of Continuous Lattices", Springer-Verlag, Heidelberg, to appear.

Goldblatt, R., (1979). "Topoi: The Categorial Analysis of Logic", North-Holland, Amsterdam.

Hindley, R., and Longo, G., (1980). Lambda calculus models and extensionality, *Math. Logik u. Grundl. d. Math,* to appear.

Johnstone, P., (1977). "Topos Theory", London Math. Soc. Monographs No. 10, Academic Press, London.

Joyal, A., and Reyes, G.E., (1980). Forcing and generic models in categorical logic, *J. Pure and Applied Algebra,* to appear.

Lawvere, F.W., (1975). Continuously variable sets: algebraic geometry = geometric logic, *In* "Logic Colloquium '73", (Eds. H.E. Rose and J.C. Shepherdson), North-Holland, Amsterdam, pp. 135-156.

MacLane, S. (1971). "Categories for the Working Mathematician", Springer-Verlag, Heidelberg.

Plotkin, G. (1972). A set-theoretical definition of application, *Memo,* MIP-R-95, *School of Artificial Intelligence, Univ. of Edinburgh.*

Scott, D.S., (1976). Data types as lattices, *Siam J. Computing,* Vol. 5, pp. 522-587.

Scott, D.S., (1979). Identity and existence in intuitionistic logic, *In* "Applications of Sheaves", Springer-Verlag, Lecture Notes in Math. Vol. 753, pp. 660-696.

Scott, D.S., (1980). Lambda calculus: Some models, some philosophy, *In* "The Kleene Symposium", (Eds. J. Barwise, H.J. Keisler, and K. Kunen), North-Holland, Amsterdam, pp. 381-421.

Szabo, M.E., (1978). "Algebra of proofs", North-Holland, Amsterdam.

TYPED TERMS

AN EARLY PROOF OF NORMALIZATION BY A.M. TURING

R.O. Gandy

Mathematical Institute, 24-29 St. Giles, Oxford OX1 3LB, UK

Dedicated to H.B. Curry on the occasion of his 80th birthday

In the extract printed below, Turing shows that every formula of Church's simple type theory has a normal form. The extract is the first page of an unpublished (and incomplete) typescript entitled 'Some theorems about Church's system'. (Turing left his manuscripts to me; they are deposited in the library of King's College, Cambridge). An account of this system was published by Church in 'A formulation of the simple theory of types' (J. Symbolic Logic 5 (1940), pp. 56-68). Church had previously described the system in lectures given at Princeton (1937-38) which Turing attended; he was a graduate student at Princeton 1936-1938. He is mentioned as having contributed to results about the system in footnote 12 of Church's paper. In an undated letter to M.H.A. Newman (which must have been written early in 1942) Turing outlines the contents of his proposed paper (including the normal form theorem); he refers to it as 'forthcoming' in his paper 'The use of dots as brackets in Church's system' (J. Symbolic Logic 7 (1942), pp. 146-156, received 17 June 1942). For some further details about Turing's work on type theory, see my paper 'The simple theory of types' in Logic Colloqium '76, Ed. R.O. Gandy & J.M.E. Hyland, North Holland Pub. Co. Amsterdan 1977, pp. 173-181. Thus Turing's proof antedates by many years any published proof of the theorem.

Turing's proof depends on the rather obvious remark that if one reduces the rightmost (or an innermost) redex $(\lambda x_\beta A_\alpha)B_\beta$ whose head $\lambda x_\beta A_\alpha$ is of highest type in a formula F, then the resulting formula has fewer redexes with head of type $(\alpha\beta)$. The theorem follows by use of simple induction with a π^0_1 predicate, or by transfinite induction up to ω^2 with a primitive recursive predicate. (Turing's use of an ordering of formulae with order-type ω^ω is not necessary). A very meticulous account of this method of proof is given by P.B. Andrews in 'Resolution in type theory' (J. Symbolic Logic 36 (1971), pp. 414-432). Andrews writes in a footnote: 'This proposition is part of the folklore of type-theoretic λ-conversion. The author first heard the idea of the proof given here from Dr. James R. Guard'. The same method of proof, applied to contractions of proofs in systems of natural deduction, is used by Pravitz in his *Natural Deduction: a proof-theoretical study* (Stockholm 1965).

The earliest published proof known to me is in Curry & Feys' book *Combinatory Logic* (North Holland Pub. Co. Amsterdam 1958; 2nd printing 1968). The normal form theorem is included in their Theorem 9 on page 340. The proof depends on the 'elimination theorem' (Theorem 5, p. 326), which may be viewed as sort of cut-eliminiation theorem for the theory of functionality. In a very loose sense, one may say that Gentzen's Hauptsatz was the first normal form theorem.

I should like to thank J.P. Seldin for his help in providing references.

THE EXTRACT

Proof that every typed formula has a normal form

We will well-order the formulae of Church's system as follows. We say that the formula has an unreduced part of order n if it has a part of form $(\lambda x_\alpha A_\beta)B_\alpha$ where $\beta\alpha$ is of length n. If we wish to decide which of two formulae precedes we find out what is the highest order of any unreduced part in each. The formula which has an unreduced part of higher order than any part of the other comes later. Suppose however that the maximum orders are the same, n say, then the one which has the more unreduced parts of order n comes the later. But these numbers may be also equal, and in this case we compare the number of unreduced parts of order n-1, and if we fail with these we go to those of order n-2. If eventually there is a difference the formula with the greater number of unreduced parts comes later, if however the numbers remain the same to the end, i.e. as far as those of order 1 then the longer formula comes later. It is not difficult to see that this is a well ordering of formulae, of type ω^ω.

Now when we perform a reduction on a formula, in which we reduce one of the unreduced parts of highest order, we necessarily decrease the number of unreduced parts of the highest order, for we destroy one and we do not create any more: this at any will be the case if we choose the unreduced part of highest order whose λ lies farthest to the right. We therefore reduce the formula to one which is earlier in the sequence, and as the sequence is well-ordered the sequence of reductions must come to an end.

This has been copied verbatim: 'rate' should be inserted after 'any' in the last line but four.

PROOFS OF STRONG NORMALIZATION

R.O. Gandy

Mathematical Institute, 24-29 St. Giles, Oxford OX1 3LB, UK

Dedicated to H.B. Curry on the occasion of his 80th birthday

In this paper I present a rather transparent method for proving that all reduction sequences terminate[1]. The method is explained in detail for the typed lambda calculus and then it is shown how it can be extended to other calculi, including the calculus of primitive recursive functionals of finite type (Godel 1958) and a term calculus, having direct products and sums as types, which corresponds to the calculus of proofs by natural deduction for the intuitionistic propositional calculus as presented in Prawitz (1971). (Details of these systems and of normalization proofs for them will be found in the invaluable and encyclopoedic book edited by Troelstra (1973)). Besides its essential simplicity the method shows one how, given a term of the calculus, one can write down a numerical term whose value is an upper bound for the length of reduction sequences starting with the given term. However, the method does not prove weak normalization; it assumes that numerical terms have numerical values. Nor, as it stands does it cope with permutative reductions or with proofs incorporating a rule of induction.

1 THE MONOTONICITY OF λ-I TERMS

The method is based on the following observation: if one partially orders the ground type(s) and carries this order pointwise up through the types, then, in the hierarchy of hereditarily strictly monotonic functions every term of the typed λ-I calculus denotes a strictly monotonic function.

1.1 For simplicity we consider only a single ground type, with type symbol o. Let T_o be a non-empty set, partially ordered by a relation $<_o$. Let $T = \{T_\alpha : \alpha$ a type symbol$\}$ be a type structure: i.e. (a) for each type symbol $(\alpha\to\beta)$, $T_{(\alpha\to\beta)}$ is a non-empty collection of functions from T_α into T_β, and (b) T is closed under application and definition by λ-abstraction. We introduce the collection

$$M = \{M_\alpha : \alpha \text{ a type symbol}\}$$

of hereditarily monotonic members of T, and the relation $<_\alpha$ on M_α as follows.

(1) $(M_o,<_o) = (T_o,<_o)$

(2) $M_{(\alpha\to\beta)} = \{f \in T_{(\alpha\to\beta)} : \forall a,\ a' \in M_\alpha .\ fa \in M_\beta \wedge (a <_\alpha a' \Rightarrow fa <_\beta fa')\}$.

(3) For $f,\ g \in M_{(\alpha\to\beta)}$

$$f <_{(\alpha\to\beta)} g \Leftrightarrow \forall a \in M_\alpha \ .\ fa <_\beta ga \quad .$$

The M is closed under application and $M_\alpha \subset T_\alpha$; $<_\alpha$ is transitive and, if all the M_α are not empty, a partial ordering.

1.2 Under an assignment of a value x_α^A in M_α to each free variable x_α in a term σ (of the typed λ-calculus) of type β the term σ will denote an element of T_β which may or may not belong to M_β . For example, $\lambda x_\alpha\, y_\beta$ and $\lambda f_{((\alpha\to\beta)\to\gamma)} f(\lambda x_\alpha\, y_\beta)$ will not have denotations in M. (Here, and in future, we suppress the type subscript from all occurrences of a bound variable other than the binding occurrence.)

We recall that the class of typed λ-I terms is defined by: (i) variables are λ-I terms; (ii) if σ and τ are λ-I terms of types $(\alpha\to\beta)$ and α, then $(\sigma\tau)$ is a λ-I term of type β; (iii) if τ is a λ-I term of type β which contains a free occurrence of x_α, then $(\lambda x_\alpha \tau)$ is a λ-I term of type $(\alpha\to\beta)$. As is customary, we shall often omit the outermost parentheses. Now we can prove the observation made at the beginning of this section.

1.3 THEOREM (i) Every typed λ-I term ρ has a denotation in M for every assignment of values in M to its free variables. (ii) If the variable x_α actually occurs free in a term ρ of type β and if the assignments A,B agree on all the free variables of ρ except x_α, then

$$x_\alpha^A <_\alpha x_\alpha^B \Rightarrow \rho^A <_\beta \rho^B \quad .$$

Proof. Both (i) and (ii) are trivially true if ρ is a variable. Suppose that ρ is $\sigma\tau$ and that (i) and (ii) hold for σ and τ. Then (i) holds for ρ. Suppose that $x_\alpha^A <_\alpha x_\alpha^B$. Since x_α must occur in σ, in τ, or in both there are three cases. Case A: $\sigma^A <_{(\gamma\to\beta)} \sigma^B$ and $\tau^A = \tau^B$. Then, by the definition of $<_{(\gamma\to\beta)}$, $\rho^A <_\beta \rho^B$. Case B: $\sigma^A = \sigma^B$ and $\tau^A <_\gamma \tau^B$. Then, since by hypothesis $\sigma \in M_{(\gamma\to\beta)}$, $\rho^A <_\beta \rho^B$. Case C: $\sigma^A <_{(\gamma\to\beta)} \sigma^B$ and $\tau^A <_\gamma \tau^B$.
Then

$$\sigma^A \tau^A <_\beta \sigma^B \tau^A <_\beta \sigma^B \tau^B$$

and $\rho^A <_\beta \rho^B$ follows because $<_\beta$ is transitive.

Suppose now that ρ is $\lambda y_\gamma\, \tau$ where τ is of type δ (so that $\beta = (\gamma\to\delta)$). By induction hypothesis, (i) and (ii) (with y_γ as the indicated variable) hold for τ, and so $\rho^A \in M_{(\gamma\to\delta)}$; but (ii) also holds for τ with x_α as the indicated variable, irrespective of the value assigned to y_γ. Hence

$$x^A{}_\alpha <_\gamma x^B{}_\alpha \Rightarrow (\lambda y_\gamma \tau)^A <_{(\gamma\to\delta)} (\lambda y_\gamma \tau)^B \quad ,$$

and so (ii) holds also for ρ.

1.4 COROLLARY Let σ be a particular occurrence of a subterm (of type β) of the λ-I term τ (of type α) and let τ_1 be the result of substituting the λ-I term σ_1 (of type β) for σ in τ; and suppose that τ_1 is also a λ-I term. Let $z^1_{\gamma_1},\ldots,z^r_{\gamma_r}$ be a list of the variables which occur free in σ or σ_1 but which are bound in τ; (we assume they are distinct from the bound variables of σ and σ_1). Let A be an assignment in M for the free variables of σ and σ_1, and suppose that for every assignment B which extends A and which assigns values in M to $z^1_{\gamma_1},\ldots,z^r_{\gamma_r}$:

(1) $\sigma_1^B <_\beta \sigma^B$.

Then $\tau_1^A <_\alpha \tau^A$.

Proof. We proceed by induction on r. For r = 0, the condition (1) is simply $\sigma_1^A <_\beta \sigma^A$; then $\tau_1^A <_\alpha \tau^A$ follows directly from the theorem. Assume now (IH) that the corollary holds when there are fewer than r free variables of σ or σ_1 which occur bound in τ. Assume the premises (and notations) of the corollary and let $z^1_{\gamma_1}$ be the variable whose binding occurrence lies outside the binding occurrences of $z^2_{\gamma_2},\ldots,z^r_{\gamma_r}$. Thus σ occurs in a part π of τ of type $(\gamma_1\to\delta)$ and of the form $\lambda z^1_{\gamma_1}\,\rho$ and τ_1 has a corresponding part $\pi_1 = \lambda z^1_{\gamma_1}\rho_1$. For any assignment C which extends A by assigning a value in M to $z^1_{\gamma_1}$ we have, by (IH) (applied with ρ replacing τ and C replacing A):

$$\rho_1^C <_\delta \rho^C$$

But then, since π and π_1 are both λ-I terms, the theorem tells us that

$$\pi_1^A <_{(\gamma_1\to\delta)} \pi^A ,$$

and further that $\tau_1^A <_\alpha \tau^A$. This completes the inductive step and the proof of the corollary. The slight fussiness arises from the fact that we wish to apply to corollary in cases where some of the 'z's' do not occur in σ_1 .

2 INTRODUCTION OF ADDITION

2.1 We now suppose that $(T_0, <_0)$ is the set of natural numbers with the usual ordering and that addition belongs to $T_{(0\to(0\to0))}$ and hence also to M. We introduce new symbols 0_o, (for zero) S^0 (for successor) and $+_0$ (for addition), and call the resulting system the 'λ^+ calculus'. A λ-I^+ term is a λ-I term which may contain these new symbols. Since S and + are monotonic, 1.3 and 1.4 also hold for λ-I^+ terms.

We carry S and + up through the types:

2.2 DEFINITION

$$S^{(\alpha\to\beta)} = \lambda f_{(\alpha\to\beta)}\, \lambda x_\alpha \,.\; S^\beta(fx),$$

$$+_{(\alpha\to\beta)} = \lambda f_{(\alpha\to\beta)}\, \lambda g_{(\alpha\to\beta)}\, \lambda x_\alpha \,.\; fx +_\beta gx \,,$$

where for ease of reading we have placed '+' between its arguments. By 1.3, S^α and $+_\alpha$ belong to M; here and elsewhere we do not make any notational distinction between a closed term and its denotation. Our estimate for the height of a reduction tree rests on the following lemma, which is an immediate consequence of the definitions.

2.3 LEMMA

$$x_\alpha^A,\ y_\alpha^A \in M \Rightarrow x_\alpha^A <_\alpha (S^\alpha(x_\alpha +_\alpha y_\alpha))^A \quad .$$

The presence of 0_o and $+_0$ ensures that, for every type α, there are closed terms with denotations in M_α .

2.4 DEFINITION

(i) $L_0 = 0_o$. (ii) $L_{(0\to0)} = \lambda x_0 . x$

(iii) $L_{((\alpha\to\beta)\to0)} = \lambda f_{(\alpha\to\beta)} . L_{(\beta\to0)}\, (fL_\alpha)$.

(iv) $L_{(\alpha\to(\beta\to\gamma))} = \lambda x_\alpha\, \lambda y_\beta . L_{(\alpha\to\gamma)}\, x +_\gamma L_{(\beta\to\gamma)} y$.

These are all λ-I^+ terms and so have denotations in M. These definitions may be made more digestible by remarking that if

$$\alpha = (\alpha_1 \to (\alpha_2 \to \ldots (\alpha_n \to 0)\ldots))$$

then, omitting brackets with association to the left, we have:

(v) $L_{(\alpha\to0)}\, x_\alpha = x_\alpha\, L_{\alpha_1}\, L_{\alpha_2} \ldots L_{\alpha_n}$

(vi) $L_\alpha x_{\alpha_1} x_{\alpha_2} \ldots x_{\alpha_n} = \sum_{i=1}^{n} p_i (L_{(\alpha_i \to 0)} x_{\alpha_i})$,

where $p_1 = 1$, and for $i > 1$, $p_i = 2^{i-2}$.

2.5 DEFINITION The <u>norm</u>, written $|\tau|^A$, of any term τ of type α with respect to the assignment A of values in M to its free variables, is $L_{(\alpha\to 0)}\tau^A$.

2.6 Assuming weak normalisation for the typed λ-calculus, if τ is a closed term then $L_{(\alpha\to 0)}\tau$ will reduce to a numerical term built up from 0, S^0 and $+_0$, and so $|\tau|$ can be easily computed. Note that for closed terms σ, τ of type α

(1) $\sigma <_\alpha \tau \Rightarrow |\sigma| < |\tau|$.

The converse implication does not hold; for example, if α is $(0\to 0)$ then $|\sigma|$ is the value of the function denoted by σ at 0, so that e.g., $|\lambda x_0 . x + x| < |\lambda x_0 . S^0 x|$.

2.7 *A note on proofs.* For clarity we have used a type structure to present our definitions and proofs. It would however be possible to carry out the arguments in pure numerical terms; variables of type α would then be interpreted as ranging over terms of type α. If one introduced symbols for all the relevant functions (certainly primitive recursive), one could conduct the proofs using just π_1^0-induction, and so, I presume, in primitive recursive arithmetic.

3 STRONG NORMALISATION

3.1 We introduce new abstraction operators λ^* as abbreviations by

(1) $\lambda^* x_\alpha \tau$ stands for $\lambda x_\alpha . S^\beta(\tau +_\beta L_{(\alpha\to\beta)^+} x)$,

where τ is a term of type β. If τ is a λ-I^+ term (which need not contain any occurrences of x_α) then so in $\lambda^* x_\alpha \tau$. Further, if τ and σ are λ-I^+ terms, and A is any assignment of values in M to the free variables of $(\lambda x_\alpha \tau(x))\sigma$ (which are assumed to be distinct from the bound variables of τ) then, by 2.3,

(2) $\quad (\tau(\sigma))^A <_\beta ((\lambda^* x_\alpha \tau)\sigma)^A$

(where, as usual, $\tau(\sigma)$ is the result of substituting σ at all the free occurrences of x_α in $\tau(x_\alpha)$). This inequality is the nub of our proof of strong normalisation; it only remains to fill in some details.

3.2 Let X be any typed λ-calculus and let K be the set of its primitive type symbols. We define a mapping of terms of X into terms of the λ^+ calculus of §2 as follows:

Type symbols: (i) α^* is o if $\alpha \in K$;

(ii) $(\alpha \to \beta)^*$ is $(\alpha^* \to \beta^*)$.

Terms: (iii) x^*_α is x_{α^*} for any variable x_α of X, and we assume that distinct letters are used for variables in distinct types, so that $\alpha \neq \beta \Rightarrow (x_\alpha)^* \neq (y_\beta)^*$ for any variables x_α, y_β;

(iv) $(\tau\sigma)^*$ is $(\tau^*\sigma^*)$;

(v) $(\lambda x_\alpha \tau)^*$ is $(\lambda^* x_{\alpha^*} \tau^*)$.

Plainly for every term τ of X, τ^* is a λ-I^+ term. Also, by induction on the construction of any term $\tau(= \tau(x_\beta)$, say), one sees that $\tau^*(\sigma^*)$ is $(\tau(\sigma))^*$; we assume all bound variables are chosen so as to avoid collisions.

Now we can prove strong normalisation for X in the following form.

3.3 THEOREM Let ρ be any term of type α of the typed λ-calculus X, and let $\rho_0(=\rho), \rho_1, \ldots, \rho_n$ be any sequence of immediate reductions. Let A be any assignment of values in M to the free variables of ρ; then $n \leq |\rho^*|^A$. In particular, an upper bound for n is obtained by substituting constants L_β of appropriate type for the free variables of ρ^* and computing (as in 2.6) the norm of the resulting term.

Proof. Let ρ_{i+1} be obtained from ρ_i by replacing a part

$((\lambda x_\gamma.\tau(x))\pi)$ (= σ, of type β, say) by $\tau(\pi)$ (= σ_1). Let B be any extension of A to the free variables of σ. Then, by 3.1 (2) $(\sigma_1{*})^B <_\beta (\sigma{*})^B$. But the ρ_i^* are λ-I^+ terms; hence, by 1.4,

$$(\rho^*_{i+1})^A <_\alpha (\rho^*_i)^A \quad .$$

The theorem now follows by 2.6 (1).

4 EXTENSION TO PRIMITIVE RECURSIVE FUNCTIONALS

4.1 The system T of Gödel (1958) can be obtained by adding recursors R^α, for each type α, to the typed λ^+ calculus of §2; (for details see Troelstra (1973)). For simplicity we keep the now redundant constant $+_0$. R^α is of type $(0{\to}((0{\to}(\alpha{\to}\alpha)) \to (\alpha{\to}\alpha)))$ and satisfies

$$\text{(1)} \qquad R^\alpha\, 0\, \tau\, \sigma = \sigma \quad ,$$
$$R^\alpha(S^0\rho)\, \tau\, \sigma = \tau\, \rho\, (R^\alpha\rho\tau\sigma) \quad ,$$

for any terms ρ, σ, τ of types 0, α, $(0{\to}(\alpha{\to}\alpha))$ respectively. Reductions in this calculus consist of λ-reductions and the substitution of terms on the RHS of (1) for the correponding terms on the LHS.

4.2 We take over the definition of the * translation from §3 , setting of course, $\alpha^* = \alpha$, $0^* = 0$, $(S^0)^* = S^0$, $(+_0)^* = +_0$. If we can find, for each type α, a term $(R^\alpha)^*$ which satisfies:

(1) $\qquad (R^\alpha)^* \in M$;

(2a) $\qquad G <_\alpha (R^\alpha)^*\, 0\, H\, G$

(2b) $\qquad Hm((R^\alpha)^*\, m\, HG) <_\alpha (R^\alpha)^*\, (Sm) H\, G$

for any $m \in T_0$, $G \in M_\alpha$, $H \in M_{(0\to(\alpha\to\alpha))}$; then, <u>mutatis mutandis</u>, we can take over the proof of theorem 3.3.

In fact, if we set $(R^\alpha)^*$ =

$$\lambda y_0\, \lambda h_{(0\to(\alpha\to\alpha))}\, \lambda g_\alpha . R^\alpha y(\lambda x_0\, \lambda u_\alpha . S^\alpha(hxu +_\alpha u))(S^\alpha(g +_\alpha hOL_\alpha)),$$

then (1), (2a), (2b) are indeed satisfied. For we have

(3) $\quad (R^{\alpha})^{*}\ 0\,H\,G = S^{\alpha}(G +_{\alpha} H\,0\,L_{\alpha})$,

(4) $\quad (R^{\alpha})^{*}(Sm)HG = S^{\alpha}(Hm((R^{\alpha})^{*}m\,H\,G) +_{\alpha} (R^{\alpha})^{*}m\,H\,G)$.

Now (3) shows that $(R^{\alpha})^{*}\ 0\,H\,G$ in monotonic in H and G; (4) shows by induction that this is also true for $(R^{\alpha})^{*}m\,G\,H$ for any m. Also by (4) and an obvious variant of lemma 2.3. we see that $(R^{\alpha})^{*}m\,H\,G <_{\alpha} (R^{\alpha})^{*}(Sm)HG$, so that $(R^{\alpha})^{*}$ is monotonic in its first argument. Finally (3) and (4) and lemma 2.3 show that (2a), (2b) are satisfied. This completes the proof of strong normalisation for Gödel's T; we have of course assumed weak normalisation by supposing that a term of type 0 will denote a natural number.

5 EXTENSION TO DIRECT PRODUCT AND DIRECT SUM

5.1 We extend any typed λ-calculus to a typed 'λ-ν-π-calculus' by adding new type symbols, constants and terms as follows.

(A) If α_1 and α_2 are type symbols so are $\alpha_1 \times \alpha_2$ and $\alpha_1 + \alpha_2$.

(B) $P^{\alpha_1 \times \alpha_2}$ (for 'pair') is a constant of type $(\alpha_1 \to (\alpha_2 \to (\alpha_1 \times \alpha_2)))$.

(C) For k = 1,2, ${}^{k}\Pi^{\alpha_1 \times \alpha_2}$ (for 'projection') is a constant of type $((\alpha_1 \times \alpha_2) \to \alpha_k)$.

(D) For k = 1,2, ${}^{k}\iota^{\alpha_1 + \alpha_2}$ (for 'injection') is a constant of type $(\alpha_k \to (\alpha_1 + \alpha_2))$.

From now on, k will always range over {1,2}.

(E) If $\sigma_1 (= \sigma_1(x_{\alpha_1}))$ and $\sigma_2 (= \sigma_2(y_{\alpha_2}))$ are terms of type β, then

$$\nu x_{\alpha_1}\, y_{\alpha_2} [\sigma_1(x_{\alpha_1}),\ \sigma_2(y_{\alpha_2})]$$

is a term of type $((\alpha_1 + \alpha_2) \to \beta)$; in it the free occurrences of x_{α_1} in σ_1 and of y_{α_2} in σ_2 are bound by the operator $\nu x_{\alpha_1}\, y_{\alpha_2}$.

5.2 These constants satisfy the following identities:

(1) $\quad {}^{k}\Pi^{\alpha_1\times\alpha_2}\,(P^{\alpha_1\times\alpha_2}\,\sigma_1\,\sigma_2) = \sigma_k$.

(2) $\quad (\nu x_{\alpha_1}\,y_{\alpha_2}[\sigma_1(x_{\alpha_1}),\,\sigma_2(y_{\alpha_2})])({}^{k}\iota^{\alpha_1+\alpha_2}\tau_k) = \sigma_k(\tau_k)$,

where τ_k is a term of type α_k, and the free variables of τ_k do not occur bound in σ_k .

An immediate π (resp. ν) reduction consists of substituting for a part of a term having the form of the LHS of (1) (resp. (2)) the corresponding RHS.

5.3 All the above corresponds exact to the usual category-theoretic treatment of $\times$ and $+$. In particular,

$$f_1 \times f_2 = \lambda z_\beta\; P^{\alpha_1\times\alpha_2}\,(f_1 z)(f_2 z) \qquad (f_k \text{ of type } (\beta\to\alpha_k)) ,$$

$$g_1 + g_2 = \nu x_{\alpha_1}\; y_{\alpha_2}\;[g_1 x,\; g_2 y] \qquad (g_k \text{ of type } (\alpha_k\to\beta)) ,$$

are the maps from β into in $\alpha_1\times\alpha_2$ and from $\alpha_1+\alpha_2$ into β which are given by the universal properties of $\times$ and $+$.

In proof-theoretic terms $\times$ and $+$ correspond to $\wedge$ and $\vee$, and P and ${}^{k}\Pi$ correspond to the introduction and elimination rules for $\wedge$. ${}^{k}\iota$ corresponds to 'strong' $\vee$-introduction; (i.e. in an inference from α to $\alpha\vee\alpha$ the rule indicates whether the proof of α is to be attached to the left or right occurrence of α in $\alpha\vee\alpha$). Finally, $(\nu x_{\alpha_1}\;y_{\alpha_2}[\sigma_1,\;\sigma_2])\tau$ is a (term for a) derivation which ends with the $\vee$-elimination rule, where τ is the derivation of the major premise, σ_1 and σ_2 are the derivations of the minor premises, and x_{α_1}, y_{α_2} are those assumptions of α_1 in σ_1 and α_2 in σ_2 which are cancelled by the application of the rule.[3]

5.4 In order to extend the proof of strong normalisation to the new calculus, we first extend the definition of §1.

For direct product the extension of the notion of monotonicity is unproblematic. We set

(1) $\quad M_{(\alpha_1\times\alpha_2)} = M_{\alpha_1}\times M_{\alpha_2} = \{(a,b)\colon a\in M_{\alpha_1},\; b\in M_{\alpha_2}\}$,

(2) $(a,b) <_{(\alpha_1\times\alpha_2)} (a',b') \Leftrightarrow a <_{\alpha_1} a' \wedge b <_{\alpha_2} b'$.

For direct sum the most natural definitions are:

(3) $M_{(\alpha_1+\alpha_2)} = \{(k,a): k = 1,2 \wedge a\epsilon M_{\alpha_k}\}$,

(4) $(k,a) <_{(\alpha_1+\alpha_2)} (k',a') \Leftrightarrow k = k' \wedge a <_{\alpha_k} a'$.

The denotations of the constants P, ${}^k\Pi$, ${}^k\iota$ are the obvious ones; e.g., if $a\epsilon M_{\alpha_k}$ then ${}^k\iota a = (k,a)$; evidently the denotations of ${}^k\Pi$ and ${}^k\iota$ belong to M. For a ν term:

(5) $(\nu x_{\alpha_1} y_{\alpha_2} [\sigma_1, \sigma_2])^A a = (\lambda x_{\alpha_1} \sigma_1)^A a'$ if $a = (1,a')$,

$= (\lambda y_{\alpha_2} \sigma_2)^A a'$ if $a = (2,a')$.

An extension of the notion of λ-I term, namely the property of being ν-I term is obtained by adding to the definition given in 1.2 the following clauses:-

(i') the constants ${}^k\Pi$, ${}^k\iota$ are ν-I terms;

(iv) if σ_1 and σ_2 are ν-I terms of types α_1 and α_2 which have exactly the same free variables, then ${}^k_1P^{\alpha_1\times\alpha_2 2}\sigma_1\sigma_2$ is a ν-I term of type $(\alpha_1\times\alpha_2)$;

(v) if σ_1 and σ_2 are ν-I terms of type β then $\nu x_{\alpha_1} y_{\alpha_2} [\sigma_1,\sigma_2]$ (= ρ, say) is a ν-I terms of type $((\alpha_1+\alpha_2)\to\beta)$ provided that:

(a) x_{α_1} occurs free in σ_1 and y_{α_2} occurs free in σ_2 ;

(b) any variable which occurs free in ρ occurs free in both σ_1 and σ_2; note that this precludes, e.g. x_{α_1} from occurring free in σ_2 .

It is now straightforward to extend the proof of 1.3 to obtain:

5.5 THEOREM Under any assignment A of values in M to the free variable of the ν-I term ρ, $\rho^A \epsilon M$ and the value ρ^A is monotonic in the assignment A.

Observe that if ρ is as in (iv) above, then condition (a) ensures that ρ^A belong to $M_{((\alpha_1+\alpha_2)\quad)}$ and condition (b) ensures that ρ^A is monotonic in the assignment made to any of its free variables.

The corollary 1.4 and its proof are now extended simply by replacing 'λ-I' by 'ν-I'.

5.6 We now extend the work of §2. We will denote by ν^+ the calculus got by extending the λ^+ calculus of §2 (with 0 as the basic type and primitive constants 0, S^0, $+_0$) as in 5.1. The extensions of S and + to the new types are got by adding to the definitions in 2.2 the following clauses:

(1) $$S^{\alpha_1\times\alpha_2} = \lambda z_{(\alpha_1\times\alpha_2)}\ P(S^{\alpha_1}({}^1\Pi z)(S^{\alpha_2}({}^2\Pi z)).$$

(2) $$S^{\alpha_1+\alpha_2} = \nu x_{\alpha_1}\ y_{\alpha_2}\ [{}^1\iota(S^{\alpha_1}x),\ {}^2\iota(S^{\alpha_2}y)]\ .$$

(3) $$+_{\alpha_1\times\alpha_2} = \lambda x_{(\alpha_1\times\alpha_2)}\ y_{(\alpha_1\times\alpha_2)}\ P({}^1\Pi x +_{\alpha_1} {}^1\Pi y)({}^2\Pi x +_{\alpha_2} {}^2\Pi y)$$

Note that we often omit type superscripts and subscripts; note also that the RH sides of the above are ν-I^+ terms (and so have denotations in M). For addition in the direct sum there are several alternatives; the reason for the particular choice made is that it preserves 2.3. We assume (see below) that we have already defined closed ν-I^+ terms $L_{(\alpha_1\to\alpha_2)}$ and $L_{(\alpha_2\to\alpha_1)}$. Then we define

$$L^1_{((\alpha_1+\alpha_2)\to\alpha_1)} = \nu x_{\alpha_1}\ y_{\alpha_2}\ [x,\ L_{(\alpha_2\to\alpha_1)}y]\ ,$$

$$L^2_{((\alpha_1+\alpha_2)\to\alpha_2)} = \nu x_{\alpha_1}\ y_{\alpha_2}\ [L_{(\alpha_1\to\alpha_2)}\ x, y]\quad .$$

Now we set

(4) $$+_{\alpha_1+\alpha_2} = \lambda u_{(\alpha_1+\alpha_2)}\ \lambda v_{(\alpha_1+\alpha_2)}.(\nu x_{\alpha_1}\ y_{\alpha_2}\ [\sigma_1,\sigma_2])u\ ,$$

where

$$\sigma_1 \text{ is } {}^1\iota(x +_{\alpha_1} L^1_{((\alpha_1+\alpha_2)\to\alpha_1)}v) ,$$

$$\sigma_2 \text{ is } {}^2\iota(y +_{\alpha_2} L^2_{((\alpha_1+\alpha_2)\to\alpha_2)}v) .$$

By inspection the RHS of (4) is a ν-I^+ term. Two examples may illuminate:-

$$(1,a) +_{\alpha_1+\alpha_2} (2,b) = (1,a +_{\alpha_1} L_{(\alpha_2\to\alpha_1)}b) ,$$

$$(2,b) +_{\alpha_1+\alpha_2} (2,b') = (2,b +_{\alpha_2} b') ,$$

(where $a \in M_{\alpha_1}$, $b,b' \in M_{\alpha_2}$) . This addition is not commutative, but as is readily checked, the inequality 2.3 continues to hold.

We now add clauses to the inductive definition 2.4 of the (monotonic) constants.

(v) $\qquad L_{\alpha_1\times\alpha_2} = P\ L_{\alpha_1}\ L_{\alpha_2}$.

(vi) $\qquad L_{\alpha_1+\alpha_2} = {}^1\iota\ L_{\alpha_1}$, $\qquad$ (or $= {}^2\iota\ L_{\alpha_2}$)

(vii) $\qquad L_{((\alpha_1\times\alpha_2)\to 0)} = \lambda z_{(\alpha_1\times\alpha_2)}.L_{(\alpha_1\to 0)}({}^1\Pi z) +_0 L_{(\alpha_2\to 0)}({}^2\Pi z)$

(viii) $\qquad L_{((\alpha_1+\alpha_2)\to 0)} = \nu x_{\alpha_1}\ y_{\alpha_2}\ [L_{(\alpha_1\to 0)}\ x,\ L_{(\alpha_2\to 0)}\ y]$.

(ix) $\qquad L_{(\beta\to(\alpha_1\times\alpha_2))} = \lambda y_\beta\ P(L_{(\beta\to\alpha_1)}y)(L_{(\beta\to\alpha_2)}\ y)$.

(x) $\qquad L_{(\beta\to(\alpha_1+\alpha_2))} = \lambda y_\beta.{}^1\iota(L_{(\beta\to\alpha_1)}y)$,

$\qquad\qquad$ or $= \lambda y_\beta.{}^2\iota(L_{(\beta\to\alpha_2)}y))$.

In conjunction with 2.4 (ii)-(iii), (vii) and (viii) give a definition of $L_{(\alpha\to 0)}$ for every type α; then 2.4 (iv), (ix) and (x) give a definition of $L_{(\alpha\to\beta)}$ for all α,β. Finally 2.4 (i), (v), (vi) give a definition of L_γ when γ is not of the form $(\alpha\to\beta)$. By 5.5, the denotations of all these constants belong to M. Of course there is a good deal of arbitrariness in the particular definitions we have made; in particular, $L_{((\alpha_1+\alpha_2)\to\alpha_1)}$ may differ from the L^1 used in the definition of $+_{\alpha_1+\alpha_2}$.

5.7 Now we can extend the results of §3. To avoid a galaxy of stars we consider only the case where 0 is the only basic type; so we take $\alpha^* = \alpha$. To the clauses (iii) to (v) of 3.2 we add:

(vi) $({}^k\Pi)^*$ is ${}^k\Pi$ and $({}^k\iota)^*$ is ${}^k\iota$.

(vii) $(P^{\alpha_1\times\alpha_2})^*$ is $\lambda x_{\alpha_1} y_{\alpha_2} P^{\alpha_1\times\alpha_2}(S^{\alpha_1}\sigma_1)(S^{\alpha_2}\sigma_2)$, where

σ_1 is $(x +_{\alpha_1} L_{(\alpha_2\to\alpha_1)}y)$ and σ_2 is $(y +_{\alpha_2} L_{(\alpha_1\to\alpha_2)}x)$.

(viii) If σ_1, σ_2 are of type β then $(\nu x_{\alpha_1} y_{\alpha_2}[\sigma_1(x), \sigma_2(y)])^*$ is $\nu x_{\alpha_1} y_{\alpha_2}\ [S^\beta\ \pi_1(x), S^\beta\ \pi_2(y)]$,

where

$\pi_1(x)$ is $\sigma_1^*(x) +_\beta (\sigma_2^*(L_{(\alpha_1\to\alpha_2)}x) +_\beta L_{(\alpha_1\to\beta)}x)$,

$\pi_2(y)$ is $\sigma_2^*(y) +_\beta (\sigma_1^*(L_{(\alpha_2\to\alpha_1)}y) +_\beta L_{(\alpha_2\to\beta)}y)$:

we assume that the bound variable x_{α_1} is chosen so as not to collide with the free variables σ_2^* , similarly for y_{α_2} .

By inspection (and induction) for any term ρ, $(\rho)^*$ is a ν-I^+ term. Further, the operation * commutes with substitution and, as remarked in 5.6, the inequality 2.3 still holds. Hence 3.1(2) holds; and further:

(1) $\quad ((\sigma_k(\tau))^{*A} <_\beta (((\nu x_{\alpha_1} y_{\alpha_2}[\sigma_1(x), \sigma_2(y)])({}^k\iota\tau))^*)^A$,

(2) $\quad (\sigma_k^*)^A <_{\alpha_k} (({}^k\Pi^{\alpha_1\times\alpha_2}(P^{\alpha_1\times\alpha_2}\sigma_1\ \sigma_2))^*)^A$,

for any assignment A of values in M to all the relevant free variables. Thus in any immediate reduction, the value of the * of the reduct is less than the value of the * of the corresponding redex. So we can conclude:

5.8 THEOREM Theorem 3.3 holds for the typed λ-ν-π calculus. (For the computation of the bound for n see the Postscriptum).

5.9 *Discussion.* The reductions we have considered correspond exactly to the $\supset$, & and $\vee$-reductions given in Prawitz (1971); further his 'immediate simplifications')p. 254) correspond to norm-reducing operations on terms. But in order to arrive at proofs without maximal segments, Prawitz also introduces '$\vee$E reductions' (p. 253). In our notation the trickiest of these is represented by the passage from the LHS to the RHS of the equation:

(1) $$\rho((\nu x_{\alpha_1}\ y_{\alpha_2}[\sigma_1,\sigma_2])\tau) = (\nu x_{\alpha_1}\ y_{\alpha_2}[\rho\sigma_1,\rho\sigma_2])\tau \quad ,$$

where τ is of type $\alpha_1 + \alpha_2$, σ_1 and σ_2 are of type $\beta_1 + \beta_2$ and ρ is a ν-term of type $((\beta_1+\beta_2)\to\gamma)$. Note that both sides of (1) will have the same value. Suppose that ρ is capable of p successive reductions; then the RHS of (1) will be capable of 2p reductions, and so we may expect that the norm of the RHS will be greater than the norm of the LHS. At this point we seem to be paying a price for the conceptual clarity gained by our extensional interpretation: the value of $(\sigma\tau)$ depends only on the values assigned to σ and τ, not on the particular forms of the terms σ and τ. I think I have found a way round this difficulty: one interprets terms in a structure built up by (hereditarily) forming finite sets of elements in the original type structure; in particular, a typical ν-term will have a denotation which is the disjoint union of the sets denoted by σ_1 and σ_2. But the formal complications make the idea rather unattractive.

6. FURTHER EXTENSIONS

6.1 The method extends easily to the intuitionistic predicate calculus. The description of proofs by terms involves the introduction of atomic types indexed by the type I of individuals. But when we pass to the star interpretation all these atomic types are mapped into type 0, and we can take I^* (= κ, say) to consist of a single element: $M_\kappa = \{k\}$. We do not admit κ as a proper type-symbol, but give the rules: if α is a type symbol so are $(\kappa\to\alpha)$ and $(\kappa\times\alpha)$, corresponding respectively to universal and existential quantification. The rules of term formation are extended by: (i) if ρ is a variable x_κ or the constant k_κ and τ is a $[\nu\text{-}I^+]$ term of type $(\kappa\to\alpha)$ then $(\tau\rho)$ is a $[\nu\text{-}I^+]$ term of type α and $(P^{\kappa\times\alpha}\rho)$ is a $[\nu\text{-}I^+]$ term of type $(\alpha\to(\kappa\times\alpha))$; (ii) if τ is a $[\nu\text{-}I^+]$ terms of type α then $(\lambda x_\kappa\ \tau)$ is a $[\nu\text{-}I^+]$ term of type $(\kappa\to\alpha)$. Then $M_{\kappa\to\alpha}$ will consist of <u>all</u> functions from M_κ to M_α, ordered by their unique value. The ordering of $M_{\kappa\times\alpha}$ (= $M_\kappa \times M_\alpha$) is, naturally, given by:

$$(k,\ a) <_{\kappa\times\alpha} (k,\ a') \iff a <_\alpha a' \quad .$$

Then it is readily checked that Theorem 5.5 holds for the extended system. There is no difficulty in extending the definitions of S, + and < to the new types, nor in proving Theorem 5.8 for the extended system. In applying this result to proofs we first observe that a $\forall$ contraction will correspond to a λ-reduction. Instead of introducing further constants, we suppose that the term corresponding to a proof ending with $\exists$-elimination is formed in such a way that its star has the form:

$$(\lambda x_\kappa\ \lambda y_\alpha\ \tau)({}^1\Pi\sigma)({}^2\Pi\sigma)$$

(But note that if, e.g. σ is a variable of type $\kappa\times\alpha$ corresponding to an assumption $\exists x\ A(x)$, there are no proof-theoretic operations corresponding to ${}^k\Pi\sigma$, so that the λ-reduction is not allowed). Then an $\exists$-contraction corresponds to the replacement of $(\lambda x_\kappa\ y_\alpha\ \tau(x,y))({}^1\Pi(p_{\rho\sigma}))({}^2\Pi(p_{\rho\sigma}))$ by $\tau(\rho,\sigma)$. Hence Thereom 5.8 does yield an upper bound for the number of proof reduction.

6.3 The method can also be applied if $(T_0,<)$ is taken to be an initial segment of the ordinals; 2.3 will hold if $+_0$ is the natural sum. With suitable term forming operations this could, for example, be applied to Howard's (1972) calculus for primitive recursive functionals of finite type over the ordinals.

6.3. In conclusion we comment on the difference between our method of proof and those previously published. They use an inductively defined property of terms (proofs) such as 'strong derivability' or 'computability'; the clauses of the definition involve a universal quantifier over certain collections of terms. In our proof this inductive definition is tucked away in the definitions of 'ϵM_α' and '$<_\alpha$'. This has two advantages: firstly we only require an induction over types, not over terms; and secondly our inductively defined properties correspond to natural and easily graspable concepts, so that one does not have continually to recall the details of the definition when following the proof.

POSTSCRIPTUM

Theorem 5.8 tells us that the number denoted by $|\rho^T|$ is an upper bound to the length of any reduction sequence in a typed λ-ν-π-calculus starting with the term ρ, where ρ^T is a closed term of the ν^+-calculus got by substituting constants L_β of appropriate type for the free variables in ρ^*. We wish to show how $|\rho^T|$ can be computed. In the first place, we have

THEOREM I Any term of a typed λ-ν-π-calculus can be reduced to a normal form (i.e. one for which no immediate reduction can be made).

The proof of this is a straightforward extension of the proof of normalization for the typed λ-calculus (see, e.g., Turing's proof appearing in this volume) and we omit it.

One can also prove a Church-Rosser theorem for the typed λ-ν-π-calculus, but that is not necessary for our purpose. All we need to do is to extend 2.6 and this we now do.

THEOREM II A closed term τ of type 0 of the ν^+-calculus in normal form contains no symbols other than 0, S^0, $+_0$ (and parentheses).

The proof of this is distinctly more tiresome than for the λ-calculus. We apologise in advance for the excessive use of *reduction and absurdum*; more direct proofs seem necessarily to involve much subdivision into cases and several repeated arguments.

Proof of Theorem II. In analysing the forms of terms we ignore parentheses. A term is *peculiar* if it is of type $\alpha_1 \times \alpha_2$ and not of the form $P\sigma_1\sigma_2$, or of the type $\alpha_1 + \alpha_2$ and not of the form ${}^k\iota\sigma$. A term is *standard* if it is a λ term or a ν term or if it is one of the form $P\sigma$, $P\sigma_1\sigma_2$, ${}^k\iota\sigma$. A term is *non-numerical* if it is not 0_o and if it does not begin with S^0 nor with $+_0$ (which we are now supposing to be written in front of its arguments).

Observe that peculiar and standard terms are non-numerical and cannot be of type 0; also that a peculiar term is not standard.

LEMMA 1 If ρ is a closed non-numerical term in normal form which consists of more than one symbol, then either ρ is standard, or it contains a proper sub-term which is both closed and peculiar.

Proof. Suppose ρ satisfies the premise and is not standard. It cannot be a λ or a ν term, and so must have the form $(\rho_1\ \rho_2)$. Here ρ_1 cannot begin with λ since ρ is in normal form; nor can ρ_1 begin with P or ${}^k\iota$, for then ρ would be standard. Hence either ρ_1 is ${}^k\Pi$ and ρ_2 is a closed peculiar term, or ρ_1 is, or begins with, a ν term σ_1 and then ρ begins with, or is, a term $(\sigma_1\ \sigma_2)$ where σ_2 is closed and peculiar.

COROLLARY 1 A peculiar term in normal form cannot be closed.

For consider, if possible, a minimal closed peculiar term in normal form. It satisfies the premise of Lemma 1 and is not standard, contradicting the conclusion.

COROLLARY 2 If ρ satisfies the premise of the lemma it is standard.

LEMMA 2 If τ is a closed term of type 0 in normal form then it cannot have any standard sub-terms.

Proof. Consider, if possible, a maximal standard sub-term σ of τ. Then σ cannot be a part of a λ term or a ν term and so it is closed and must occur in a closed part ρ of τ having the form $\rho = (\sigma\pi)$ or $\rho = (\pi\sigma)$. By the observation preceeding Lemma 1, σ is non numerical and not of type 0. But then in either case ρ is non numerical, and therefore satisfies the promise of Lemma 1; so, by Corollary 2, ρ is standard, contradicting the maximality of σ.

Proof of Theorem II resumed . Let τ be a closed term of type 0 in normal form. By Lemma 2 τ cannot contain λ or ν , and hence cannot contain any variables. Suppose that τ contains one of the constants P, ${}^{k}\iota$, ${}^{k}\Pi$ and let σ be the leftermost occurrence of such a constant. Since σ is not of type 0 it must occur in a part of τ of the form $(\sigma\rho)$. By Lemma 2, σ cannot be P or ${}^{k}\iota$; and if it were ${}^{k}\Pi$ then ρ would be closed and peculiar, contradicting Corollary 1. Thus τ can only contain the symbols 0_o, S^0, $+_0$, as was to be proved.[2]

FOOTNOTES

1. I was stimulated to discover the idea of the proof presented here by my distaste for the proof given in Prawitz (1971) which he presented when he was visiting Oxford in 1975-76. A manuscript version of the idea (dated March 1976) has been in circulation for sometime; the occasion of Haskell Curry's 80-th birthday provided the necessary spur to produce a printed and fuller version.

2. *Notes added in proof*. (i) It seems that everyone who extends the typed λ-calculus to include cartesian product,

direct sum and so on does it in their own way. The advantage of the ν-terms introduced here is that they follow closely the category-theoretic approach. But other notations may in fact be simpler to handle; in particular one can replace a ν-term by a term built up from two λ-terms. Martin-Löf ((in 1975) and many mimeographed typescripts) is perhaps the ultimate authority on extensions of simple type-theory; but he has been more concerned to bring out the meaning of the various operations than to provide a smooth-running mechanism of notations.

(ii) Since I handed over my manuscript to the typist I have learnt of two other forms of proof of strong normalization.

(a) Nederpelt (1973) replaces β-reduction by 'β_1-reduction' in which each reduction step leaves a trace of the unreduced term behind. The hard work in the proof is then concentrated in proving a Church-Rosser theorem for β_1-reduction. (b) Minc (1979) (of which I have only read the abstract) gives a way of computing the maximum length of a reduction sequence from the length of a standard reduction sequence. In this connection the following remarks may be of interest. Let τ be a closed λ^+-term of type 0, let n be its length (not counting type subscripts) and let p be the length of the type-symbol for the type of the subterm(s) of highest type occurring in τ. By considering the standard reduction sequence it is a straightforward matter to compute an upper bound for the value of τ; it can be given in the form of exponentiating n to the base 2 p' times were p' increases with p. The work of Statman (1979) suggests that a lower bound for the value of τ will have the same form. In other words a function which takes one from a description of a term to its value may not be elementary. On the other hand it is not hard to show that, for τ as above, the length of τ^* is bounded by n.p' where p' is got by exponentiating p a small fixed number of times. The complexity of the function giving the maximum length of a reduction sequence is only slightly greater than the complexity of the function giving the minimum length. It seems to me just possible that this will be false if one allows Prawitz's reductions (see 5.9).

(3) The remark made in 5.3 about 'strong ∨' is ill informed. However in the λ-ν-π^+ calculus, + is strong in the following sense. Set

$C = \nu x_{\alpha_1} y_{\alpha_2} [O, SO]$, $D^2 = \nu x_{\alpha_1} y_{\alpha_2} [x_{\alpha_1}, L_{\alpha_1}]$, $D^2 = \nu x_{\alpha_1} y_{\alpha_2} [L_{\alpha_2}, y_{\alpha_2}]$;

then $C({}^1\iota\sigma)$ reduced to O, $C({}^2\iota\tau)$ reduces to SO, $D^1({}^1\iota\sigma)$ reduces to σ and $D^2({}^2\iota\tau)$ reduces to τ, where σ is of type α_1, τ of type α_2. (This footnote added in proof.)

REFERENCES

Gödel, K. (1958). Über eine bisher nicht benutzte Erweiterung des finiten Standpunktes, *Dialectica* 12, 280-287.

Howard, A.A. (1972). A system of abstract constructive ordinals, *Journal of Symbolic Logic* 37, 355-374.

Martin-Löf, P. (1975). An intuitionistic theory of types: predicative part. *In* "Logic Colloquium '73" (Ed. H.E. Rose and J.C. Shepherdson), pp. 73-118. North-Holland Pub. Co., Amsterdam-London.

Minc, G.E. (1979). A primitive recursive bound of strong normalization for predicate calculus (in Russian with English abstract), *Zapiski Naucnyh Seminarov Leningradskogo Otdelenija Matematiceskogo Instituta im V.A. Steklova Akademii Nauk SSSR (LOMI)* 88, 131-135.

Nederpelt, R.P. (1973). Strong normalization in a typed lambda calculus with lambda structured types, Proefschrift, Technische Hogeschool, Eindhoven.

Prawitz, D. (1971). Ideas and results in proof theory. *In* "Proceedings of the 2nd Scandinavian Symposium" (Ed. J.E. Fenstad), pp. 235-307. North-Holland Pub. Co. Amsterdam-London.

Statman, R. (1979). The typed λ-calculus is not elementary, *Theoretical Computer Science* 9, 73-81.

Troelstra, A.S. (Editor) (1973). "Metamathematical investigations of intuitionistic arithmetic and analysis" (Lecture Notes in Mathematics No. 344), Springer-Verlag, Berlin-Heidelberg-New York.

THE FORMULAE-AS-TYPES NOTION OF CONSTRUCTION

W. A. Howard

Department of Mathematics, University of Illinois at Chicago Circle, Chicago, Illinois 60680, U.S.A.

Dedicated to H. B. Curry on the occasion of his 80th birthday.

The following consists of notes which were privately circulated in 1969. Since they have been referred to a few times in the literature, it seems worth while to publish them. They have been rearranged for easier reading, and some inessential corrections have been made.

The ultimate goal was to develop a notion of construction suitable for the interpretation of intuitionistic mathematics. The notion of construction developed in the notes is certainly too crude for that, so the use of the word *construction* is not very appropriate. However, the terminology has been kept in order to preserve the original title and also to preserve the character of the notes. The title has a second defect; namely, a *type* should be regarded as a abstract object whereas a *formula* is the name of a type.

In Part I the ideas are illustrated for the intuitionistic propositional calculus and in Part II they are applied to Heyting arithmetic.

I. INTUITIONISTIC PROPOSITIONAL CALCULUS

H. Curry (1958) has observed that there is a close correspondence between *axioms* of positive implicational propositional logic, on the one hand, and *basic combinators* on the other hand. For example, the combinator $K = \lambda X.\lambda Y.X$ corresponds to the axiom $\alpha \supset (\beta \supset \alpha)$.

The following notion of construction, for positive implicational propositional logic, was motivated by Curry's observation. More precisely, Curry's observation provided *half* the motivation. The other half was provided by W. Tait's discovery of the close correspondence between cut elimination and reduction of λ-terms (W. W. Tait, 1965). It is convenient to use λ-terms rather than combinators. This corresponds to the sequent formulation of propositional logic.

1. Formulation of the sequent calculus

Let $P(\supset)$ denote positive implicational propositional logic. The prime formulae of $P(\supset)$ are propositional variables. If α and β are formulae, so is $\alpha \supset \beta$. A *sequent* has the form $\Gamma \to \beta$, where Γ is a (possibly empty) finite sequence of formulae and β is a formula. The axioms and rules of inference of $P(\supset)$ are as follows.

(1.1) Axioms: all sequents of the form
$$\alpha \to \alpha$$

(1.2)
$$\frac{\Gamma, \alpha \to \beta}{\Gamma \to \alpha \supset \beta}$$

(1.3)
$$\frac{\Gamma \to \alpha \qquad \Delta \to \alpha \supset \beta}{\Gamma, \Delta \to \beta}$$

(1.4) Thinning, permutation and contraction rules

2. Type symbols, terms and constructions

By a type symbol is meant a formula of $P(\supset)$. We will consider a λ-formalism in which each term has a type symbol α as a superscript (which we may not always write); the term is said to be of type α. The rules of term formation are as follows.

(2.1) Variables $X^{\alpha}, Y^{\beta}, \ldots$ are terms

(2.2) λ-abstraction: from F^{β} get
$$(\lambda X^{\alpha}.F^{\beta})^{\alpha \supset \beta} .$$

(2.3) Application: from $G^{\alpha \supset \beta}$ and H^{α} get $(G^{\alpha \supset \beta} H^{\alpha})^{\beta}$.

By a *construction* of a sequent $\Gamma \to \beta$ is meant a term F^{β} of type β such that for every free variable X^{α} occurring in F^{β} there is a corresponding occurrence of α in Γ (it being understood that the existence of k distinct free variables of the same type α in F^{β} is reflected by at least k occurrences of α in Γ) . Thus X^{α} is a construction of $\alpha \to \alpha$ (but X^{α} is also a construction of $\beta, \alpha \to \alpha$). Another example: $\lambda X^{\alpha}.\lambda Y^{\beta}.X^{\alpha}$ is a construction of $\to \alpha \supset (\beta \supset \alpha)$.

3. *Correspondence between derivation and terms*

Clearly the axioms and rules of inference 1.1-1.3 of $P(\supset)$ correspond exactly to the rules 2.1-2.3 of term formation. A construction of $\Gamma \to \beta$ is clearly also a construction of $\Gamma, \alpha \to \beta$ (thinning); similarly for a permutation of Γ ; and the contraction $\dfrac{\Gamma,\ \alpha,\ \alpha \to \beta}{\Gamma,\ \alpha \to \beta}$ corresponds to replacing two distinct variables of type α by one variable of type α in the corresponding construction.

Hence:

THEOREM 1. Given any derivation of $\Gamma \to \beta$ in $P(\supset)$ we can find a construction of $\Gamma \to \beta$ and conversely.

4. *Interpretation of terms*

For an interpretation in ordinary set theory let each propositional variable (i.e., prime type symbol) denote a specific set of basic objects. Then every type symbol can be taken to denote a set of things according to the rule: $\alpha \supset \beta$ denotes the set of all functions whose domain is a superset of α and whose range is a subset of β . (According as to whether the superset depends on the function in question, or whether it just depends on α, we get somewhat differing interpretations). The variables of type α are interpreted as ranging over the set α .

It is now clear, by induction on the rules 2.1-2.3 of term formation, how each term is to be interpreted as a function of the objects over which its free variables range. Thus the closed terms can be interpreted as a perfectly concrete set of functionals of finite type over the basic objects. This interpretation is used by H. Laüchli in the paper he read at the Summer Conference at Buffalo in 1968: see (Laüchli, 1970) pp. 227-229.

Of course a constructivist would be interested in other interpretations; for example, interpretations related to the *calculation* of terms (i.e., reduction to irreducible form). It is easy to prove that the terms given above can be reduced to normal (i.e., irreducible) form by λ-contractions. The relation between this and *cut elimination* will now be discussed briefly.

5. *Normalization of terms and cut elimination*

Clearly the cut rule for $P(\supset)$ corresponds to the following rule of term formation: from F^α and G^β get $[F^\alpha/X^\alpha]G^\beta$ (the result of substituting F^α for the free variable X^α in G^β, where no free variable in F^α becomes bound in $[F^\alpha/X^\alpha]G^\beta$). Though we did not include substitution in our rules of term formation, the rule 2.3 (application) is just about as bad--from the viewpoint of obtaining irreducible terms. Professor Curry is fond of pointing out how to get irreducible terms: simply replace the rule 2.3 by:

(5.1) from a variable X of type $\alpha_1 \supset (\alpha_2 \supset (\ldots \supset (\alpha_n \supset \beta)\ldots))$ and terms $F_1,\ldots, F_n$ of types $\alpha_1,\ldots,\alpha_n$, respectively, get the term $XF_1\ldots F_n$ of type β .

Correspondingly, replace the rule 1.3 of $P(\supset)$ by the n-premise rule

$$(5.2)\qquad \frac{\Gamma_1 \to \alpha_1 \quad \Gamma_2 \to \alpha_2 \ \dots \ \Gamma_n \to \alpha_n}{\Gamma_1, \Gamma_2, \dots, \Gamma_n,\ \alpha_1 \supset (\alpha_2 \supset (\dots \supset (\alpha_n \supset \beta)\dots)) \to \beta}\ .$$

Of course 5.2 can be obtained by n applications of one of the Gentzen rules

$$(5.3)\qquad \frac{\Gamma \to \alpha \qquad \beta, \Delta \to \gamma}{\alpha \supset \beta,\ \Gamma,\ \Delta \to \gamma}\ ,$$

with γ equal to β, and use of $\beta, \Delta \to \beta$. We could replace 5.1 by a rule of term formation corresponding to 5.3, but 5.1 seems more natural. As a modification of Theorem 1 we have:

THEOREM 2. Let $P^*(\supset)$ be $P(\supset)$ with the rule 1.3 replaced by 5.2. ThenI given any derivation of $\Gamma \to \beta$ in $P^*(\supset)$ we can find an irreducible construction of $\Gamma \to \beta$ and conversely.

Cut *elimination* can be taken to mean: the transformation of a proof of $\Gamma \to \beta$ in $P(\supset)$ into a proof of $\Gamma \to \beta$ in $P^*(\supset)$. Thus cut elimination can be obtained as a consequence of the reduction of terms to normal form. As mentioned in §4, such reduction is easy to prove for the terms under discussion. Results following from cut elemination in $P(\supset)$ (e.g.) the nonderivability of Peirce's Law $(\alpha \supset \beta. \supset \alpha) \supset \alpha)$ seem to be obtainable at least as easily from the normalizability of constructions.

6. *Addition of* $\neg$, $\wedge$ *and* $\vee$ *to* $P(\supset)$

Corresponding to each of these connectives we add certain closed prime terms to our supply of terms.

(i) For $\neg$: add a new prime formula f to $P(\supset)$. Then, for each formula α, introduce a term $A^{f \supset \alpha}$. As an exercise, the reader may wish to prove--for the resulting system--that there are no closed terms of type f . (By normalizability it is sufficient to prove this for irreducible terms). There *are* open terms of type f; for example, the variable X^f--which is a construction of $f \to f$.

(ii) For $\wedge$: add terms $B_1^{\alpha \supset (\beta \supset \alpha \wedge \beta)}$, $B_2^{\alpha \wedge \beta \supset \alpha}$ and $B_3^{\alpha \wedge \beta \supset \beta}$. These are just pairing and projection functionals ($\alpha \wedge \beta$ is the type of a pair of terms of types α and β). We do not need to add a term of type $\beta \supset (\alpha \supset \alpha \wedge \beta)$ because such a term can be defined as $\lambda Y^\beta . \lambda X^\alpha . B_1 X^\alpha Y^\beta$.

In connection with the theory of reducibility of constructions it is useful to postulate the contraction $B_2(B_1FG)$ contr F and $B_3(B_1FG)$ contr G. Then we get the following theorem.

THEOREM 3. Every closed irreducible term of type $\alpha \wedge \beta$ has the form $B_1 F^\alpha G^\beta$, where B_1 is as above.

[*Note added 1979* This treatment of $\neg$ and $\wedge$ does not seem to be appropriate for cut elimination in Gentzen's sequent calculus. As P. Martin-Löf soon pointed out to me, it *is* appropriate for D. Prawitz's theory of Gentzen's system of natural deduction. See Prawitz (1965). The terms $A^{f \supset \alpha}$ and $B_1^{\alpha \supset (\beta \supset \alpha \wedge \beta)}$ correspond to the inference rules $\frac{f}{\alpha}$ and $\frac{\alpha \quad \beta}{\alpha \wedge \beta}$, respectively, while $B_2^{\alpha \wedge \beta \supset \alpha}$ and $B_3^{\alpha \wedge \beta \supset \beta}$ correspond to $\frac{\alpha \wedge \beta}{\alpha}$ and $\frac{\alpha \wedge \beta}{\beta}$, respectively.]

(iii) For $\vee$: there are two possibilities, corresponding to the discussion of weak existence and strong existence in §12, below. Corresponding to the case of weak existence we add terms $C_1^{\alpha \supset \alpha \vee \beta}$, $C_2^{\beta \supset \alpha \vee \beta}$ and $C_3^{\alpha \vee \beta \supset (\alpha \supset \gamma . \supset (\beta \supset \gamma . \supset \gamma))}$

It is useful to postulate the contractions $C_3(C_1M)FG$ contr FM and $C_3(C_2N)FG$ contr GN for all terms M, N, F, G of types α, β, $\alpha \supset \gamma$, $\beta \supset \gamma$, respectively. Then we get the following theorem.

THEOREM 4. Every closed irreducible term of type $\alpha \vee \beta$ has the form $C_1 F^\alpha$ or $C_2 G^\beta$, where C_1 and C_2 are as above.

II. HEYTING ARITHMETIC

We will be concerned mainly with the subsystem $H(\supset, \wedge, \forall)$ of Heyting arithmetic obtained by omitting $\vee$ and $\exists$. As is well-known, $\neg\alpha$ can be defined as $\alpha \supset 0 = 1$. In §12 we will make some remarks about the question of including $\exists$. Of course $\vee$ can be defined by use of $\exists$. The variables belonging to $H(\supset, \wedge, \forall)$ will be called *number variables*.

7. *Constructions*

Our constructions will be terms built up from prime terms by means of rules of term formation as indicated in (ii)-(iv), below. Every term is supplied with a unique type symbol. The numerical terms--namely, the terms belonging to $H(\supset, \wedge, \forall)$--have type 0.

(i) *Type symbols* The prime type symbols are: 0 and every equation of $H(\supset, \wedge, \forall)$. From these we generate all type symbols by the following two rules.

(a) From α and β get $\alpha \supset \beta$ and $\alpha \wedge \beta$.

(b) From α and a number variable x get $\forall x\alpha$.

(ii) *Prime terms* These are:

(a) number variables $x, y, \ldots$; constants 0 and 1; function symbols for plus and times,

(b) variables X^α, $Y^\beta, \ldots$,

(c) certain special terms, mentioned in §8, below, corresponding to axioms and rules of inference of $H(\supset, \wedge, \forall)$.

(iii) λ-*abstraction*:

(a) From F^β get $(\lambda X^\alpha . F^\beta)^{\alpha \supset \beta}$ as in §2.

(b) If x does not occur free in the type symbol of any free variable of F, form $(\lambda x F^\beta)^{\forall x\beta}$.

(iv) *Application:*

(a) From F^α and $G^{\alpha \supset \beta}$ form $(GF)^\beta$ as in §2.

(b) From $G^{\forall x\alpha(x)}$ and t of type 0 form $G(t)^{\alpha(t)}$.

8. *Special terms*

(i) terms of types $x + 0 = x$ and $x + (y + 1) = (x + y) + 1$,

(ii) terms of types $x\cdot 0 = 0$ and $x\cdot(y+1) = x\cdot y + x$,

(iii) a term of type $x = x$,

(iv) a term of type $x = y \supset t(x) = t(y)$ for each term $t(x)$ of type 0,

(v) the terms B_1, B_2 and B_3 discussed in §6(ii),

(vi) a term $R^{\forall y\beta(y)}$ for each $\beta(y)$ of the form $\alpha(0) \supset [\forall x(\alpha(x) \supset \alpha(x+1)) \supset \alpha(y)]$, also a term $R_n^{\beta(n)}$ for each numeral n.

9. Constructions and derivations in $H(\supset, \wedge, \forall)$

Define constructions as in §2. As in the case of $P(\supset)$--see §3-- the axioms of $H(\supset, \wedge, \forall)$ correspond to the existence of certain terms, and the rules of inference correspond to rules of term formation. In particular $\forall x\alpha(x) \to \alpha(t)$ has the construction $Y^{\forall x\alpha(x)}t$. If F is a construction of $\Gamma \to \alpha$, then $\lambda x.F$ is a construction of $\Gamma \to \forall x\alpha$. If $G(X^{\alpha(t)})$ is a construction of $\Gamma, \alpha(t) \to \beta$, then $G(Y^{\forall x\alpha(x)}t)$ is a construction of $\Gamma, \forall x\alpha(x) \to \beta$. Thus we obtain:

THEOREM 5. Given any derivation of $\Gamma \to \beta$ in $H(\supset, \wedge, \forall)$ we can find a construction of $\Gamma \to \beta$ and conversely.

10. Interpretation of terms of $H(\supset, \wedge, \forall)$

We extend the discussion of §4 as follows.

(i) We interpret each closed formula α as a set α^* in the following manner. If α is a closed equation, then α^* is the singleton set $\{1\}$ if the equation is true and the set $\{0\}$ if the equation is false. If $\alpha(n)^*$ has been defined for each numeral n, then define $(\forall x\alpha(x))^*$ as the set of all functions f such that $f(n) \in \alpha(n)^*$ for every n. Define $(\alpha \supset \beta)^*$ in terms of α^* and β^* as in §4. Define $(\alpha \wedge \beta)^*$ as the Cartesian product of α^* and β^*.

(ii) To each term we associate an object F^* by use of the following clauses. It should be noted that when F is closed, then the type symbol of F is a closed formula α; and F^* is an element of α^* .

(a) Suppose F has the form $G(x_1,\ldots,x_k)$, where $x_1,\ldots,x_k$ are the free number variables of F . Assuming that $G(n_1,\ldots,n_k)^*$ has been defined for all numerals $n_1,\ldots,n_k$, we define F^* to be the mapping which sends each k-tuple $n_1,\ldots,n_k$ into $G(n_1,\ldots,n_k)^*$.

(b) Suppose F has no free number variables. If F has the form $\forall x.G(x)$, then define F^* as the mapping which takes each numeral n into $G(n)^*$. If F does not have this form, then the free variables of F will have types $\beta_1,\ldots,\beta_k$ which have been interpreted as sets $\beta_1^*,\ldots,\beta_k^*$ in (i), above, and F^* will be a mapping from the Cartesian product of $\beta_1^*,\ldots,\beta_k^*$ into α^* . In particular, if F has the form $\lambda Y^\beta.G$, then F^* is defined in terms of G^* by the usual interpretation of λ-abstraction. To be systematic we must allow free variables, here and in (a), to occur vacuously.

(c) If F has the form $R_n^{\beta(n)}$ as in §8(vi), then F^* is defined by primitive recursion on n . If F is $R^{\forall y\beta(y)}$, then F^* is the mapping which takes each numeral n into $(R_n^{\beta(n)})^*$.

11. Normalization of terms

For the theory of reducibility of terms we postulate the following contraction schemes

(i) $$(\lambda X.F(X))^{\alpha\supset\beta}G \quad \text{contr} \quad F(G)^\beta$$
$$(\lambda x.F(x))^{\forall x\alpha(x)}t \quad \text{contr} \quad F(t)^{\alpha(t)}$$

(ii) $$B_2(B_1FG) \quad \text{contr} \quad F$$
$$B_3(B_1FG) \quad \text{contr} \quad G$$

(iii) $$R_0^{\beta(0)}FG \quad \text{contr} \quad F$$
$$R_{n+1}^{\beta(n+1)}FG \quad \text{contr} \quad Gn(R_n^{\beta(n)}FG)$$
$$R^{\forall x\beta(x)}n \quad \text{contr} \quad R_n^\beta(n) \quad .$$

Also we consider the steps in the calculation of closed numerical terms to be reduction steps.

By the method of Tait (1967) it is easy to prove:

THEOREM 6. Every term can be reduced to irreducible form.

Theorem 3 of §6 extends to the present terms. By use of Theorem 3 it is easy to show that there is no irreducible construction of a sequent of the form $\to m = n$ with $m \neq n$. Thus by finitistic reasoning it can be proved that Theorem 6 implies the consistency of $H(\supset, \wedge, \forall)$.

12. *Introduction of* $\exists$

How are we to handle the existential quantifier in our theory of constructions? This appears to be a nontrivial question. The following two alternatives suggest themselves.

(i) *Weak existence* In the formulation of $H(\supset, \wedge, \forall)$ take the sequents

$$\to \alpha(t) \supset \exists y\alpha(y)$$
$$\to \exists y\alpha(y) \supset (\forall x(\alpha(x) \supset \beta) \supset \beta)$$

(x not free in β) to be axioms and introduce new prime terms C_1 C_2 of types $\alpha(t) \supset \exists y\alpha(y)$ and $\exists y\alpha(y) \supset (\forall x(\alpha(x) \supset \beta) \supset \beta)$, respectively. Postulate the contraction $C_2(C_1 F)G$ contr GtF for all terms F and G of types $\alpha(t)$ and $\forall x(\alpha(x) \supset \beta)$, respectively. Theorems 3, 5 and 6 extend to $H(\supset, \wedge, \forall, \exists)$. Corresponding to Theorem 4 we have:

THEOREM 7. Every closed irreducible term of type $\exists y\alpha(y)$ has the form $C_1^{\alpha(n) \supset \exists y\alpha(y)} F^{\alpha(n)}$.

Thus from an irreducible construction of $\to \exists y\alpha(y)$ we get a numeral n and an irreducible construction of $\to \alpha(n)$ (assuming $\exists y\alpha(y)$ closed).

(ii) *Strong existence (choice operators)* It is natural to interpret an object of type $\exists y\alpha(y)$ as a pair $\langle t, F^{\alpha(t)} \rangle$. Thus, in §10, $(\exists y\alpha(y))^*$ would be defined as the set of all pairs $\langle n,Z\rangle$ such that $Z\varepsilon\alpha(n)^*$. Hence introduce projection operators P_1 and P_2 which give the required components t and $F^{\alpha(t)}$ when applied to a pair $\langle t,F^{\alpha(t)}\rangle$ regarded as an object of type $\exists y\alpha(y)$. The operators P_1 and P_2 can be considered to have types $\exists y\alpha(y) \supset 0$ and $X^{\exists y\alpha(y)}\alpha(P_1 X^{\exists y\alpha(y)})$, respectively. This takes us outside the formalism of $H(\supset,\wedge,\forall,\exists)$: the type symbols of P_1 and P_2 are not formulae of $H(\supset,\wedge,\forall,\exists)$. Nevertheless, it is clear what meaning we are to attach to P_1 and P_2. Namely, P_1 applies to an object F of type $\exists y\alpha(y)$ and yields an object P_1F of type 0; and P_2 when applied to F yields an object of type $\alpha(P_1F)$.

To illustrate the use of P_1 as a choice operator, let $\alpha(x,y)$ be a formula of $H(\supset,\wedge,\forall,\exists)$ with free variables x and y, and let F be a construction of type $\exists y\alpha(x,y)$. Then $\lambda x.P_1F$ is a term ϕ satisfying $\forall x\alpha(x,\phi(x))$.

In general a type symbol of the form $\forall X^\alpha\beta$ arises by λ-abstraction from a term F^β, where the variable X^α occurs in β (as well as in other locations in F). A theory of such terms would extend the theory developed in the preceding sections. In the latter theory the only variables occurring in the type symbols have type 0 (they are, namely, numerical variables of $H(\supset,\wedge,\forall,\exists)$).

It might be of interest to know the answer to the following question. Let G be a closed term obtained by extending our supply of constructions in the manner just described. Suppose G has type α where α is a formula of $H(\supset,\wedge,\forall,\exists)$. Must there be a derivation of $\to\alpha$ in $H(\supset,\wedge,\forall,\exists)$?

13. *Infinite constructions*

As is well-known, we do not have *cut elimination* for

$H(\supset,\wedge,\forall)$ unless the axiom of mathematical induction is replaced by an ω-rule. There is no difficulty in developing a theory of constructions for the ω-rule version of $H(\supset,\wedge,\forall)$. In fact, if one uses Tait's notion of infinite terms (Tait, 1967), one gets a very simple theory (modulo the question of handling infinite terms constructively).

REFERENCES

Curry, H. B. and Feys, R. (1958). "Combinatory Logic", North-Holland, Amsterdam.

Lauchli, H. (1970). An abstract notion of realizability for which intuitionistic predicate calculus is complete. In "Intuitionism and Proof Theory. Proceedings of the Summer Conference at Buffalo N.Y." (Eds. G. Myhill, A. Kino, R. E. Vesley) pp. 227-234. North-Holland, Amsterdam.

Prawitz, D. (1965). "Natural Deduction" Almqvist and Wiksell, Stockholm.

Tait, W. W. (1965). Infinitely long terms of transfinite type. *In* "Formal Systems and Recursive Functions" (Eds. J. N. Crossley and M. A. E. Dummett) North-Holland, Amsterdam.

Tait, W. W. (1967). Intensional interpretations of functionals of finite type, *Journal of Symbolic Logic* 32, 198-212.

MODIFIED REALIZATION AND THE FORMULAE-AS-TYPES NOTION

Justus Diller

Institut für mathematische Logik und Grundlagenforschung, Westfälische Wilhelms-Universität, Roxeler Straße 64, D - 4400 Münster, Germany

Dedicated to H.B. Curry on the occasion of his 80th birthday

There is an intriguing isomorphism between intuitionistic logics and typed λ-calculi. Under it, formulae correspond to types and natural deductions correspond to λ-terms. This formulae-as-types notion originated with Curry's theory of functionality, developed in Curry-Feys 1958. It was extended to predicate logic by Howard 1969 and used to prove normalization theorems for simple type theory by Girard 1971, Tait 1972, and Pohlers 1973. A radical view is taken by Martin-Löf 1975 whose predicative theory of types contains rules for the existential quantifier that formally coincide with the natural deduction rules for conjunction. This directly formalizes the intuitive idea that a proof of a statement $\exists xB$ is a pair $\langle a,b\rangle$ consisting of an object a and a proof of B[a].

We want to compare the process of normalization to the process of Kreisel's 1959 modified realization. For this purpose, it seems natural to make use of the formulae-as-types notion and study the transformations that these processes perform on terms. For terms a of a natural deduction version of Heyting arithmetic in all finite types, we construct a primitive recursive term a' and a term a^* which proves that a' modified realizes the type of a, that is, the formula proved by a. By theorem 2, normalization commutes with both of these constructions (homomorphisms) $'$ and *. The theorem of Minc 1973 is a corollary of this. Minc also uses a construction like $'$, but he seems to disregard *.

Stein 1976 and Rath 1978 have meanwhile extended our result to other functional interpretations, so far not using the formulae-as-types notion. Modified realization as a very simple functional interpretation, however, leads to a very transparent proof that it commutes with normalization.

We follow Martin-Löf 1975 as closely as seems feasible in formulating an extended system of Heyting arithmetic of finite types, using partly the notation of Howard 1969. We construct expressions of the form $a \vdash A$, a *is a term of type* A, which is interpreted as "a is an object of type A" or "a is a proof of the formula A", and a red b, a *reduces to* b. Variables may be bound by $\lambda, \forall, \exists$, but we identify expressions which differ only by a change of bound variables. An occurrence of a variable in an expression generally means a free occurrence; if we write b[x], the x denotes all free occurrences of x in the expression b, and b[a] then denotes the result of replacing every free occurrence of x in b by a. If we use this notation, we assume that a is substitutable in b. Let $x_1,\dots,x_n$ be variables and let b be an expression. We say that $x_1,\dots,x_n$, b *satisfy the c.o.v.* (condition on variables), if x_i does not occur in the type of any variable free in b, except at most for the types of the variables x_k with $k > i$.

INDUCTIVE DEFINITION 1 of *types, terms with their types,* and *reductions*.

1. o is a type.
2. If A and B are types and x^A, B satisfy the c.o.v., then $\forall x^A B$ and $\exists x^A B$ are types.
3. If a,b are terms of the same type, then a=b is a type.

4. For any type A, the variable x^A is a term of type A. (We may drop the type superscript, if it is clear from the context.)
5. ($\forall$I) If $b \vdash B$ and x^A, b (as well as x^A,B) satisfy the c.o.v., then $\lambda x^A\, b \vdash \forall x^A B$.
   ($\forall$E) If $c \vdash \forall x^A B[x^A]$ and $a \vdash A$, then $c(a) \vdash B[a]$.
   ($\forall$red) $(\lambda x^A.\, b[x^A])(a)$ red $b[a]$.
6. ($\exists$I) If $a \vdash A$ and $b \vdash B[a]$, then $<a,b> \vdash \exists x^A B[x^A]$.
   ($\exists$E) If $c \vdash \exists x^A B[x^A]$, then $j_1 c \vdash A$ and $j_2 c \vdash B[j_1 c]$.
   ($\exists$red) $j_1<a,b>$ red a and $j_2<a,b>$ red b.
7. (oI) $0 \vdash o$, and if $t \vdash o$ then $suc(t) \vdash o$.
   (oE) If $a \vdash F[0]$, $b \vdash F[suc(x)]$, $t \vdash o$, and x^o, $y^{F[x]}$, b satisfy the c.o.v., then $R(a,\lambda x^o y^{F[x]}.b,t) \vdash F[t]$.
   (o red) $R(a,\overline{b},0)$ red a and $R(a,\overline{b},suc(t))$ red $b[t,R(a,\overline{b},t)]$, where $\overline{b}$ stands for $\lambda x^o y^{F[x]}.b[x,y]$.
8. (=I) If $a \vdash A$, then $r(a) \vdash a=a$.
   (=E) If $c \vdash C[a]$ and $d \vdash a=b$, then $eq(c,d,a,b) \vdash C[b]$.
   (=red) $eq(c,r(a),a,a)$ red c.
9. If $a \vdash A$ and A red B or B red A, then $a \vdash B$.
10. Let a,b,c be terms or types.
   (refl) a red a.
   (trans) If a red b and b red c, then a red c.
   (repl) If a red b then c[a] red c[b].

Remarks. If x^A does not occur in B, we write $A \to B$ (A *implies* B) for $\forall x^A B$ and $A \wedge B$ (A *and* B) for $\exists x^A B$. We take falsehood, negation, disjunction, and equivalence to be defined operations:

$\bot := suc(0) = 0$ and $\neg A := A \to \bot$

$A \vee B := \exists x^o((x=0 \to A) \wedge (x \neq 0 \to B))$

$A \leftrightarrow B := (A \to B) \wedge (B \to A)$

The *extended system* of definition 1 corresponds to the initial fragment V_o of Martin-Löf's 1975 hierarchy V. It has the following basic properties:

SUBSTITUTION LEMMA. If $a \vdash A$ and $b[x^A] \vdash B[x^A]$, then $b[a] \vdash B[a]$.

REDUCTION LEMMA. If a red b then a and b have the same type.

In this extended system, arbitrary (deduction) terms may enter into types, in particular by the strong $\exists$-elimination rule ($\exists$E). However, we want to study modified realization which is a technique to extract primitive recursive terms of finite type from proofs of arithmetical formulae. We shall therefore separate the deduction terms from the types as far as seems necessary to make our construction work.

DEFINITION 2. The *finite types* are the types generated by 1 and 2 of defn 1, i.e. the finite types are the closure of o under $\rightarrow$ and $\wedge$. The *primitive recursive terms* (PRT) are the terms all of whose subterms are of finite type, i.e. the PRT are the closure of the variables of finite type under the rules 5,6, and 7 of defn 1. A term is called *$\exists$-free*, if in its construction the rules ($\exists$I) and ($\exists$E) have only been used to introduce and eliminate conjunctions.

Obviously, if a is a PRT and a red b, then b is a PRT. We use ρ,σ,τ to denote finite types.

DEFINITION 3. The *restricted system* is obtained from the system of defn 1 by the following restrictions:
In 2, add the premiss: A is a finite type, or x^A does not occur in B. In ($\forall$E) and ($\exists$I), add the premiss: for finite types A, a is a PRT.
($\exists$E) takes the form: If $c \vdash \exists x^A B[x^A]$ and $d[x^A, y^{B[x]}] \vdash D$, where x,y do not occur in D and x,y,d satisfy the c.o.v., then $d[j_1 c, j_2 c] \vdash D$.
($\exists$red) takes the form: $d[j_1\langle a,b\rangle,\ j_2\langle a,b\rangle]$ red $d[a,b]$.
In (oE) add the premiss: t is a PRT.
In (=E) add the premiss: C[a] is an equation.

Due to the restrictions on 2, (∀E), (∃I), and (oE), genuine quantification is used only over finite types, and the terms that enter into a deduction and its type at the same time are PRT. The restriction on (=E) does not weaken the system. The essential restriction is the one on (∃E) which now has the usual weak form as in Prawitz 1965, ch. I, § 2. Our restricted form of (∃E) and (∃red), however, includes ∧-elimination and ∧-reduction as well as the commutative ∃-reductions of Prawitz 1965, ch. IV, § 1.

The restricted system is essentially a natural deduction version of N-HA$^{\omega}$(cf. Troelstra 1973, ch. I, § 6). The extended system is not conservative over it, since the axiom of choice is provable in the extended system (Martin-Löf 1975, § 2.13), but not in the restricted system.

The class of terms of the restricted system is not closed under substitution, since there are terms of finite type which are not PRT. It is only closed under the following kinds of substitution:

(1) If $a \vdash \sigma$ is a PRT and $b[x^{\sigma}]$ belongs to the restricted system, then so does $b[a]$.

(2) If A is not a finite type and $a \vdash A$ and $b[x^{A}]$ belong to the restricted system, then so does $b[a]$.

This suffices for the restricted system to be closed under reductions:

CLOSURE LEMMA. If a red b and a belongs to the restricted system, then so does b.

This is proved by induction on the definition of a red b, using (1) and (2).

INDUCTIVE DEFINITION 4 of a type v *mr* A (v *modified realizes* A) for types A in the restricted system, where v is a variable of a finite type determined by A, but not occurring in A.

1. v^{o} mr o $\quad := v^{o} = v^{o}$

2. v^o mr a=b := a=b
3. v mr $\forall x^A B[x^A]$:= $\forall u$(u mr A $\rightarrow$ v(u) mr B[u])
4. v mr $\exists x^A B[x^A]$:= $j_1 v$ mr A $\wedge$ $j_2 v$ mr $B[j_1 v]$

Remarks. If σ is a finite type, v mr σ is defined by 1,3, and 4 such that $v \vdash \sigma$. Moreover, it is easy to construct a term $r^*[v^\sigma] \vdash v^\sigma \mathrm{mr}\sigma$. So, v mr σ is just a complicated way of expressing that v is of type σ. Hence, 3 and 4 of this definition are coherent, since x^A occurs in B only if A is a finite type in which case u mr A and $j_1 v$ mr A imply that B[u] and $B[j_1 v]$ are types (in the restricted system).

Substitutions in types b mr B are executed piecewise, i.e.
(3) (b mr B) [a] $\equiv$ b[a] mr B[a].

Definition 4 is equivalent to the one given in Troelstra 1973, ch. III, § 4, but not identical with it. In the presence of pairing types $\sigma \wedge \tau$, we use single terms instead of tuples as realizing terms. This leads to the superfluous v^o which, instead of the empty tuple, modified realizes equations. Also, v mr $\forall x^\sigma B$ (for finite type σ) contains a redundant u mr σ following $\forall u^\sigma$. Our formulation, however, gives formal expression to the similarity between the $\forall x^\sigma$-case and the $\rightarrow$-case of mr, and in v^σ mr A by the above definition, the finite type σ reflects the logical structure of A more closely. For $\exists$-free types A, we can construct a closed $\exists$-free term
(4) $s_A \vdash \forall v(A \leftrightarrow v \text{ mr } A)$,
whereas Troelstra has A $\equiv$ v mr A.

THEOREM 1, *soundness theorem for modified realization.* If $a \vdash A$ in the restricted system, then we can construct a PRT a' and an $\exists$-free term

$a^* \vdash a'$ mr A

as follows: Only if y^B occurs in a, the variable y' may occur in a', and y' and $y^* \vdash y'$ mr B may occur in a^*.

The *proof* is standard. We construct a' and a^* by induction on terms and leave it to the reader to verify the theorem. With each variable x^A we uniquely associate a variable x' of appropriate finite type and a variable $x^* \vdash x'$ mr A. If A is a finite type, we take x' to be x. The construction now shows at the same time:

(5) If a is a PRT, then a' is a;

(6) If y, a satisfy the c.o.v. then so do y', a' and y', y^*, a^*.

$(\forall I)$ $(\lambda x^A.b)' := \lambda x'.b'$ and $(\lambda x^A.b)^* := \lambda x'.\lambda x^*.b^*$.

$(\forall E)$ $c(a)' := c'(a')$ and $c(a)^* := c^*(a')(a^*)$.

$(\exists I)$ $\langle a,b\rangle' := \langle a',b'\rangle$ and $\langle a,b\rangle^* := \langle a^*,b^*\rangle$.

$(\exists E)$ $d[j_1c,j_2c]' := d'[j_1c',j_2c']$

and $d[j_1c,j_2c]^* := d^*[j_1c',j_1c^*,j_2c',j_2c^*]$.

(oI) $0' := 0$ and $0^* := r(0)$,

$suc(t)' := suc(t')$ and $suc(t)^* := r(suc(t'))$.

(oE) Writing $f[t] := R(a',\lambda xy'.b',t)$,

we have $a^* \vdash f[0]$ mr $F[0]$ and

$b^*[x,r(x),f[x],y^{*f[x]\text{mr } F[x]}] \vdash f[suc(x)]$ mr $F[suc(x)]$.

So we put $R(a,\lambda xy.b,t)' := f[t]$ and

$R(a,\lambda xy.b,t)^* := R(a^*,\lambda xy^*.b^*[x,r(x),f[x],y^{*f[x]\text{mr}F[x]}],t)$.

$(=I)$ $r(a)' := 0$ and $r(a)^* := r(a)$

$(=E)$ $eq(c,d,a,b)' := c'$ and $eq(c,d,a,b)^* := eq(c^*,d^*,a,b)$.

Remarks. The construction of a' is faithful except for =-rules; the construction of a^* is faithful only in the case of $\exists$- and =-rules. If in $(\forall E)$, A is a finite type, we have

$c^*(a')(a^*) \vdash c'(a')$ mr $B[a']$.

Since a, however, is a PRT, a' is identical with a, and this is already the desired conclusion. Without this consideration, we might also use the standard argument.

$c^*(a)(r^*(a)) \vdash c'(a)$ mr $B[a]$

for an alternative choice for $c(a)^*$, but then the homomorphism * would be still less faithful and less compatible with reduc-

tions. Similar arguments apply to ($\exists$I) and (oE). The same problem prevents the proof of mr-soundness of the strong $\exists$-elimination rule:
The induction hypothesis yields

$$j_2c^* \vdash j_2c' \text{ mr } B[j_1c'],$$

and though j_1c and j_1c' have the same finite type, there is no way to identify them as long as c and c' depend on variables y,y' of different types. We therefore restrict ourselves to the case in which the type D of a deduction continuing j_2c becomes independent of j_1c, and that is exactly the case covered by the restricted, weak $\exists$-elimination rule of defn 3.

Theorem 1 has the usual

COROLLARY. The restricted system is conservative over its $\exists$-free fragment.

Proof. If A is $\exists$-free and $a \vdash A$ is closed, we have an $\exists$-free closed term $a^* \vdash a'$ mr A, and from (4) above, the term $j_2(s_A(a'))(a^*) \vdash A$ is $\exists$-free and closed.

This corollary would be more interesting for the extended system. We now compare reductions and the maps ' and *.

LEMMA 1. If a red b then a' red b' and a^* red b^*.

This can be read off directly from the construction of a' and a^*.

LEMMA 2. If a is normal then a' and a^* are normal.

Proof by induction on a. If a is constructed by an I-rule, by (oE) or by (=E), then a' and a^* are normal by induction hypothesis or immediately.

A term c(a) is normal, iff c and a are normal and c is not constructed by ($\forall$I). Let c(a) be normal. By induction hypothesis, c',a',c^*, and a^* are normal, and c',c^* are also not constructed by ($\forall$I), since in all cases except ($\forall$I), c',c^* start with a symbol different from λ, even in the case of (=E). In this case, the type of c is an equation by the restriction of defn 3, hence $c' \vdash o$ and is therefore not constructed by an I-rule. So, $c(a)'$ and $c(a)^*$ are normal.

A term $e := d[j_1c, j_2c]$ is normal, iff c,d are normal and c is not constructed by ($\exists$I). As above, we see that then e' and e^* are normal, too.

We denote by a^N the *unique normal form* of a term a (cf. Martin-Löf 1973, § 3). By the closure lemma, a^N is a term of the restricted system if a is.

THEOREM 2. The process of normalization commutes with the maps $'$ and *:

$$a^{N'} \equiv a^{'N} \quad \text{and} \quad a^{N*} \equiv a^{*N}$$

Proof. Given a, the term a^N also belongs to the restricted system, and a red a^N; hence a' red $a^{N'}$ and a^* red a^{N*} by lemma 1. Furthermore, a^N is normal; hence $a^{N'}$ and a^{N*} are normal by lemma 2. Therefore, $a^{N'}$ is the normal form of a', and a^{N*} is the normal form of a^*, q.e.d.

In total, $'$ and $*$ are homomorphisms also with respect to reduction (by lemma 1) and to normalization (by theorem 2). More precisely, any reduction step is transformed by $'$ as well as by $*$ into zero, one, or two reduction steps of the same kind. Only (=red) is transformed into zero steps (i.e. is omitted) by $'$. This gives upper and lower bounds for the length of reduction chains of a' and a^* in terms of the length of reduction chains of a.

COROLLARY (Minc 1974). Let $c \vdash \exists x^\sigma B[x^\sigma]$ be a closed term, σ a finite type. Then c^N is a pair $\langle a,b \rangle$ such that $b \vdash B[a]$ and a, a closed PRT, is the normal form of j_1c'.

Proof. If $c \vdash \exists x^\sigma B[x^\sigma]$ is a closed term, then c^N is a closed normal term constructed by an I-rule and of the same type as c. Hence, c^N is a pair $\langle a,b \rangle$ with normal $a \vdash \sigma$ and $b \vdash B[a]$. By the closure lemma, a is a PRT, and therefore a' is identical with a by (5). By theorem 2, c' red $c^{N'} \equiv \langle a,b' \rangle$, hence j_1c' red a, and since a is normal, $j_1 c'^N \equiv a$.

In this corollary, only the homomorphism ' is used. We do not know of an equally convincing application of the homomorphism *.

The last results, from lemma 1 on, remain unchanged if ξ-reduction is included in definitions 1 and 3.

In a less stream-lined natural deduction context, these results were presented at Oberwolfach in 1975. Stein 1976 and Rath 1978 have proved similar results for an infinity of other functional interpretations, like Gödel's 1958 Dialectica interpretation and the Λ-interpretation of Diller-Nahm 1974. The methods are related, but more sophisticated. The corresponding transformations ' and * do no longer have the pleasant properties they enjoy here. In particular, they are no longer straight forward homomorphisms with respect to reduction. Within a class of "first order" functional interpretations, optimal properties of ' and * as described above, seem to characterize modified realization.

REFERENCES

Curry, H.B. and Feys, R. 1958: Combinatory logic I. 2nd ed. 1968. North-Holland, Amsterdam.

Diller, J., and W. Nahm 1974: Eine Variante zur Dialectica-Interpretation der Heyting-Arithmetik endlicher Typen. *Arch. math. Logik* 16, 49 - 66.

Girard, J.-Y., 1971: Une extension de l'interpretation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types, *in*: Proc. 2nd Scandinavian Logic Symp., North-Holland, Amsterdam, 63 - 92.

Gödel, K., 1958: Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica* 12, 280 - 287.

Howard, W.A., 1969: The formulae-as-types notion of constructions. Manuscript, to appear in this Festschrift.

Kreisel, G., 1959: Interpretation of analysis by means of constructive functionals of finite type, *in*: A. Heyting (ed.), Constructivity in mathematics, North-Holland, Amsterdam, 101 - 128.

Martin-Löf, 1975: An intuitionistic theory of types, predicative part, *in*: Logic colloquium '73, North-Holland, Amsterdam, 73 - 118.

Minc, G., 1974: On E-theorems (Russian), *in*: Investigation in constructive mathematics VI. Zapisky Nauk. Sem. Leningrad, Steklov Inst. 40, 110 - 118.

Pohlers, W., 1973: Ein starker Normalisationssatz für die intuitionistische Typentheorie. *Manuscripta Mathematica* 8, 371 - 387.

Prawitz, D., 1965: Natural deduction. Almqvist and Wiksell, Stockholm, Göteborg, Uppsala.

Rath, P., 1978: Eine verallgemeinerte Funktionalinterpretation der Heyting-Arithmetik endlicher Typen. Dissertation Münster.

Stein, M., 1976: Interpretationen der Heyting-Arithmetik endlicher Typen. Dissertation Münster.

Tait, W.W., 1972: Unpublished lecture, Munich.

Troelstra, A.S., 1973: Metamathematical investigation of intuitionistic arithmetic and analysis. Lecture Notes in Mathematics 344, Springer, Berlin-Heidelberg-New York.

A POSITIVE LOGIC PROOF PROCEDURE

David Meredith

Old Blood Road, Merrimack, NH 03054

United States of America

Dedicated to H.B. Curry on the occasion of his 80th Birthday

This paper presents a procedure which has application in both propositional and combinatory logic. In the former it provides a proof procedure, for the Hilbert Positive Implicational Logic, that involves no reduction to canonical form, and that gives proofs which are very concise. In the latter it provides a method for finding, for any functional type isomorphic with a Positive Logic thesis, a combinator of that type.

Section I gives some needed definitions. Section II describes and illustrates the procedure, and Section III gives a proof of its sufficiency. The final section deals with the application to functional types. Save for the proof of one theorem , the first three sections are independent of combinatory logic.

I. PRELIMINARIES

Basic to the procedure is C.A. Meredith's condensed detachment operator $\underline{D}$. For propositional formulae ϕ and ψ, $\underline{D}\phi\psi$ is the strongest formula that can be obtained by applying Modus Ponens with ϕ, or some substitution in it, as major premiss, and ψ, or some substitution in it, as minor premiss. Thus

(1) $\underline{D}$ CCpCqrCCpqCpr CpCqp = CCpqCpp

while

(2) $\underline{D}$ CCqrCCpqCpr CCpCpqCpq = CCrCpCpqCrCpq

and

(3) $\underline{D}$ CCpCpqCpq CCqrCCpqCpr = CCppCpp

(There are of course cases where $\underline{D}\phi\psi=\Lambda$, but these are not germane to present interest.)

DEFINITION 1. An *application expression* is defined inductively in two steps.

I. Any propositional formula in implication is an application expression.

II. Let α and β be application expressions: then $(\alpha\beta)$ is an application expression.

To avoid unnecessary parentheses in application expressions, left associating parentheses are omitted.

DEFINITION 2. The *result* of an application expression is defined inductively in two steps.

I. A propositional formula in implication is its own result.

II. Let there be application expressions α and β, and formulae ϕ and ψ, such that $C\phi\psi$ is the result of α, and ϕ is the result of β: then ψ is the result of $\alpha\beta$.

Obviously, not every application expression has a result. Thus while the result of

(4) CpCqr p (Cpq p)

is r,

(5) CCpCqrCCpqCpr CpCqp

has no result.

Four *operators* on application expressions correspond to four axiom schemata as follows.

$$B \simeq CC\nu\pi CC\mu\nu C\mu\pi$$

$$C \simeq CC\mu C\nu\pi C\nu C\mu\pi$$

$$W \simeq CC\mu C\mu\nu C\mu\nu$$

$$K \simeq C\mu C\nu\mu$$

In the proof procedure these operators are governed by the following rules. α, β and γ may be operators, or application expressions having results.

RULE B. $\alpha(\beta\gamma) = B\alpha\beta\gamma$

RULE C. $\alpha\beta\gamma = C\alpha\gamma\beta$

RULE W. Where if β is an operator both instances of it are identical

$$\alpha\beta\beta = W\alpha\beta$$

RULE K. $\alpha = K\alpha\beta$

In addition to the four primitive operators it is convenient to have one defined operator.

DEFINITION 3. $I =_{df} WK$

This operator is governed by the following rule.

RULE I. $\alpha = I\alpha$

The rule holds since $\alpha = K\alpha\alpha = WK\alpha$.

II. PROOF PROCEDURE

For any thesis of Positive Implicational Logic, $C\phi_1 \,..C\phi_n\nu$ where ν is elementary, an application expression can be found whose result is ν and whose every component formula is either one of the ϕ_i or a Positive Logic thesis. (For a proof of this see Meredith (1978).) Finding such an application expression for a given thesis is not difficult: one simply works back from the elementary consequent, seeing what is required to obtain this from the antecedents. Clearly, for instance, in the case of CCpCqrCCspCqCsr, since the final r is consequent in CpCqr and p is given by Csp s, the required application expression is

(6) CpCqr (Csp s) q

and in the case of CCCpqrCsCCqCrtCqt it is

(7) CqCrt q (CCpqr (CqCpq q))

In the last instance of course one of the component formulae in the application expression -- CqCpq -- is a Positive Logic thesis and not one of the antecedents of the elementary consequent.

Given a Positive Logic thesis $C\phi_1 \,..C\phi_n\nu$ and an associated application expression, an axiomatic derivation of the thesis can be found simply by using Rules B through I to construct from the application expression an expression $\Pi\phi_1 \,..\phi_n$ where Π

contains only propositional operators and parentheses. In the event that the application expression includes a thesis distinct from the antecedents, this is replaced by an arbitrary temporary operator Δ: when the expression $\Pi\phi_1 \ldots \phi_n$ is arrived at, every Δ in Π and every occurrence of I is then replaced by its definition in terms of the primitive operators. If the binary juxtaposition in Π is now interpreted as the detachment operator $\underline{D}$, Π represents a $\underline{D}$-derivation from the axioms corresponding to the operators, either of $C\phi_1 \ldots C\phi_n\nu$ or of a thesis of which this is a substitution instance.

For an illustration of the procedure consider the thesis CCCCpqqrCpr and the associated application expression

(8) CCCpqqr (CpCCpqq p)

Here CpCCpqq is a thesis not among the antecedents of the elementary consequent in CCCCpqqrCpr. Starting with the replacement of this thesis by X, the successive steps of the procedure are as follows.

(9) CCCpqqr (X p)
B CCCpqqr X p
C B X CCCpqqr p

Applying the same procedure to the application expression associated with CpCCpqq gives

(10) Cpq p
I Cpq p
C I p Cpq

which with the replacement of I by WK allows the replacement of X by C(WK) to give CB(C(WK)) as the derivation of the original thesis. That CB(C(WK)) does indeed represent such a derivation is easily established.

1. CCqrCCpqCpr
2. CCpCqrCqCpr
3. CCpCpqCpq

4. CpCqp

D21 = 5 CB

5. CCpqCCqrCpr

D34 = 6 WK

6. Cpp

D26 = 7 C(WK)

7. CpCCpqq

D57 = 8 CB(C(WK))

8. CCCCpqqrCpr

The restriction on Rule W -- that if β is a propositional operator both instances of it must be identical -- is important to the procedure. This can be seen by considering the thesis CCpCpCpqCpq and the associated application expression

(11) CpCpCpq p p p

From this application expression the procedure might yield

(12) W CpCpCpq p p
W (W CpCpCpq) p
B W W CpCpCpq p

Application of Rule W at this point is impossible. The two instances of W in the last line of (12) are not identical -- the one on the left is CCpCpqCpq while the other is CCpCpCpqCpCpq.

It is worth pointing out also that the general stipulation for all the rules that if α, β, γ be application expressions they have results, is pertinent even to the seemingly irrelevant β in Rule K. The application expression

(13) p

associated with the thesis CpCqCrp, for instance, cannot be transformed into K p (q r) since (q r) has no result. To obtain a derivation for the thesis from (13) the correct steps are

(14) K p q
K (K p q) r

B K (K p) q r

B (B K) K p q r

III. SUFFICIENCY OF THE PROOF PROCEDURE

THEOREM 1. Where B* is a substitution instance of CCqrCCpqCpr. If $\alpha(\beta\gamma)$ is an application expression having a result, then there is an application expression $B^*\alpha\beta\gamma$ having the same result.

Proof. Let ϕ be the result of $\alpha(\beta\gamma)$, and let α^*, β^*, γ^* be the results of α, β, γ respectively. Then for some ψ, $\beta = C\gamma^*\psi$ and $\alpha^* = C\psi\phi$. But $CC\psi\phi CC\gamma^*\psi C\gamma^*\phi$ is a substitution instance of CCqrCCpqCpr, and the theorem follows.

Proofs of the next three theorems are similar to that of Theorem 1.

THEOREM 2. Where C* is a substitution instance of CCpCqrCqCpr. If $\alpha\beta\gamma$ is an application expression having a result, then there is an application expression $C^*\alpha\gamma\beta$ having the same result.

THEOREM 3. Where W* is a substitution instance of CCpCpqCpq. If $\alpha\beta\beta$ is an application expression having a result, then there is an application expression $W^*\alpha\beta$ having the same result.

THEOREM 4. Where K* is a substitution instance of CpCqp. If α is an application expression having a result, then there is an application expression $K^*\alpha\beta$ having the same result.

THEOREM 5. Let $\Pi\phi_1 .. \phi_n$ be an application expression whose result is ν, and let every element in Π be a substitution instance of one of the Hilbert axioms CCqrCCpqCpr, CCpCqrCqCpr, CCpCpqCpq, CpCqp. Then both of the following are true.

a. Π is an application expression whose result is $C\phi_1 .. C\phi_n\nu$

b. The $\underline{D}$-derivation Π^D that results when each of the Hilbert axioms replaces its substitution instances in Π and the binary juxtaposition is interpreted as the condensed detachment operator $\underline{D}$, is a $\underline{D}$-derivation either of $C\phi_1 .. C\phi_n\nu$ or of a thesis of which this is a substitution instance.

Proof. Part (a) follows immediately from the hypothesis. Part (b) follows from the hypothesis and the rule governing D̲.

THEOREM 6. For any thesis of Positive Implicational Logic $C\phi_1 \ldots C\phi_n\nu$ where ν is elementary, the procedure described in Section II will yield a derivation from the Hilbert axioms.

Proof. As noted earlier, there must be an application expression associated with $C\phi_1 \ldots C\phi_n\nu$ whose result is ν and whose every component formula is either one of the ϕ_i or a thesis of Positive Logic. From Theorems 1 through 5 it is clear that any D̲-derivation resulting from the use of the procedure on the application expression will be a derivation from the Hilbert axioms either of $C\phi_1 \ldots C\phi_n\nu$ or of a thesis of which this is a substitution instance. From the considerations urged below it is clear that the procedure will always be effective in obtaining such a derivation. The theorem follows.

It is known from combinatory logic that any arbitrary combination of elements from $\{\kappa_1 \ldots \kappa_m\}$ can be expressed as a function $\Phi\kappa_1 \ldots \kappa_m$ where Φ is constructed from four combinators whose defining rules -- save for the general stipulation on α, β, γ, and the restriction on Rule W -- are identical to the rules for the propositional operators (see Curry and Feys (1958) Chapter 5). The only propositional operator limited by the general stipulation is K, but since $K\alpha(\beta_1 \ldots \beta_k)$ can be expressed as $K(\ldots K(K\alpha\beta_1)\ldots\beta_{k-1})\beta_k$ this limitation does not weaken the procedure. Nor does the restriction on Rule W weaken the procedure, since for a propositional operator Σ, $\alpha\Sigma\Sigma$ need never be taken into $W\alpha\Sigma$.

IV. APPLICATION TO FUNCTIONAL TYPES

It is clear that for any functional type isomorphic with a Positive Logic thesis, the procedure described in Section II will yield a combinator in B, C, W and K, which has that type.

To find a combinator in the more usual two primitive base, the first three operators must be replaced by

$$S \simeq CC\pi C\mu\nu CC\pi\mu C\pi\nu$$

which is governed by

RULE S. Where if γ is an operator both instances of it are identical

$$\alpha\gamma(\beta\gamma) = S\alpha\beta\gamma$$

and in place of Definition 3 one must use

DEFINITION 4. $I =_{df} SKK$

The restriction on the repeated term in Rules W and S, is similar to a condition of Curry and Fey's Subject-Expansion Theorem (Curry and Feys (1958) Theorem 9C3 clause (d)). This suggests the following analogue of a subject-conversion theorem.

THEOREM 7. Where α and β are application expressions, and $\Psi(\ )$ is a context such that $\Psi(\alpha)$ and $\Psi(\beta)$ are application expressions. If α and β have the same result, then $\Psi(\alpha)$ and $\Psi(\beta)$ have the same result.

Proof. The result of $\Psi(\alpha)$ depends only on the results of its component application expressions. The interchange of α and β, therefore, will leave the result of the whole unchanged.

REFERENCES

Curry, H. B., and Feys, R. (1958). "Combinatory Logic Volume I", North-Holland Publishing Co., Amsterdam.

Meredith, D. (1978). Positive Logic and λ-Constants. *Studia Logica* 37, 269-285.

ON THE EXISTENCE OF CLOSED TERMS IN THE TYPED λ-CALCULUS I

Richard Statman

Department of Mathematics, Rutgers University, New Brunswick, New Jersey

Dedicated to H.B. Curry on the occasion of his 80th Birthday

INTRODUCTION

In Statman (1979b) we showed that the problem of determining if there is a closed term of type τ (over a single ground type) can be solved in linear time (over finitely many ground types in polynomial time, and over infinitely many ground types complete in polynomial space). In this note we consider the problem of determining if there is a closed term of type τ (over a single ground type) which acts w.r.t. $\beta\eta$ conversion as a one to one resp. onto function on closed terms.

More precisely, we say σ is *reducible* to τ if there exists $M \in \sigma \to \tau$ such that for all $N_1, N_2 \in \sigma$, $N_1 =_{\beta\eta} N_2 \iff MN_1 =_{\beta\eta} MN_2$. We say σ *enumerates* τ if there exists $M \in \sigma \to \tau$ such that for all $N_1 \in \tau$ there exists $N_2 \in \sigma$ satisfying $N_1 =_{\beta\eta} MN_2$.

We shall give the complete structure of the reducibility relation save one bit of information. In particular, the equivalence classes are well ordered in type $\omega + 2$ or $\omega + 3$. Our proof gives a linear time algorithm for determining if one type is reducible to another. Much less is known about enumeration. We will show that each type containing a closed term enumerates only finitely many others but linear ordering fails. In another paper we shall show that enumeration is an extension of a polynomial time decidable well quasi ordering.

We shall also give an application of our results to the problem of unification. In particular, any non trivial unification problem of cumulative degree ≥ 9 is undecidable.

PRELIMINARIES

We adopt for the most part the notation of Statman (1979c).

Lower case Greek letters, save λ, denote types built up from 0 by $\rightarrow$. The number of occurrences in τ of 0 is $|\tau|$. If $\tau = \tau(1) \rightarrow (\ldots(\tau(t) \rightarrow 0)\ldots)$ the $\tau(i)$ are the *components of* τ. The number of components of a type is denoted by the corresponding lower case Roman letter. If $n_1 \ldots n_m$ is a sequence of positive integers we write $\tau(n_1 \ldots n_m)$ for $(\ldots(\tau(n_1))\ldots)(n_m)$. This will be undefined if for some $i \in [1,m]$ $t(n_1 \ldots n_{i-1}) < n_i$. τ is *tiny* if it has no positive occurrence of a subtype with at least two components, *small* if it has no negative occurrence of a sub-type with at least two components and *large* if it is not small. If for each τ, Γ_τ is a permutation of $[1,t]$ we define recursively $\Gamma(\tau) = \Gamma(\tau(\Gamma_\tau(1))) \rightarrow (\ldots(\Gamma(\tau(\Gamma_\tau(t))) \rightarrow 0)\ldots)$. If Ψ is any ordinal valued function on types, τ is Ψ *righteous* if the components of τ are in non increasing order of Ψ-value from left to right.

Lower case Roman letters denote variables and upper case Roman letters terms. The letters A through N always denote closed terms. If T has type τ we write $T \in \tau$. If $Z \in 0$, $\Lambda\ \tau\ Z$ is the term of type τ which results from Z by prefixing to Z a dummy λ for each component of τ.[1]

PERMUTATION LEMMA. For each Γ and τ there is an $M_\tau^\Gamma \in \Gamma(\tau) \rightarrow \tau$ and an $N_\tau^\Gamma \in \tau \rightarrow \Gamma(\tau)$ such that for $x \in \tau$, $M_\tau^\Gamma(N_\tau^\Gamma x) \underset{\beta\eta}{=} x$ and for $y \in \Gamma(\tau)$, $N_\tau^\Gamma(M_\tau^\Gamma y) \underset{\beta\eta}{=} y$.

Proof. Define M_τ^Γ and N_τ^Γ simultaneously as follows. $M_0^\Gamma = N_0^\Gamma = \lambda xx$ for $x \in 0$. If $\tau \neq 0$ select $x \in \tau$, $y \in \Gamma(\tau)$ for $1 \leq i \leq t$, $x_i \in \Gamma(\tau(i))$ and for $1 \leq j \leq t$, $y_j \in \tau(\Gamma_\tau(j))$; then

$$M_\tau^\Gamma = \lambda y \lambda y_{\Gamma_\tau^{-1}(1)} \ldots y_{\Gamma_\tau^{-1}(t)} . y(N^\Gamma_{\tau(\Gamma_\tau(1))} y_1) \ldots (N^\Gamma_{\tau(\Gamma_\tau(t))} y_t)$$

and

$$N_\tau^\Gamma = \lambda x \lambda x_{\Gamma_\tau(1)} \ldots x_{\Gamma_\tau(t)} . x(M_{\tau(1)}^\Gamma x_1) \ldots (M_{\tau(t)}^\Gamma x_t).$$

The lemma follows easily by induction. End of proof.

The significance of the permutation lemma is that if Ψ is Γ invariant for all Γ, that is $\Psi(\Gamma(\tau)) = \Psi(\tau)$, we can always without loss of generality assume τ is Ψ righteous.

For the following see Statman (1979c). If A is a structure and $\theta \subseteq A$ with $\theta \cap A^0 \neq \emptyset$ then the structure $(\Lambda_A(\theta))^E/\approx$ is called here the *Gandy hull* of θ in A. We shall write $[\![M]\!]$ for $\mathrm{Val}_A(M)$ when no confusion can result.

REDUCIBILITY

Two types are *equivalent* if each is reducible to the other. We shall consider the partial ordering of reducibility among equivalence classes. This partial ordering has a least element consisting of the types with no closed terms. For the remainder of this section we shall consider only types containing closed terms.

LEMMA 1. Each type is reducible to a type of rank ≤ 3.

Proof. Define $\check{0} = \hat{0} = 0$, $\check{\tau} = \underbrace{0 \to (\ldots(0}_{t} \to 0)\ldots)$ and $\hat{\tau} = 0 \to (\tau\check{(}1) \to (\ldots(\tau\check{(}t) \to 0)\ldots)$. For each τ let $u_\tau \in \hat{\tau}$ and $x, x_1, \ldots \in 0$. Define simultaneously $V_\tau(x) \in \tau$ and $U_\tau \in \tau \to \check{\tau}$ as follows.

$$V_0(x) = x, \quad U_0 = \lambda x V_0(x).$$

If $\tau \neq 0$ select $z \in \tau$ and for $1 \leq i \leq t$, $y_i \in \tau(i)$; then

$$V_\tau(x) = \lambda y_1 \ldots y_t (u_\tau x)(U_{\tau(1)} y_1) \ldots (U_{\tau(t)} y_t),$$

$$U_\tau = \lambda z \lambda x_1 \ldots x_t \; z V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t).$$

Let $M = \lambda y_1 \ldots y_t . \; y_i(M_1 y_1 \ldots y_t) \ldots (M_{t(i)} y_1 \ldots y_t) \in \tau$ be in Φ normal form. We have

$$U_\tau M \underset{\beta\eta}{=} \lambda x_1 \ldots x_t V_{\tau(i)}(x_i)(M_1 V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t)) \ldots (M_{t(i)} V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t))$$

$$=_{\beta\eta} \lambda x_1 \ldots x_t (u_{\tau(i)} x_i)(U_{\tau(i1)}(M_1 V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t))) \ldots (U_{\tau(it(i))}(M_{t(i)} V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t))$$

For $1 \leq j \leq t(i)$ let $\sigma_j = \tau(1) \to (\ldots(\tau(t) \to \tau(ij))\ldots)$, and we have

$$U_{\tau(ij)}(M_j V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t)) =_{\beta\eta}$$
$$\lambda x_{t+1} \ldots x_{t+s_j} M_j V_{\tau(1)}(x_1) \ldots V_{\tau(t)}(x_t) V_{\sigma_j(1)}(x_{t+1}) \ldots V_{\sigma_j(s_j)}(x_{t+s_j})$$
$$=_{\beta\eta} (U_{\sigma_j} M_j) x_1 \ldots x_t.$$

Thus $U_\lambda M =_{\beta\eta}$

$$\lambda x_1 \ldots x_t (u_{\tau(i)} x_i)((U_{\sigma_1} M_1) x_1 \ldots x_t) \ldots ((U_{\sigma_{t(i)}} M_{t(i)}) x_1 \ldots x_t).$$

It follows easily by induction on the length of M that for Φ normal $N \neq M$ of type τ, $U_\tau M \neq_{\beta\eta} U_\tau N$.

The lemma follows directly. End of proof.

LEMMA 2. Each type is reducible to a type of rank ≤ 3 all of whose components of rank 2 are tiny.

Proof. It suffices to prove the lemma for rank righteous types. The proof is similar to the proof of Lemma 1. We illustrate with $((0 \to 0) \to ((0 \to 0) \to 0)) \to 0$ which is reducible to $((0 \to 0) \to 0) \to ((0 \to (0 \to 0)) \to 0)$ by means of $\lambda u \lambda xy\ u(\lambda z_1 z_2 y(xz_1)(xz_2))$ where $z_1, z_2 \in 0 \to 0$, $y \in 0 \to (0 \to 0)$, $x \in (0 \to 0) \to 0$ and $u \in ((0 \to 0) \to ((0 \to 0) \to 0)) \to 0)$. End of proof.

LEMMA 3. Each type is reducible to a type of rank ≤ 3 all of whose components of rank 2 are $(0 \to 0) \to 0$.

Proof. The proof is similar to the proof of Lemma 1. We illustrate with $((0 \to (0 \to 0)) \to 0) \to 0$ which is reducible to $(0 \to (0 \to 0)) \to ((0 \to 0) \to ((0 \to 0) \to 0))$ by means of

$$\lambda u \lambda x \lambda y_1 y_2\ u(\lambda z\ x(\lambda w\ z(y_1 w)(y_2 w)))$$

where $w \in 0$, y_1, $y_2 \in 0 \to 0$, $z \in 0 \to (0 \to 0)$, $x \in (0 \to 0) \to 0$ and $u \in ((0 \to (0 \to 0)) \to 0) \to 0$. End of proof.

LEMMA 4. Each type is reducible to a type of the form

$((0 \to 0) \to 0) \to \tau$ for τ of rank ≤ 2.

Proof. Again it suffices to prove the lemma for rank righteous types, and again the proof is like the proof of Lemma 1. We illustrate with $((0 \to 0) \to 0) \to (((0 \to 0) \to 0) \to 0)$ which is reducible to $((0 \to 0) \to 0) \to ((0 \to 0) \to ((0 \to 0) \to 0))$ by means of $\lambda u \lambda x \lambda y_1 y_2\, u(\lambda z_1\, y_1(xz_1))\, (\lambda z_2\, y_2(xz_2))$ where $y_1, y_2, z_1, z_2 \in 0 \to 0$, $x \in (0 \to 0) \to 0$ and $u \in ((0 \to 0) \to 0) \to (((0 \to 0) \to 0) \to 0)$. End of proof.

LEMMA 5. If τ has rank ≤ 2 then $((0 \to 0) \to 0) \to \tau$ is reducible to a type of rank ≤ 2.

Proof. The proof is not like the proof of Lemma 1. Let $z \in 0$, $y, v \in 0 \to 0$, $x \in 0 \to 0$, and $u \in (0 \to 0) \to 0$. Let $T, T_1, \ldots \in 0$ be in long $\beta\eta$ normal form not containing x, y or z and with u as the only free variable of rank ≥ 1. Let $U = \lambda v\, x(v(y(vz)))$ and put T^+ = the long $\beta\eta$ normal form of $[U/u]T$ and T^* = the long $\beta\eta$ normal form of $[\lambda wz/y](T^+)$ for $z \neq w \in 0$.

CLAIM 1. Let $y(T')$ be a maximal subterm of T^+ beginning with y, then T'^* is a proper subterm of T^*.

Proof. By induction on the length of T. First note that T'^* = the long $\beta\eta$ normal form of $[\lambda wz/y]T'$. The basis case, when T is a variable, is vacuous. Suppose T is an application.

Case 1: $T = vT_1 \ldots T_n$ for $v \in \underbrace{0 \to (\ldots(0}_{n} \to 0)\ldots)$. We have $T^+ = vT_1^+ \ldots T_n^+$ and $T^* = vT_1^* \ldots T_n^*$, so the case follows from the induction hypothesis applied to $T_1 \ldots T_n$.

Case 2: $T = u(\lambda wT_1)$ for $w \in 0$. Now $T^+ =_{\beta\eta} x((\lambda wT_1^+)\ (y([z/w]T_1^+)))$ and $T^* =_{\beta\eta} x([z/w]T_1^*)$. Suppose $y(T')$ is a maximal subterm of T^+ beginning with y, then either $T' = [z/w]T_1^+$ or there is a maximal subterm of $[z/w]T_1^+$ beginning with y, say $y(T'')$, with $T''^* \equiv T'^*$. In the first case $T'^* = [z/w]T_1^*$ which is a proper subterm of T^*. In the second case T'^* is a proper subterm of T^* by the induction hypothesis applied to $[z/w]T_1$. This proves claim 1.

CLAIM 2. $T_1^+ = T_2^+ \Rightarrow T_1 \equiv T_2$.

Proof. By induction on the length of T_1. The basis case, when

T_1 is a variable, can be seen by inspection. Suppose T_1 is an application.

Case 1: $T_1 = u(\lambda w T_3)$ for $w \in 0$. This case is obvious unless $T_2 = u(\lambda w T_4)$. We have $T_1^+ = x((\lambda w T_3^+)(y([z/w]T_3^+)))$ and $T_2^+ = x((\lambda w T_4^+)(y([z/w]T_4^+)))$. Now if $T_1 \neq T_2$, then $T_3 \neq T_4$, so by induction hypothesis $T_3^+ \underset{\beta\eta}{\neq} T_4^+$. By the first claim, each maximal $y(T')$ in $[z/w]T_i^+$ has T'^* a proper subterm of $[z/w]T_i^*$ where $3 \leq i \leq 4$. Thus for $3 \leq i \leq 4$, the maximal $y(T')$ in $[y([z/w]T_i^+)/w]T_i^+$ with length T'^* = length T_i^* = length $([y([z/w]T_i^+))/w]T_i^+)^*$ are just the maximal $y(T') = y([z/w]T_i^+)$ which result from the substitution for w. Hence

$$[y([z/w]T_3^+)/w]T_3^+ = [y([z/w]T_4^+)/w]T_4^+ \Rightarrow T_3^+ = T_4^+.$$

So $T_1^+ \neq T_2^+$ if $T_1 \neq T_2$.[2]

Now let

$$\sigma = ((0 \to 0) \to 0) \to \tau$$

and

$$\rho = (0 \to 0) \to ((0 \to 0) \to (0 \to \tau)).$$

Select $v \in \sigma$ and set $M_\tau = \lambda v \lambda xyz(vU) \in \sigma \to \rho$. By the second claim if $N_1, N_2 \in \sigma$ are in long $\beta\eta$ normal form then $M_\tau N_1 \underset{\beta\eta}{=} M_\tau N_2 \Rightarrow N_1 = N_2$. End of proof.

LEMMA 6. Each type of rank ≤ 2 is reducible to $(0 \to (0 \to 0)) \to (0 \to 0)$.

Proof. The proof is a simple coding of labeled trees by unlabeled binary trees. We illustrate with

$$(0 \to (0 \to 0)) \to ((0 \to (0 \to 0)) \to (0 \to 0)).$$

The desired term is

$$\lambda u \lambda xy\ u(\lambda z_1 z_2\ xy(xz_1z_2))(\lambda z_1 z_2\ x(xyy)(xz_1z_2))y$$

where $z_1, z_2, y \in 0$, $x \in 0 \to (0 \to 0)$,

$$u \in (0 \to (0 \to 0)) \to ((0 \to (0 \to 0)) \to (0 \to 0)).$$

End of proof.

Summarizing Lemmas 1–6:

PROPOSITION 1. Each type is reducible to $(0 \to (0 \to 0)) \to (0 \to 0)$.

LEMMA 7. $(0 \to (0 \to 0)) \to (0 \to 0)$ is reducible to each large type.

Proof. Define U_τ for τ large and possibly not containing a closed term by induction over the inductive definition of large. Select $x,y,z \in 0$. For the basis case suppose $\tau(k)$ has at least two components and for $1 \leq i \leq t$ choose $x_i \in \tau(i)$; then

$$U_\tau = \lambda x_1 \ldots x_t x_k (\Lambda\tau(k1)x)(\Lambda\tau(k2)y)(\Lambda\tau(k3)z)\ldots(\Lambda\tau(kt(k))z).$$

For the induction step suppose $\tau(k)$ has one component and it is large. Select $x_1 \ldots x_t$ as in the basis case and set $U_\tau = \lambda x_1 \ldots x_t x_k U_{\tau(k1)}$.

Now let τ be large and contain a closed term. Let

$$\lambda x_1 \ldots x_t \; x_i X_1 \ldots X_{t(i)}$$

be a closed long $\beta\eta$ normal term in τ. Put

$$V_\tau = [x_i X_1 \ldots X_{t(i)}/z] \quad (U_\tau x_1 \ldots x_{t(i)})$$

and set

$$M_\tau = \lambda u \lambda x_1 \ldots x_t \; u(\lambda xy V_\tau)(x_i X_1 \ldots X_{t(i)})$$

for $u \in (0 \to (0 \to 0)) \to (0 \to 0)$.

CLAIM. Let $M_1 M_2$ be distinct Φ normal $\in (0 \to (0 \to 0)) \to (0 \to 0)$; then $M_\tau M_1 \neq M_\tau M_2$.

Proof. By induction on the length of M_1. First let $M = \lambda uv \; u(L_1 uv)(L_2 uv) \in (0 \to (0 \to 0)) \to (0 \to 0)$. We have

$$M_\tau M =_{\beta\eta} \lambda x_1 \ldots x_t (\lambda xy V_\tau)(L_1(\lambda xy V_\tau)(x_i X_1 \ldots X_t))(L_2(\lambda xy V_\tau)(x_i X_1 \ldots X_t))$$

$$=_{\beta\eta} \lambda x_1 \ldots x_t (\lambda xy V_\tau)(M_\tau L_1 x_1 \ldots x_t)(M_\tau L_2 x_1 \ldots x_t).$$

The remainder of the argument is routine using the above computation at the induction step. End of proof.

THEOREM 1. All large types are equivalent.

We shall see the large types are a maximum element of our partial ordering.

LEMMA 8. Each small type is reducible to a small type of rank ≤ 4.

Proof. For each τ let $\tilde{\tau} = \underbrace{(0 \to 0) \to (\ldots((0 \to 0)}_{t} \to 0)\ldots)$, and for τ tiny set $\approx\tau = \tilde{\tau}(1) \to 0$. For tiny τ select $u(\tau) \in \approx\tau$ and let $y, y_1, y_2, \ldots \in 0 \to 0$. For tiny τ and small σ define terms $V_\tau(y) \in \tau$ and $U_\sigma \in \sigma \to \tilde{\sigma}$ simultaneously as follows. $V_0(y) = yu(0)$ and $U_0 = \lambda x\, x$ for $x \in 0$.

$$V_\tau(y) = \lambda v\; y(u(\tau)(U_{\tau(1)}v)),$$

$$U_\sigma = \lambda z \lambda y_1 \ldots y_s\; zV_{\sigma(1)}(y_1)\ldots V_{\sigma(s)}(y_s)$$

for $\tau \neq 0 \neq \sigma$ and $v \in \tau(1)$, $z \in \sigma$.

Now let σ be small and let $M = \lambda x_1 \ldots x_s x_i(Nx_1 \ldots x_s) \in \sigma$ be in Φ-normal form. We have

$$\begin{aligned}
U_\sigma M &=_{\beta\eta} \lambda y_1 \ldots y_s\; V_{\sigma(i)}(y_i)(NV_{\sigma(1)}(y_1)\ldots V_{\sigma(s)}(y_s)) \\
&=_{\beta\eta} \lambda y_1 \ldots y_s\; y_i(u(\sigma(i))(U_{\sigma(i1)}(NV_{\sigma(1)}(y_1)\ldots V_{\sigma(s)}(y_s)))) \\
&=_{\beta\eta} \lambda y_1 \ldots y_s\; y_i(u(\sigma(i))(\lambda y_{s+1}\ldots y_{s+s(i1)} NV_{\sigma(1)}(y_1)\ldots V_{\sigma(s)}(y_s) \\
&\qquad\qquad V_{\sigma(i11)}(y_{s+1})\ldots V_{\sigma(i1s(i1))}(y_{s+s(i1)}))) \\
&=_{\beta\eta} \lambda y_1 \ldots y_s\; y_i(u(\sigma(i))(U_\rho N\, y_1 \ldots y_s))) \\
&\qquad\qquad \text{for } \rho = \sigma(1) \to (\ldots(\sigma(s) \to \sigma(i1)\ldots).
\end{aligned}$$

Using the above computation it is easy to prove by induction on the length of M that if $M \neq N \in \sigma$ are Φ-normal then $U_\sigma M \neq_{\beta\eta} U_\sigma N$. End of proof.

LEMMA 9. Each small type is reducible to a small type of rank ≤ 4 all of whose components of rank 3 are $((0 \to 0) \to 0) \to 0$.

Proof. The proof is just like the proof of Lemma 3. We illustrate with

$$(((0 \to 0) \to ((0 \to 0) \to 0)) \to 0) \to (0 \to 0)$$

which is reducible to

$$(0 \to 0) \to ((0 \to 0) \to ((((0 \to 0) \to 0) \to 0) \to (0 \to 0))$$

by means of

$$\lambda u\lambda y_1 y_2\lambda x\ u(\lambda z\ x(\lambda y_3\ z(\lambda v_1\ y_1(y_3v_1))(\lambda v_2\ y_2(y_3v_2))))$$

where $v_1,v_2 \in 0$, $y_1,y_2,y_3 \in 0 \to 0$, $z \in (0 \to 0) \to ((0 \to 0) \to 0)$, $x \in (((0 \to 0) \to 0)) \to 0$ and $u \in (((0\to 0)\to((0\to 0)\to 0))\to 0)\to(0\to 0)$. End of proof.

LEMMA 10. Each small type is reducible to a small type of the form $(((0 \to 0) \to 0) \to 0) \to \sigma$ where σ has rank ≤ 2.

Proof. Just like the proof of Lemma 4. End of proof.

LEMMA 11. If σ is small and of rank ≤ 2 then

$$(((0 \to 0) \to 0) \to 0) \to \sigma$$

is reducible to

$$(((0 \to 0) \to 0) \to 0) \to (0 \to 0).$$

Proof. We can assume that σ is rank righteous and has r components equal to $0 \to 0$. Let τ be the small rank righteous type of rank ≤ 2 with $s = t-1$ components equal to $0 \to 0$, then $(((0 \to 0) \to 0) \to 0) \to \sigma$ is reducible to $(((0 \to 0) \to 0) \to 0) \to \tau$ by means of the term

$$\lambda u\lambda x\lambda y_1\ldots y_s\lambda z\ uxy_1\ldots y_r\ (y_{r+1}z)\ldots(y_sz)$$

where $z \in 0$, $y_1\ldots y_s \in 0 \to 0$, $x \in ((0 \to 0) \to 0) \to 0$ and $u \in (((0 \to 0) \to 0) \to 0) \to \sigma$. Finally $(((0 \to 0) \to 0) \to 0) \to \tau$ is reducible to $(((0 \to 0) \to 0) \to 0) \to (0 \to 0)$ by means of

$$\lambda v\lambda x\lambda z\ x(\lambda y_1\ x(\lambda y_2\ldots x(\lambda y_s(vxy_1\ldots y_sz))\ldots))$$

for $v \in (((0 \to 0) \to 0) \to 0) \to \tau$. End of proof.

Summarizing Lemmas 9-11:

PROPOSITION 2. Each small type is reducible to the type $(((0 \to 0) \to 0) \to 0) \to (0 \to 0)$.

LEMMA 12. If σ is small and has rank 3 with at most one component of rank ≥ 1 or has rank 2 with at most one component of rank 1 then σ is equivalent to $(0 \to 0) \to (0 \to 0)$.

Proof. First recall that $(0 \to 0) \to (0 \to 0)$ is the type of λ-numbers $\underline{m} = \lambda xy\ \underbrace{x(\ldots(xy)\ldots)}_{m}$ and that the λ-definable number theoretic functions are just the extended polynomials (Statman (1979a), Proposition 2).

Suppose σ has rank 2 with at most 1 component of rank 1 and is rank righteous. A typical closed long $\beta\eta$ normal form $M \in \sigma$ is

$$M = \lambda x \lambda y_1 \ldots y_{s-1} \underbrace{x(\ldots(xy_i)\ldots)}_{k}.$$

Such a term is determined by the numbers k and i. Let $L_1 = \lambda u \lambda xy\ ux\underbrace{y\ldots y}_{s-1}$ and $L_2 = \lambda u \lambda xy\ ux(\underline{1}xy)\ldots(\underline{s-1}\ xy)$ for $y \in 0$, $x \in 0 \to 0$, and $u \in \sigma$. Then $L_1 M =_{\beta\eta} \underline{k}$ and $L_2 M =_{\beta\eta} \underline{k+i}$. Since there is a polynomial pairing function, σ is reducible to $(0 \to 0) \to (0 \to 0)$. The converse is obvious.

Now suppose σ has rank 3 with at most one component of rank ≥ 1 and is rank righteous. A typical closed long $\beta\eta$ normal form $\in \sigma$ is

$$M = \lambda u \lambda v_{01} \ldots v_{0s-1} u(\lambda v_{n1} \ldots v_{ns(11)} u(\ldots u(\lambda v_{11} \ldots v_{1s(11)} v_{ij})\ldots)).$$

Such a term is determined by the numbers n, i, and j. Let

$$L_1 = \lambda w \lambda xy\ w(\lambda z\ x(z\underbrace{y\ldots y}_{s(11)}))\underbrace{y\ldots y}_{s-1}$$

$$L_2 = \lambda w \lambda xy\ w(\lambda z\ x(\underbrace{z(\underbrace{z(xy)\ldots(xy)}_{s(11)})\ldots(\underbrace{z(xy)\ldots(xy)}_{s(11)})}_{s(11)}))\ \underbrace{y\ldots y}_{s-1}$$

$$L_3 = \lambda w \lambda xy\ w(\lambda z\ x(z(\underline{1}xy)\ldots(\underline{s(11)}\ xy))(\underline{1}xy)\ldots(\underline{s-1}xy)$$

where $w \in \sigma$ and $z \in \underbrace{0 \to (\ldots(0}_{s(11)} \to 0)\ldots)$. Then $L_1 M =_{\beta\eta} \underline{n}$, $L_2 M =_{\beta\eta} \underline{n+i}$ and $L_3 M =_{\beta\eta} \underline{n+j}$. Now proceed as in the previous case. The converse is obvious. End of proof.

LEMMA 13. Each small type of rank 3 is reducible to a small type of rank 3 whose only component of rank 2 is $(0 \to 0) \to 0$.

Proof. Let $\sigma = \sigma(1) \to \tau$ be small of rank 3 and rank righteous. Put $\rho = \underbrace{(0 \to 0) \to (\ldots((0 \to 0)}_{s(1)} \to (((0 \to 0) \to 0) \to \tau))\ldots)$. Then σ is reducible to ρ by means of the term

$$\lambda u \lambda y_1 \ldots y_{s(1)} \lambda x\ u(\lambda w\ x(\lambda z\ w(y_1 z)\ldots(y_{s(1)} z)))$$

where $z \in 0$, $y_1,\ldots,y_{s(1)} \in 0 \to 0$, $w \in \underbrace{0 \to (\ldots(0}_{s(1)} \to 0)\ldots)$, $x \in (0 \to 0) \to 0$ and $u \in \sigma$. The lemma follows by repeatedly alternating this reduction with permutation of components. End of proof.

LEMMA 14. Each small type of rank 3 is reducible to a small type of the form $((0 \to 0) \to 0) \to \sigma$ for σ a small type of rank ≤ 2.

Proof. Just like the proof of Lemma 4. End of proof.

LEMMA 15. If σ is small and has rank ≤ 2 then $((0 \to 0) \to 0) \to \sigma$ is reducible to a small type of rank 2.

Proof. Let σ be small of rank ≤ 2 and rank righteous with r rank 1 components. A typical closed long $\beta\eta$ normal form $\in ((0 \to 0) \to 0) \to \sigma$ is

$$M = \lambda x \lambda y_1 \ldots y_r\ \lambda z_{n+s-r} \ldots z_{n+1}\ W_{n+1}(x(\lambda z_n \ldots W_2(x(\lambda z_1\ W_1(z_i)))\ldots))$$

where $z_1 \ldots z_{n+s+r} \in 0$, $y_1 \ldots y_r \in 0 \to 0$, $x \in (0 \to 0) \to 0$ and for $1 \leq j \leq n+1$ W_j is a word on $y_1 \ldots y_r$ associated to the right. Let

$$N = \lambda u \lambda y_{r+1}\ z_{n+s-r+1} u(\lambda w\ y_{r+1}(w z_{n+s-r+1}))$$
$$\in (((0 \to 0) \to 0) \to \sigma) \to ((0 \to 0) \to (0 \to \sigma))$$

where $z_{n+s-r+1} \in 0$, y_{r+1}, $w \in 0 \to 0$ and $u \in ((0 \to 0) \to 0) \to \sigma$. We have

$$NM =_{\beta\eta} \lambda y_{r+1} \lambda z_{n+s-r+1} \lambda y_1 \ldots y_r z_{n+s-r} \ldots z_{n+1}$$
$$W_{n+1}(y_{r+1}(\ldots(W_2(y_{r+1}(W_1(z_j))))\ldots)).$$

Thus M is determined by NM and the number i. By the method of Lemma 12 it is easy to construct $L \in (((0 \to 0) \to 0) \to \sigma) \to ((0 \to 0) \to (0 \to 0))$ satisfying $LM =_{\beta\eta} \underline{n+s-r+i-1}$. Since s-r is fixed and NM determines n, NM and LM determine M. Now let

$$\tau = (0 \to 0) \to ((0 \to 0) \to ((0 \to 0) \to (0 \to \sigma)))$$

and set

$$T = \lambda u \lambda y_{r+3} y_{r+2} y_{r+1} \lambda z_{n+s-r+1} \lambda y_1 \ldots y_r \lambda z_{n+s-r} \ldots z_{n+1}$$
$$Nu\ y_{r+3}(y_{r+2}(Luy_{r+1}\ z_{n+s-r+1} y_1 \ldots y_r\ z_{n+s-r} \ldots z_{n+1}))$$

for $u \in ((0 \to 0) \to 0) \to \sigma$ and y_{r+2}, $y_{r+3} \in 0 \to 0$. It is easy to see that $((0 \to 0) \to 0) \to \sigma$ is reducible to τ by means of T. End of proof.

LEMMA 16. Each small type of rank 2 is reducible to $(0 \to 0) \to ((0 \to 0) \to (0 \to 0))$.

Proof. The proof is a simple binary coding of words on an n letter alphabet combined with the construction in Lemma 12. End of proof.

LEMMA 17. If σ has rank 3 with at least 2 components of rank 2, or rank 2 with at least 2 components of rank 1 and small, then $(0 \to 0) \to ((0 \to 0) \to (0 \to 0))$ is reducible to σ. If σ has rank ≥ 4 and is small then $(0 \to 0) \to ((0 \to 0) \to (0 \to 0))$ is reducible to σ.

Proof. Just like the proof of Lemma 7. End of proof.

LEMMA 18. $(0 \to 0) \to ((0 \to 0) \to (0 \to 0))$ is not reducible to $(0 \to 0) \to (0 \to 0)$.

Proof. Consider the structure P_ω with ground domain = the set of natural numbers N, and let s be the successor function. Let A = the Gandy hull of $M \cup \{s\}$ in P_ω, then for $M,N \in (0 \to 0) \to (0 \to 0)$, $A \models M = N \iff M =_{\beta\eta} N$, but for $z \in 0$ and $x,y \in 0 \to 0$,

$$A \models \lambda xy\lambda z\ x(x(y(yz))) = \lambda xy\lambda z\ x(y(x(yz))).$$

End of proof.

OPEN QUESTION. Is $(((0 \to 0) \to 0) \to 0) \to (0 \to 0)$ reducible to $(0 \to 0) \to ((0 \to 0) \to (0 \to 0))$?

PROPOSITION 3. $(0 \to (0 \to 0)) \to (0 \to 0)$ is not reducible to $(((0 \to 0) \to 0) \to 0) \to (0 \to 0)$.

Proof. Consider P_ω with ground domain = the domain of the free algebra generated by the unary functions f, g and the constant a. Let B = the Gandy hull of $\{a,f,g\}$ in P_ω. A typical closed long $\beta\eta$ normal form $\in (((0 \to 0) \to 0) \to 0) \to (0 \to 0)$ is

$$M = \lambda xz\ x(\lambda y_1 U_1(x(\lambda y_2 \ldots \lambda y_n U_n(x(\lambda y_{n+1} U_{n+1}(z)))\ldots)))$$

where $-1 \leq n$, and $1 \leq i \leq n+1 \Rightarrow U_i$ is a word on $y_1 \ldots y_i$ associated to the right. Let

$L = \lambda u \lambda v w\ u(\lambda y\ w(yv))$
$\quad \varepsilon\ ((((0 \to 0) \to 0) \to 0) \to (0 \to 0)) \to ((0 \to 0) \to ((0 \to 0) \to (0 \to 0)))$

for $u \ \varepsilon\ (((0 \to 0) \to 0) \to 0) \to (0 \to 0)$, $w, v \ \varepsilon\ 0 \to 0$, and $y \ \varepsilon\ (0 \to 0) \to 0$. Then

$$LM =_{\beta\eta} \lambda v w \lambda z\ w(V_1(w(\ldots(V_n(w(V_{n+1}(z)))\ldots)))$$

where for $1 \leq i \leq n+1$, $V_i = [v/y_1, \ldots, v/y_i]U_i$. Now for all $N_1, N_2 \ \varepsilon\ (0 \to 0) \to ((0 \to 0) \to (0 \to 0))$,

$$\mathcal{B} \models N_1 = N_2 \iff N_1 =_{\beta\eta} N_2 ,$$

thus for $N \ \varepsilon\ (((0 \to 0) \to 0) \to 0) \to (0 \to 0)$, if $\mathcal{B} \models N = M$ then the long $\beta\eta$ normal form of N has the same length as M. A typical closed long $\beta\eta$ normal form $\varepsilon\ (0 \to (0 \to 0)) \to (0 \to 0)$ can be thought of as a λ-prefix of length 2 followed by a binary tree. It is easy to see that two such terms are equal in $\mathcal{B}$ <=> they have rightmost paths of the same length and leftmost paths of the same length. In particular, there is a closed long $\beta\eta$ normal form $\varepsilon\ (0 \to (0 \to 0)) \to (0 \to 0)$ equal in $\mathcal{B}$ to infinitely many distinct closed long $\beta\eta$ normal forms. End of proof.

Summarizing Lemmas 8-18 and Proposition 3.

THEOREM 2. The equivalence classes of types are well ordered in type $\omega + 2$ or $\omega + 3$. A system of representatives is the following:

0. types without closed terms

1. $0 \to 0$

$\vdots$

n. $\underbrace{0 \to (\ldots(0}_{n} \to 0)\ldots)$

$\vdots$

ω. $(0 \to 0) \to (0 \to 0)$ (small types of rank 3 with at most one component of rank ≥ 1 or rank 2 with at most one component of rank 1)

$\omega + 1$. $(0 \to 0) \to ((0 \to 0) \to (0 \to 0))$ (all other small types of rank 2 or 3)

$\omega + 2(?)$. $(((0 \to 0) \to 0) \to 0) \to (0 \to 0)$ (small types of rank ≥ 4)

$\omega + 3$. $(0 \to (0 \to 0)) \to (0 \to 0)$ (large types)

ENUMERATION

Here two types are equivalent if each enumerates the other. We shall consider the partial ordering of enumeration among equivalence classes. Unlike reducibility this partial ordering is not a linear ordering since the equivalence class of all types not containing closed terms is incomparable with any other. Below we shall consider only types with closed terms.

We say σ *is higher than* τ if σ can be obtained from a third type $\neq 0$ by substituting τ for 0.

LEMMA 19. If $\tau \neq 0 \to 0$ then there is an $M \in \tau \to \tau$ such that for $N \in \tau$ $MN \underset{\beta\eta}{\neq} N$.

Proof. Case 1: τ has rank 1 with $t > 1$. Let

$$M = \lambda y \lambda x_1 \ldots x_t \; y x_2 \ldots x_t x_1$$

where $y \in \tau$ and for $1 \leq i \leq t$, $x_i \in 0$.

Case 2: rank $(\tau) > 1$. Select a closed $\lambda x_1 \ldots x_t \; x_i X_1 \ldots X_{t(i)} \in \tau$ and put

$$M = \lambda y \lambda x_1 \ldots x_t \; x_i (\Lambda\tau(i1) y x_1 \ldots x_t) X_2 \ldots X_{t(i)}.$$

End of proof.

PROPOSITION 4. If σ is higher than τ then τ does not enumerate σ.

Proof. We modify Cantor's diagonal argument. Let

$$\sigma(1) = \sigma(11) \to (\ldots(\sigma(1r) \to \tau)\ldots).$$

If $\tau = 0 \to 0$ we are done since τ is then the only type with exactly one $\beta\eta$ normal form. Assume $\tau \neq 0 \to 0$. Let M be the term given by Lemma 19 for $\sigma(2) \to (\ldots(\sigma(s) \to 0)\ldots)$ and let $\lambda x_2 \ldots x_s \; x_i X_1 \ldots X_{s(i)}$ be a closed long $\beta\eta$ normal form of this type. Select $u \in \tau \to \sigma$ and $v \in \sigma\,(1)$ and put

$N = \lambda u \lambda v \lambda x_2 \ldots x_s \; M(u(v(x_i X_1 \ldots X_{s(i)}) \ldots (x_i X_1 \ldots X_{s(i)}))v) x_2 \ldots x_s.$

For $1 \leq i \leq r$ let $y_i \in \sigma(1i)$. For $M_1 \in \tau \to \sigma$ and $N_1 \in \tau$, if

$M_1N_1 =_{\beta\eta} NM_1$ then

$$M_1N_1(\lambda y_1\ldots y_rN_1) =_{\beta\eta} NM_1(\lambda y_1\ldots y_rN_1)$$
$$=_{\beta\eta} \lambda x_2\ldots x_s\ M(M_1N_1(\lambda y_1\ldots y_rN_1))x_2\ldots x_s$$
$$=_{\beta\eta} M(M_1N_1\lambda y_1\ldots y_rN_1)$$

which contradicts the choice of M. End of proof.

PROPOSITION 5. If σ occurs positive in τ then τ enumerates σ.

Proof. Let $\lambda x_1\ldots x_s\ x_iX_1\ldots X_{s(i)}\ \varepsilon\ \sigma$ be a closed long $\beta\eta$ normal term. Define terms $U_\tau(x)$ for $x\ \varepsilon\ 0$ and $V_\tau(x)$ by induction over the definition of 'σ occurs positive in τ' for all τ, possibly not containing a closed term, as follows. If $\sigma = \tau$ then $U_\tau(x) = \lambda u\ ux_1\ldots x_s$ and $V_\tau(x) = \lambda u\ u$ for $u\ \varepsilon\ \tau$. If $\sigma \neq \tau$ let $\tau = \tau(1) \to \rho$.

Case 1: σ occurs positive in ρ. Define

$$U_\tau(x) = \lambda u_1\ U_\rho(x)(u_1\Lambda\tau(1)x),$$
$$V_\tau(x) = \lambda u_2\lambda y\ V_\rho(x)u_2$$

for $u_1\ \varepsilon\ \tau$, $u_2\ \varepsilon\ \sigma$ and $y\ \varepsilon\ \tau(1)$.

Case 2: σ occurs negative in $\tau(1)$. Let σ occur positive in $\tau(1k)$. For $1 \leq i \leq t(1)$ select $w_i\ \varepsilon\ \tau(1i)$ and for $1 \leq j \leq t$ choose $y_i\ \varepsilon\ \tau(i)$, then define

$$U_\tau(x) = \lambda u_1\ u_1(\lambda w_1\ldots w_{t(i)}\ V_{\tau(1k)}(x)w_k)(\Lambda\tau(2)x)\ldots(\Lambda\tau(t)x)$$

and

$$V_\tau(x) = \lambda u_2\lambda y_1\ldots y_t\ y_1(\Lambda\tau(11)x)\ldots(V_{\tau(1k)}(x)u_2)\ldots(\Lambda\tau(1t(11))x)$$

for $u_1\ \varepsilon\ \tau$ and $u_2\ \varepsilon\ \sigma$.

It is easy to see that for $z\ \varepsilon\ \sigma$, $U_\tau(x)(V_\tau(x)z) =_{\beta\eta} zx_1\ldots x_s$. Let τ contain a closed term and let $\lambda y_1\ldots y_t\ y_j\ Y_1\ldots Y_{t(j)}$ be a closed long $\beta\eta$ normal form $\varepsilon\ \tau$. Put

$$M_\tau = \lambda u_1\lambda x_1\ldots x_s\ U_\tau(x_iX_1\ldots X_{s(i)})u_1$$

and

$$N_\tau = \lambda u_2\lambda y_1\ldots y_t\ V_\tau(y_jY_1\ldots Y_{t(j)})uy_1\ldots y_t$$

for $u_1 \in \sigma$ and $u_2 \in \tau$. We have for $M \in \sigma$ $M_\tau(N_\tau M) =_{\beta\eta} M$. End of proof.

LEMMA 20. Let $a \in P_2^0$, then the number of elements of P_2^τ λ-definable from a is at least $|\tau|$.

Proof. By induction on $|\tau|$. When $\tau = 0$ the lemma is obvious. Let $\tau \neq 0$. For $1 \leq i \leq t$ select $x_i \in \tau(i)$, and for $1 \leq j \leq t(i)$, $y_{ij} \in \tau(ij)$. For each $1 \leq i \leq t$ and each function

$$f:[1,t(i)] \to \{x_i y_{i1} \ldots y_{it(i)}, x_i(\Lambda\tau(i1)x_i y_{i1} \ldots y_{it(i)}) \ldots (\Lambda\tau(it(i))x_i y_{i1} \ldots y_{it(i)})\}$$

set

$$M_{i,f} = \lambda y_{i1} \ldots y_{it(i)} \lambda x_i\ x_i(\Lambda\tau(i1)f(1)) \ldots (\Lambda\tau(it(i))f(t(i))).$$

Let $M_1, N_1, \ldots, M_{t(i)}, N_{t(i)}$ $\in$ resp. $\tau(i1), \ldots, \tau(it(i))$ have for some $1 \leq j \leq t(i)$ $[\![M_j]\!] \neq [\![N_j]\!]$ and have a as a possible parameter. It is easy to see that

$$f \neq g \Rightarrow [\![M_{ig}M_1 \ldots M_{t(i)}]\!] \neq [\![M_{if}M_1 \ldots M_{t(i)}]\!]$$
$$\neq [\![M_{if}N_1 \ldots N_{t(i)}]\!] \neq [\![M_{ig}M_1 \ldots M_{t(i)}]\!].$$

Also observe that the normal form of $M_{if}M_1 \ldots M_{t(i)}$ has head variable x_i. By induction hypothesis for $1 \leq i \leq t$ and $1 \leq j \leq t(i)$ there are at least $|\tau(ij)|$ elements of $P_2^{\tau(ij)}$ λ-definable from a. Thus for $1 \leq i \leq t$ there are at least

$$2^{t(i)} \prod_{1 \leq j \leq t(i)} |\tau(ij)| \geq 1 + \sum_{1 \leq j \leq t(i)} |\tau(ij)| = |\tau(i)|$$

elements of $P_2^{\tau(i)\to 0}$ λ-definable from a by a term with head variable x_i. Hence there are at least

$$1 + \sum_{1 \leq i \leq t} |\tau(i)| = |\tau|$$

elements of P_2^τ λ-definable from a. End of proof.

LEMMA 21. If τ contains a closed term then the number of λ-definable members of $P_2^\tau \geq |\tau|-1$.

Proof. Let open $(\tau) = 1$ if τ does not contain a closed term and = 0 else) open is Γ invariant for each Γ. The proof is by induction on τ. We shall assume that τ is open righteous. When $\tau = 0$ the lemma is vacuous. Let $t \neq 0$ contain a closed term. Choose $\lambda x_1 \ldots x_t\ x_1 X_1 \ldots X_{t(1)}$ a closed long $\beta\eta$ normal form $\varepsilon\ \tau$ and $x\ \varepsilon\ 0$. By Lemma 20 there are closed long $\beta\eta$ normal forms $L_1, \ldots L_{|\sigma|}\ \varepsilon\ 0 \to \sigma$ such that for $1 \leq i < j \leq |\sigma|$ $[\![L_i]\!]a \neq [\![L_j]\!]a$ for any $a\ \varepsilon\ P_2^0$ where $\sigma = \tau(2) \to (\ldots(\tau(t) \to 0)\ldots)$. For $1 \leq i \leq |\sigma|$ set

$$N_i = \lambda x_1 \ldots x_t\ L_i(x_1 X_1 \ldots X_{t(1)})x_2 \ldots x_t,$$

then for $1 \leq i < j \leq |\sigma|$ $[\![N_i]\!] \neq [\![N_j]\!]$. Observe also that at most one N_i has a long $\beta\eta$ normal form with x_1 as head variable. Select for $1 \leq i \leq t(1)$ $y_i\ \varepsilon\ \tau(1i)$. Put

$$T = x_1 y_1 \ldots y_{t(i)},$$

$$\bot = x_1(\Lambda\tau(11)x_1 y_1 \ldots y_{t(1)}) \ldots (\Lambda\tau(1t(1))x_1 y_1 \ldots y_{t(1)}) .$$

For each $f:[1,t(1)] \to \{T,\bot\}$ set

$$M_f = \lambda y_1 \ldots y_{t(1)} \lambda x_1\ x_1(\Lambda\tau(11)f(1)) \ldots (\Lambda\tau(1t(1))f(t(1))),$$

and set

$$M_\phi = \lambda y_1 \ldots y_{t(1)} \lambda x_1\ T.$$

Let $\rho = \underbrace{0 \to (\ldots(0}_{t(1)} \to 0)$ and $P_2^0 = \{a,b\}$. For each $\psi\ \varepsilon\ P_2^\rho$ there is a $\Phi_{\psi,c}\ \varepsilon\ P_2^{\tau(1)}$ such that, for any $g:[1,t(1)] \to P_2^0$,

$$\Phi_{\psi,c}(\Lambda\tau(11)g(1)) \ldots (\Lambda\tau(1t(1))g(t(1))) = \psi g(1) \ldots g(t(1))$$

and for any $M_1, \ldots, M_{t(1)}$ ε resp. $\tau(11), \ldots, \tau(1t(1))$

$$\Phi_{\psi,c}[\![M_1]\!] \ldots [\![M_{t(1)}]\!] = c.$$

For $f:[1,t(1)] \to \{T,\bot\}$ define $g_f:[1,t(1)] \to P_2^0$ by $g_f(i) = a$ if $f(i) = T$ else $g_f(i) = b$, for $1 \leq i \leq t(1)$.

Now suppose $M_1, N_1, \ldots, M_{t(1)}, N_{t(1)}$ ε resp. $\tau(11), \ldots, \tau(1t(1))$ and for some $1 \leq i \leq t(1)$ $[\![M_i]\!] \neq [\![N_i]\!]$. Suppose $f_1 \neq f_2:[1,t(1)] \to \{T,\bot\}$ and select $\psi\ \varepsilon\ P_2^\rho$ such that

$$\psi g_{f_1}(1)\ldots g_{f_1}(t(1)) \neq \psi g_{f_2}(1)\ldots g_{f_2}(t(1))$$

and $\psi a \ldots a = b$. Then

$$[\![M_{f_1} M_1 \ldots M_{t(1)}]\!]\Phi_{\psi,a} \neq [\![M_{f_2} M_1 \ldots M_{t(1)}]\!]\Phi_{\psi,a}$$
$$\neq [\![M_{f_1} N_1 \ldots N_{t(1)}]\!]\Phi_{\psi,a}.$$

Also if ψ is constantly b then

$$[\![M_{f_1} M_1 \ldots M_{t(1)}]\!]\Phi_{\psi,a} \neq [\![M_{\phi} M_1 \ldots M_{t(1)}]\!]\Phi_{\psi,a}$$
$$\neq [\![M_{f_1} N_1 \ldots N_1 \ldots N_{t(1)}]\!]\Phi_{\psi,a}.$$

In addition, obviously,

$$[\![M_\phi M_1 \ldots M_{t(1)}]\!] \neq [\![M_\phi N_1 \ldots N_{t(1)}]\!].$$

Finally let ψ be invariant ε P_2^0 with $\psi a \ldots a = b$. Select $\chi \;\varepsilon\; P_2^{\tau(1)}$ such that

$$\chi[\![M_1]\!]\ldots[\![M_{t(1)}]\!] = a,$$
$$\chi[\![N_1]\!]\ldots[\![N_{t(1)}]\!] = b$$

and for any $g:[1,t(1)] \to P_2^0$

$$\chi(\Lambda\tau(11)g(1))\ldots(\Lambda\tau(1t(1))g(t(1))) = \psi g(1)\ldots g(t(1)).$$

Then for any $f:[1,t(1)] \to \{T,\bot\}$ we have

$$[\![M_f M_1 \ldots M_{t(1)}]\!]\chi \neq [\![M_f N_1 \ldots N_{t(1)}]\!]\chi.$$

By induction hypothesis, for $1 \leq i \leq t(1)$ there are at least $|\tau(1i)|-1$ λ-definable elements of $P_2^{\tau(1i)}$. Thus if $t(1) \neq 0$ then there are at least $(2^{t(1)}+1) \prod_{1\leq i\leq t(1)} |\tau(1i)|-1 > 1 + \sum_{1\leq i\leq t(1)} |\tau(1i)|$ λ-definable elements of $P_2^{\tau(1)\to 0}$. Note that each λ-definable element of $P_2^{\tau(1)\to 0}$ can be defined by a long $\beta\eta$ normal form with head variable x_1. Thus there are at least $(|\tau(1)| + |\sigma|)-1 = |\tau|-1$ λ-definable elements of P_2^τ. End of proof.

PROPOSITION 6. Let $s(1) = 2$ and $s(n+1) = 2^{s(n)}$, then if $|\tau| > 1 + s(|\sigma|)^{\frac{1}{2}}$ then σ does not enumerate τ.

Proof. P_2^σ contains at most $s(|\sigma|)$ elements and contains exactly $|P_2^{\sigma}|^{\frac{1}{2}}$ invariant ones. Thus P_2^{σ} contains at most $s(|\sigma|)^{\frac{1}{2}}$ λ-definable elements. Now apply Lemma 2.1. End of proof.

An example of a pair of incomparable types containing closed terms is $((0 \to 0) \to 0) \to 0$ and $(0 \to 0) \to (0 \to 0)$. There is only one λ-definable element of type $((0 \to 0) \to 0) \to 0$ in the model M (Statman (1979c), Lemma 5) but there are two of type $(0 \to 0) \to (0 \to 0)$. On the other hand there are only three λ-definable members of $P_2^{(0 \to 0) \to (0 \to 0)}$ but there are at least four in $P_2^{((0 \to 0) \to 0) \to 0}$.

A RANDOM OBSERVATION

The number t_n of types τ with $|\tau| = n$ is given by $t_n = \sum_{1 \le i \le n-1} t_i t_{n-i}$. t_n is the nth Catalan number and it is well known that $t_n = 1/n \binom{2n-2}{n-1}$. Let $\rho = \rho_n$ be a random variable (see Erdös and Spencer (1974)) so Prob $[\rho_n = \tau] = 1/t_n$.

LEMMA 22. $t_n \le 4\, t_{n-1}$ and $\lim_{n\to\infty}(t_{n-1}/t_n) = 1/4$.

Proof. A simple construction shows $t_n \le (2(2(n-1)-1)/n)t_{n-1}$. End of proof.

PROPOSITION 7. $0.14 \le$ Prob [ρ contains no closed term] ≤ 0.375.

Proof. The number O_n of types τ with $|\tau| = n$ not containing a closed term is given by

$$O_n = \sum_{1 \le i \le n-1} (t_i - O_i) O_{n-i}.$$

if $n \neq 1$ then $O_n \neq t_n$ and if $n \neq 2$ then $O_n \neq 0$. We prove by induction on n that

$$O_n \le \lceil t_n\ 1/2 \rceil.$$

If $n = 1,2$ the inequality is obvious. If $n > 2$ we have

$$\begin{aligned} O_n &\le \sum_{2 \le i \le n-1} (t_i - 1) \lceil t_{n-i}\ 1/2 \rceil \\ &\le \frac{1}{2} \sum_{2 \le i \le n-1} (t_i - 1)(t_i + 1) \\ &= 1/2(t_n - (t_{n-1} + (n-1)(n-2)/2 - 1) \\ &\le \lceil t_n\ 1/2 \rceil. \end{aligned}$$

Hence by the preceding calculation, if $n > 1$ then $0_n \leq t_n\ 1/2$ and if $n > 2$ then

$$0_n/t_n \leq 1/2(1-t_{n-1}/t_n).$$

Thus $\lim_{n\to\infty} 0_n/t_n \leq 3/8$. Now, for $n > 1$,

$$\begin{aligned} 0_n &\geq \left(\sum_{1\leq i\leq n-1} t_i\ 1/2\right) - \left(1/2 + t_{n-2}\ 1/2\right) \\ &\geq 1/2\ t_n\left(\sum_{1\leq i\leq n-1} 1/4^{n-i}\right) \\ &= 1/2(1+t_{n-2}) . \end{aligned}$$

Thus $\lim_{n\to\infty} 0_n/t_n \geq 1/6-1/32$. End of proof.

In short it is a good bet, but not a sure thing, that ρ contains a closed term.

APPLICATION

The *unification problem* $\tau_1,\ldots,\tau_n;\ \tau$ is the set of all pairs $M_1, M_2 \in \tau_1 \to (\ldots(\tau_n \to \tau)\ldots)$ such that for $1 \leq i \leq n$ there exists $N_i \in \tau_i$ satisfying $M_1N_1\ldots N_n =_{\beta\eta} M_2N_1\ldots N_n$.

Define simultaneously the degree and the index of τ by

$$\text{degree } (0) = 0, \text{ index } (0) = -1,$$

and if $\tau \neq 0$ then

$$\text{degree } (\tau) = \sum_{1\leq i\leq t} 1 + \text{index } (\tau(i)),$$

$$\text{index } (\tau) = \max_{1\leq i\leq t} \{\text{degree } (\tau(i))\}.$$

A simple induction shows that $\lfloor 1/2 \text{ rank } (t)\rfloor \leq \text{degree } (\tau)$. Define closed $(\tau) = 1$ if τ contains a closed term and $= 0$ else. The characteristic of τ is defined by

$$\text{char } (\tau) = \omega\cdot\text{closed}(\tau) + (-1)^{\text{open } (\tau)} \text{ degree } (\tau).$$

char is Γ invariant for each Γ.

LEMMA 23. If τ contains a closed term and has degree d then there are $D_1\ldots D_d \in \tau \to ((0 \to 0) \to (0 \to 0))$ such that for each sequence $m_1 \ldots m_d$ of positive integers there exists $M \in \tau$ satisfying for $1 \leq i \leq d$ $\underline{m}_i \underset{\beta\eta}{=} D_i M$.

Proof. We say a sequence $m_1 \ldots m_d$ of positive integers is represented by $N \in \underbrace{(0 \to 0) \to (\ldots((0 \to 0)}_{d} \to (0 \to 0))\ldots)$ if

$$N \underset{\beta\eta}{=} \lambda x_1 \ldots x_d \lambda y\ \underbrace{x_1(\ldots(x_1}_{m_1}(\ldots\underbrace{(x_d(\ldots(x_d}_{m_d}y)\ldots))\ldots))\ldots).$$

It is obvious that it suffices to find $D \in \tau \to \underbrace{((0 \to 0) \to (\ldots((0 \to 0)}_{d} \to (0 \to 0))\ldots)$ such that for each sequence $m_1 \ldots m_d$ of positive integers there exists $M \in \tau$ such that DM represents $m_1 \ldots m_d$.

Assume all types are characteristic righteous. Let σ be a type with degree d and index e and let $\vec{m} = m_1 \ldots m_d$ be a sequence of positive integers. Write $\vec{m} = \vec{n}_1 \ldots \vec{n}_s$ where for $1 \leq i \leq s$ $\vec{n}_i = \vec{m}_i n_i$ for $\vec{m}_i = m_{i1} \ldots m_{ie_i}$ and $e_i = \text{index}\ (\sigma(i))$. For $1 \leq j$ let $\vec{y}_j = y_1 \ldots y_j \in 0 \to 0$ and select $x, y, z \in 0$. In case σ contains a closed term, select $\lambda x_1 \ldots x_s\ x_k T_1 \ldots T_{s(k)} \in \sigma$ closed in long $\beta\eta$ normal form. We define terms $U_\sigma(\vec{m}, x, y) \in \sigma$, $X_\sigma(\vec{y}_{e+1}, z) \in \sigma$, $Y_\sigma(\vec{y}_d, z) \in \sigma \to 0$ and in case σ contains a closed term $R_\sigma(\vec{m}) \in \sigma$ and $D_\sigma \in \sigma \to \underbrace{((0 \to 0) \to (\ldots((0 \to 0)}_{d} \to (0 \to 0))\ldots)$ as follows.

$$U_0(\vec{m}, x, y) = x,\ X_0(\phi, z) = z,\ Y_0(\phi, z) = \lambda z\ X_0(\phi, z).$$

If $\sigma \neq 0$ and $\sigma(s) \neq 0$ for $1 \leq i \leq s$ and $1 \leq j$ define V_{ij} and Z_j as follows. If $e_i = \text{degree}\ (\sigma(i1))$ then

$$V_{i1} = x_i(U_{\sigma(i1)}(\vec{m}_i, xy))(\Lambda\sigma(i2)y)\ldots(\Lambda\sigma(is(i))y)$$

and otherwise

$$V_{i1} = x_i(\Lambda\sigma(i1)y)\ldots(U_{\sigma(is(i))}(\vec{m}_i, x, y)).$$

$$Z_1 = x_s\ R_{\sigma(s1)}(\vec{m}_s)(\Lambda\sigma(s2)y)\ldots(\Lambda\sigma(ss(s))y),$$

$$V_{ij+1} = x_i(\Lambda\sigma(i1)V_{ij})\ldots(\Lambda\sigma(is(i))V_{ij})\ ,$$

$$Z_{j+1} = x_s(\Lambda\sigma(s1)Z_j)\ldots(\Lambda\sigma(ss(s))Z_j)\ .$$

Set $V_i = V_{in_i}$ and $Z = Z_{n_s}$, then

$$U_\sigma(\vec{m},x,y) = \lambda x_1 \ldots x_s(\lambda x V_1)(\ldots((\lambda x V_{s-1})V_s)\ldots),$$

$$R_\sigma(\vec{m}) = \lambda x_1 \ldots x_s[x_k T_1 \ldots T_{s(k)}/y](\lambda x\ U_{\sigma(1)\to 0}(\vec{n}_1,x,y)x_1)$$
$$(\ldots((\lambda x\ U_{\sigma(s-1)\to 0}(\vec{n}_{s-1},x,y)x_{s-1})Z)\ldots).$$

In case $\sigma(s) = 0$ then

$$R_\sigma(\vec{m}) = \lambda x_1 \ldots x_s[x_k T_1 \ldots T_{s(k)}/y](\lambda x U_{\sigma(1)\to 0}(\vec{n}_1,x,y)x_1))\ldots$$
$$((\lambda x\ \ U_{\sigma(s-1)\to 0}(\vec{n}_{s-1},x,y)x_{s-1})x_s)\ldots).$$

$$X_\sigma(\vec{y}_{e+1},z) = \lambda x_1 \ldots x_s\ y_1([\ldots,y_{j+1}/y_j,\ldots]Y_{\sigma(r)}(\vec{y}_e,z)x_r)$$

where $r = 1$ if e = degree $(\sigma(1))$ and $= s$ otherwise.

$$Y_\sigma(\vec{y}_d,z) = \lambda u\ uX_{\sigma(1)}(\vec{y}_{e_1+1},z)\ldots X_{\sigma(s)}(y_{d-e_s-1}\ldots y_d,z)$$

for $u \in \sigma$. Finally $D_\sigma = \lambda u \lambda y_1 \ldots y_d \lambda z\ Y_\sigma(\vec{y}_d,z)u$.

It is easy to prove by induction that

$$U_\sigma(\vec{m},x,y)\ X_{\sigma(1)}(\vec{y}_{e+1},z)\ldots X_{\sigma(s)}(y_{d-e_s-1}\ldots y_d,z) \underset{\beta\eta}{=}$$
$$\underbrace{y_1(\ldots(y_1}_{m_1}(\ldots\underbrace{(y_d(\ldots(y_d}_{m_d}x)\ldots))\ldots))\ldots).$$

From this it follows by induction that

$$R_\sigma(\vec{m})\ X_{\sigma(1)}(\vec{y}_{e+1},z)\ldots X_{\sigma(s)}(y_{d-e_s-1}\ldots y_d,z) \underset{\beta\eta}{=}$$
$$\underbrace{y_1(\ldots(y_1}_{m_1}(\ldots\underbrace{(y_d(\ldots(y_d}_{m_d}z)\ldots))\ldots))\ldots)$$

and finally that $D_\sigma R_\sigma(\vec{m})$ represents $\vec{m}$. End of proof.

LEMMA 24. The unification problem

$$\underbrace{(0 \to 0) \to (0 \to 0)\ldots(0 \to 0) \to (0 \to 0)}_{n};\ (0 \to 0) \to (0 \to 0)$$

is recursively equivalent to Hilbert's 10th problem for polynomials with n variables.

Proof. Statman (1979a), Proposition 1. End of proof.

In particular, by an unpublished result of Matijacevič, the problem is undecidable for $n \geq 9$.

THEOREM 3. The unification problem $\tau_1,\ldots,\tau_n;\ \tau_{n+1}$ is

(1) decidable if rank of $\tau_{n+1} \leq 1$ or some τ_i contains no closed term and

(2) undecidable if each τ_i contains a closed term, rank $(\tau_i) \geq 2$ and $\sum_{1\leq i\leq n}$ degree $(\tau_i) \geq 9$.

Proof. (1) follows from Theorem 2 of Statman (1979c). (2) follows from Theorem 2 and Lemmas 23 and 24.

FOOTNOTES

1. We also use the following notations:

$[1,n]$ = the set of integers i such that $1 \leq i \leq n$.

$\lceil r \rceil$ = the least integer n such that $r \leq n$.

$\lfloor r \rfloor$ = the greatest integer n such that $n \leq r$.

σ occurs *positive* (*negative*) in τ if $\sigma = \tau$ or σ occurs negative (positive) in $\tau(1)$ or σ occurs positive (negative) in $\tau(2) \to (\ldots(\tau(t) \to 0)\ldots)$.

$M \in \tau$ is in Φ *normal form* if $M = \lambda x_1\ldots x_t\ x_i$ and $\tau(i) = 0$ or $M = \lambda x_1\ldots x_t\ x_j(M_1x_1\ldots x_t)\ldots(M_{t(j)}x_1\ldots x_t)$ and for $1 \leq k \leq t(j)$ M_k is in Φ normal form.

$X \in \tau$ is in *long* $\beta\eta$ *normal form* if $X = \lambda x_1\ldots x_t\ yX_1\ldots X_s$ and for $1 \leq i \leq s$ X_i is in long $\beta\eta$ normal form (here $y \in \{x_1,\ldots,x_t\}$ or y is a free variable of X).

2. <u>Case</u> 2; $T = vT_1\ldots T_n$ for $v \in 0\to(\ldots(0\to 0)\ldots)$. We have $T^+ = vT_1^+\ldots T_n^+$ and $T^* = vT_1^*\ldots T_n^*$, so this case follows from the induction hypothesis applied to $T_1\ldots T_n$.

ACKNOWLEDGEMENTS

The author would like to thank P. Hendreiks and G. Koldenhof for several useful remarks.

This material is based upon work supported by the National Science Foundation under Grant No. MCS-79-7923199.

REFERENCES

Erdos, P. and Spencer, J. (1974). Probabilistic Methods in Combinatorics, Academic Press, New York.

Statman, R. (1979a). The typed λ-calculus is not elementary recursive, *Theoretical Computer Science*, Vol. 9, No. 1, July 1979, pp. 73-81.

Statman, R. (1979b). Intuitionistic propositional logic is polynomial space complete, *Theoretical Computer Science*, Vol. 9, No. 1, July 1979, pp. 67-72.

Statman, R. (1979c). Completeness, invariance, and λ-definability submitted to the *Journal of Symbolic Logic*.

PRINCIPAL TYPE SCHEMES AND λ-CALCULUS SEMANTICS

M. Coppo, M. Dezani-Ciancaglini and B. Venneri
Istituto di Scienza dell'Informazione
Università di Torino
Corso M. D'Azeglio 42 - 10125 TORINO

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. INTRODUCTION

Basic functionality theory (as other illative systems) was introduced by Curry (Curry and Feys, 1958) for overcoming the weakness (revealed, for example, by some well-known "paradoxes") of pure combinators or λ-calculus systems as a support for studying properties of functions and logical deductions.

Let's recall that, according to (Curry *et al.*, 1958), a formula in a functionality theory is a statement τ X where X is a term and τ its functional character or type. So τ X means that τ is a type for X. Functional characters are built from a set of basic elements (which are left uninterpreted) and a composition operator F. If σ, τ are types, F σ τ is the type of a term which defines a function from terms of type σ to terms of type τ.

As pointed out by Curry (1969) and Hindley (1969), an important concept in functionality theory is the one of principal type scheme of an object. It turns out, in fact, that each typed term has an infinite set of functional characters rather than a single one. The existence of the principal type scheme of a stratified term X (in the sense that all types of X can be obtained from this one as instances) shows an internal coherence between all functional characters of X.

Curry's system, however, reveals itself too weak to be fully satisfactory. In fact types are not preserved by convertibility (unless we postulate it explicitly) and it is not possible to assign types to a large class of terms. In the papers (Coppo and Dezani-Ciancaglini, 1978), (Coppo, Dezani-Ciancaglini and Venneri, 1979 a), (Coppo, Dezani-Ciancaglini and Sallé, 1979 b) the authors have proposed (and proved consistent) an extension of Curry's system which, while preserving the essential features of the latter (in both type definition and deduction rules), increases in a substantial way the power of the system. It turns out, in particular, that types are preserved under convertibility and only unsolvable terms (in the sense of (Wadsworth, 1976)) have a trivial functional characterization (corresponding to "undefined"). Our extension is characterized by the introduction of the "universal type" ω and by the operation of sequencing in type construction, which allows a refinement of the categories into which terms are classified by types. In (Coppo *et al.*, 1979a) types are built from ω and a set of basic elements $\{\xi_1, \xi_2, \ldots\}$ (which are left uninterpreted as in Curry's basic functionality theory) by an operator of composition which generalizes the object F of Curry's theory. In fact, while F builds a new type from two types, this operator (let us call it F' for the moment) builds a new type from n + 1 types ($n \geq 1$). I.e., if τ, $\sigma_1, \ldots, \sigma_n$ are types ($n \geq 1$) then

(1) $F'[\sigma_1, \ldots, \sigma_n]\,\tau$

is a type. $F'[\sigma_1, \ldots, \sigma_n]\,\tau$ is the type of a term which maps terms which have *all* types $\sigma_1, \ldots, \sigma_n$ into terms of type τ. $[\sigma_1, \ldots, \sigma_n]$ is called a *sequence*. We shall abbreviate this notation by omitting F' (without danger of confusion), i.e. we will write $[\sigma_1, \ldots, \sigma_n]\,\tau$ instead of (1). Sequences, as types, can be assigned to terms. This will be done in the formal system by a suitable rule (Si) which allows to deduce $[\sigma_1, \ldots, \sigma_n]$ X from σ_1 X,..., σ_n X (this suggests the possible interpretation of sequences as set intersections). Let's observe that the assignment of a sequence to a term is more general than the assignment of a type to a term.

It is now natural to ask if the notion of principal type scheme can be defined also in the extended theory (we will consider sequences instead of types since, as pointed out before, they are more general). As it will turn out from the discussion of sections 3 and 4, the principal type scheme of a term, in the classical sense of (Curry, 1969), (Hindley, 1969), does not exist in our theory. This is due essentially to two facts:

1) in a deduction, different types can be assigned to the same component of a given term. Then the structure of deductions does not simply correspond, as in Curry's theory, to the structure of terms but it is more "ramified" and there is no limit to its complexity. With the only operation of substitution we cannot obtain, for a given term, all types

whose deductions are more complex than the one of the type on which we are doing the substitution. For example, for any $n \geq 1$, the types $\tau_n \equiv [[[\phi_n] \phi_n, ..., [\phi_1]\phi_1]\phi_0] \phi_0$ are deducible for $\lambda x.x(\lambda y.y)$. But, for any $i > 0$, τ_i can be obtained by substitution (modulo a suitable rule of normalization) only from all τ_j such that $j \geq i$ and so none of them can be obtained from τ_1 (except, obviously, itself). We notice that τ_1 is the principal type scheme of $\lambda x.x(\lambda y.y)$ in the classical theory.

2) in our theory also terms without normal form or, in particular, terms which have an "infinite" normal form (in the sense of (Lévy, 1978)) have types. But, as it will be proved, the set of all functional characters of these terms must carry an infinite amount of information and it cannot be represented in a finite way.

We shall approach these problems in two steps.

First we will consider, in section 3, a particular class of terms, i.e. terms of the λ-Ω-calculus in Ω-normal form. We will prove that, in this case, the difficulty presented in point 1 can be overcome if we introduce, besides substitution, the new (context-dependent) operation of expansion. In this case there exists, for each Ω-normal form A, an infinite set of sequence schemes (we shall call them *ground* ones) such that all sequence schemes which can be assigned to A can be generated from an arbitrary ground one by a finite number of expansions and substitutions. Among the ground sequence schemes for an arbitrary Ω-normal form A, there will turn out (in section 4) to exist a sequence scheme, that contains only one type scheme (say π), which is the simplest one, in the sense that it is the shortest and it is associated to the simplest deduction. We will call π the *principal type scheme* of A.

Then, we shall consider the existence of principal type schemes for arbitrary terms. We shall do this through the notion of approximant of a term (in the classical sense of (Wadsworth, 1976)). It will turn out that terms with a finite set of approximants have finite principal type schemes while terms with an infinite set of approximants have "infinite" principal type schemes. This last result, in particular, will be achieved through a classical algebraic construction.

Lastly, in section 5 it will be shown that it is possible to define a λ-calculus model (according to (Hindley and Longo, 1980)) whose domain is the power set of the set of type schemes. We obtain it by introducing a suitable notion of application suggested by type assignment rules and by assuming as value of a term the set of its type schemes. This notion of application, although obtained in an independent way, is a specialization of the one given by Plotkin (1972).

The equality relation between terms introduced by such a model turns out to be equivalent to the one introduced by the well - known models $P\omega$ (Scott, 1976), $T\omega$ (Barendregt and Longo, 1980), and by Lévy's syntactic model (Lévy, 1978).

In the present paper, section 2 is an introductory one. Sections 3 and 4 are essentially by M. Coppo and M. Dezani-Ciancaglini, while section 5 is essentially by B. Venneri.

2. THE UNIVERSE OF TYPE SCHEMES.

In this section we will give a short but complete survey of the theory developed in (Coppo *et al.*, 1979a) and we will introduce some basic definitions and properties about type schemes. As pointed out in the introduction we need, beyond the notion of type, also the one of sequence. Types, as sequences, are built from a set of basic types which contains, beside other arbitrary elements, the type ω .

DEFINITION 1. The *sets of types and sequences* are so defined:

i) each basic type is a type

ii) if $\sigma_1,\dots,\sigma_n$ $(n > 0)$ are types then $[\sigma_1,\dots,\sigma_n]$ is a sequence

iii) if $[\sigma_1,\dots,\sigma_n]$ is a sequence and τ is a type then $[\sigma_1,\dots,\sigma_n]\tau$ is a type. □

We shall indicate types with the greek letters σ,τ,μ,ν,ρ. ξ will denote a basic type.

We say that a sequence $[\sigma_1,\dots,\sigma_n]$ contains a type τ iff for some i $(1 \le i \le n)$ $\sigma_i \equiv \tau$.

We use $[\bar\sigma]$ to represent a sequence $[\sigma_1,\dots,\sigma_n]$ $(n \geq 1)$ (while $[\sigma]$ is a sequence which contains only type σ).

$[\bar\sigma_1, \bar\sigma_2]$ denotes the sequence obtained by concatenating $[\bar\sigma_1]$ and $[\bar\sigma_2]$.

The notion of subtype of a given type is obtained in a straightforward way from Definition 1. A subtype of a sequence is a subtype of one of the types contained in it.

It is useful, for the development of this paper, to consider the "natural deduction" formulation of our system. Moreover we need consider sequences independently of the order in which types are written. This can be obtained by introducing the following equivalence relation between types and sequences.

DEFINITION 2. The relation $\sim$ is so defined:

i) $\xi \sim \xi$ for any basic type ξ

ii) $[\sigma_1,\dots, \sigma_i, \sigma_{i+1}, \dots, \sigma_n] \sim [\sigma'_1,\dots, \sigma'_{i+1}, \sigma'_i,\dots, \sigma'_n]$ iff $\sigma_j \sim \sigma'_j$ for $1 \le j \le n$

iii) $[\bar{\sigma}]\tau \sim [\bar{\sigma}']\tau'$ iff $[\bar{\sigma}] \sim [\bar{\sigma}']$ and $\tau \sim \tau'$. □

We notice that $\sim$ does not include elimination of duplicates, for example $[\sigma,\sigma] \not\sim [\sigma]$.

In the present paper we will consider always types and sequences modulo $\sim$. Also the identity relation $\equiv$ between types and sequences is considered, from now on, modulo $\sim$.

In (Coppo *et al.*, 1979a) it has been proved that it is possible to define an operation of *normalization* of types whose purpose is to eliminate all types which are inessential to express the functional properties of terms.

Let's observe that, for all types σ, $\sigma \equiv [\bar{\sigma}_1] \ldots [\bar{\sigma}_n]\,\xi$ where $n \geq 0$ and ξ is a basic type. We say that σ is a *tail-proper* type iff ξ is different from ω.

DEFINITION 3. If σ is any type, we define its *normalized form* σ_N the type obtained by applying to σ, successively, the two following transformations:

1) replace with ω all occurrences of non tail-proper types in σ
2) delete all occurrences of ω contained in sequences which contain at least one type different from ω and replace with $[\omega]$ all sequences which contain only occurrences of ω. □

For example, if $\sigma \equiv [\omega,[\xi]\xi][[[\xi]\omega]\omega]\xi$ then $\sigma_N \equiv [[\xi]\xi][\omega]\xi$.

If $[\sigma_1,\ldots,\sigma_n]$ is a sequence, $[\sigma_1,\ldots,\sigma_n]_N$ is obtained by applying the transformation 2 of Definition 3 to $[\sigma_{1N},\ldots,\sigma_{nN}]$.

Let Δ denote a type or a sequence. We will say that Δ is *normalized* iff $\Delta \equiv \Delta_N$. It is easy to verify that, for each Δ, Δ_N is unique.

The intuitive meaning of transformations 1 and 2 of Definition 3 can be explained by the universal character of ω.

The assignment of types to terms is performed by a natural deduction system, as introduced in (Coppo *et al.*, 1979a). Let's define a *statement* as an expression ΔX where Δ represents a type or a sequence and X is a term. It means that the type (or sequence) Δ is assigned to X. According to (Curry *et al.*, 1958) X is referred as the *subject* of the statement and Δ as its *predicate*.

A *deduction* is a set of statements arranged as a tree according to a set of inference rules defined below. The statements in the leaves of this tree are the *premises* for the deduction of the statement which is in the root of the tree. We force the *premises* of a deduction to be statements whose subjects are variables and whose *predicates are normalized types different from* ω.

The inference rules of our system are the following:

(ωi) $$\frac{}{\omega\ X} \quad \text{for all terms X}$$

(Si) If $\sigma_i \not\sim \omega$ for $1 \leq i \leq n$ then: $\dfrac{\sigma_1\ X \ \dots\ \sigma_n\ X}{[\sigma_1,\dots,\ \sigma_n]\ X}$

Else: $$\frac{\omega X}{[\omega]\ X}$$

(F'e) $$\frac{[\bar{\sigma}]\tau\ X \quad [\bar{\sigma}]\ Y}{\tau XY}$$

(F'i) Let X be a term, τ a type different from ω and x a variable. If $\sigma_1 x,\dots,\ \sigma_n\ x$ are all and only the premises whose subject is x and which have been used to deduce τ X then:

$$\begin{matrix} [\sigma_1 x] & . & [\sigma_n x] \\ & \vdots & \\ & \dfrac{\tau\ X}{[\sigma_1,\dots,\sigma_n]\ \tau\,\lambda x.X} & \end{matrix}$$

where the square brackets indicate here that the premises $\sigma_1 x,\dots$ $\dots,\ \sigma_n$ x must be cancelled.

Else if in the deduction of τ X there is no premise whose subject is x then:

$$\frac{\tau X}{[\omega]\tau\lambda x.X}$$

Rule ωi characterizes the universality of ω.
Rule Si allows the assignment of sequences to terms.
Rules F'e and F'i generalize in an obvious way the rules Fe and Fi of the natural deduction formulation of Curry's theory for λ-calculus. Since we consider sequences modulo $\sim$, it does not matter, in rule F'i, the order in which $\sigma_1,\dots,\sigma_n$ are written to build the sequence $[\sigma_1,\dots,\sigma_n]$.

Let's notice that, in the present formulation, our system is not really a generalization of Curry's functionality theory since, with this form of rule F'i, we cannot deduce $[\sigma]\tau\lambda x.X$ for $\sigma \not\sim \omega$ if x does not occur in X. However there is no difficulty in modifying rule F'i (similarly as in (Curry *et al.*, 1972, p. 309)) to include completely Curry's theory (this is also obtained in the sequent formulation of (Coppo *et al.*, 1979a)) but this would imply a more complex definition of the operation of expansion.

Let's notice, however, that, from a semantic point of view, $[\omega]\tau$ is a "stronger" type than $[\sigma]\tau$ since it means that $(\lambda x.X)Y$ has a type τ *for all* Y and not only for the ones of type σ.

The relation $\sim$ between sequences induces in a natural way an equivalence relation between deductions which differ only for the order of the subtrees whose roots are $\sigma_1 X, \ldots, \sigma_n X$ in some application of rule Si. We will consider deductions modulo this equivalence.

We could adjoin a rule Se, complementary to Si, for deducing a type from a sequence but we don't need it. This rule could be proved as a metatheorem since it is clear that if we have a deduction of $[\sigma_1, \ldots, \sigma_n]$ X we must have also a deduction of σ_i X for all i $(1 \leq i \leq n)$.

We define a *basis* B to be a set of statements σx where σ is a type and x a variable and we write $B \vdash \Delta X$ if there is a deduction of ΔX in which the uncancelled premises are all and only the statements in B. Let's notice that the definition of basis B does not imply that there exists a deduction whose premises are the statements of B. We will need this more general notion of basis in section 3. If B is the empty basis (let's denote it with Φ) we write simply $\vdash \Delta X$. Let's notice that, if X is closed, B is always empty, while the viceversa is not always true (we have, for example, $\vdash \omega X$ for all non-closed X).

To simplify notations we shall write in a basis $\{\bar{\sigma} x\}$ instead of $\{\sigma_1 x, \ldots$ $\ldots, \sigma_n x\}$ where $\bar{\sigma} \equiv \sigma_1, \ldots, \sigma_n$. Then we can write $B = B' \cup \{\bar{\sigma} x\}$ and $B = B' - \{\bar{\sigma} x\}$ (to denote set-difference).

The notion of normalization generalizes in an obvious way to bases. If B is a basis, B_N is the basis obtained from B by normalizing first all predicates of the statements of B and then by eliminating the statements ωx for all variables x. B is normalized iff $B = B_N$.

As pointed out in (Coppo *et al.*, 1979a), in this system if $B \vdash \Delta X$ then B and Δ are normalized. This is due to the fact that we force the premises to have normalized predicates and we don't allow, in rules F'i and Si, to build any non-normalized type or sequence.

Let $< B; [\bar{\tau}] >$ denote a pair basis, sequence. We introduce the symbol $\Rightarrow$ to denote the operation of normalization, i.e. $\Delta \Rightarrow \Delta_N$, $B \Rightarrow B_N$ and $< B; [\bar{\tau}] > \Rightarrow < B_N ; [\bar{\tau}]_N >$.

The principal results which can be proved (Coppo *et al.*, 1979a) are the following:

THEOREM 1. Any two β-convertible terms have the same set of types for any basis. □

As observed in (Coppo *et al.*, 1979a) this theorem is not true, in general, for β - η -convertibility.

THEOREM 2. A term X has a head normal form iff there are some basis B and type $\tau \not\equiv \omega$ such that $B \vdash \tau X$. □

As a consequence of this theorem we have that an unsolvable term X can have only type ω. Moreover, in this case, we must have $B = \Phi$.

THEOREM 3. It is possible to give an (effective) characterization of all types τ and bases B such that, if X is an arbitrary term, $B \vdash \tau X$ implies that X has a β -normal form (and viceversa). □

The *set T of type and sequence schemes* is built according to the rules given for types but starting from a set of type variables $\{\psi_1, \psi_2, \ldots\}$ (in addition to the set of basic types). Obviously T contains properly the set of types and sequences. A *basis scheme* will contain statements whose predicates are type schemes and whose subjects are variables. Rules ωi, F'i, F'e, Si and all previous definitions and results generalize, in an obvious way, to type, sequence and basis schemes.

We say that two type schemes σ, τ are *disjoint* iff no type variable occurs both in σ and τ. The notion of disjunction can be extended to sequence, basis schemes and pairs in an obvious way.

A *substitution* s is a set of pairs (ψ_i, μ_i) for $1 \leq i \leq n$ where ψ_i are distinct type variables and μ_i are normalized type schemes. If τ is a type scheme, s(τ) is the type scheme obtained from τ by replacing simultaneously each occurrence of ψ_i in τ by μ_i for $1 \leq i \leq n$. A type scheme σ is an instance of a type scheme τ iff there is a substitution s such that $\sigma \equiv s(\tau)$.

If s_1, s_2 are two substitutions then their *composition* $s_1 \circ s_2$ is defined, as usual, as $s_1 \circ s_2(\sigma) \equiv s_1(s_2(\sigma))$.

We can extend the notion of substitution also to sequences (basis) schemes writing $[\bar{\sigma}] \equiv s([\bar{\tau}])$ $(B_1 = s(B_2))$ if $[\bar{\sigma}]$ (B_1) is obtained from $[\bar{\tau}]$ (B_2) by applying s to all type schemes in $[\bar{\tau}]$ (B_2). Analogously we have $s(\langle B;[\bar{\tau}]\rangle) = \langle s(B); s([\bar{\tau}])\rangle$.

A type scheme σ is a *trivial variant* of a type scheme τ iff $\sigma \equiv s(\tau)$ and s is a substitution of the shape $s = \{(\psi_i, \zeta_i) \mid 1 \leq i \leq n\}$ where ζ_i are distinct type variables and $\psi_1, \ldots, \psi_n$ include all the variables which occur in τ, that is σ and τ are equal modulo a re-naming of type variables. It is clear that if σ is a trivial variant of τ, then also τ is a trivial variant of σ, and therefore we can say that σ is equivalent to τ. Through the paper we shall always consider type, sequence and basis schemes modulo this equivalence.

We will say that a substitution s is an *ω-substitution* iff s = $=\{ (\psi_i, \omega) \mid 1 \leq i \leq n \}$. s is a *proper* substitution iff s = $\{ (\psi_i, \mu_i) \mid 1 \leq i \leq n \}$ and $\mu_i \not\equiv \omega$ for $1 \leq i \leq n$. Let's observe that, if s is a proper substitution and τ is a normalized type scheme, $s(\tau)$ is still normalized. This is no more true in the general case of non-proper substitutions.

In section 3 we shall use the following technical lemma, whose proof follows easily from the definitions of normalization and substitution.

LEMMA 1. If $< B;[\bar{\tau}] >$ is any pair, s is a substitution and $s(< B_N;[\bar{\tau}]_N >) \Rightarrow <B';[\bar{\tau}']>$ then $s(<B;[\bar{\tau}]>) \Rightarrow <B';[\bar{\tau}']>$. □

The same property holds, as a particular case, also for type, sequence and basis schemes.

The meaning of this Lemma is that, in a chain of substitutions and normalizations, we can perform a unique normalization after all substitutions.

As pointed out in the introduction the notion of substitution is not enough for our purpose.

We define first the notion of nucleus of a multiset of type schemes. As usual, a multiset is like a set except that it can have repetitions of identical elements. The definitions of submultiset, multi-union etc. are obvious. We introduce the notion of multiset since we will need to distinguish between the different occurrences of the same type.

DEFINITION 4. Let $C = \tau_1, \ldots, \tau_n$ be a multiset of type schemes. Let $\rho_1, \ldots, \rho_m$ $(m > 0)$ be *occurrences* of subtypes of $\tau_1, \ldots, \tau_n$ and let E denote the *multiset* $\rho_1, \ldots, \rho_m$. E is a nucleus of C iff:

i) $\rho_i \not\equiv \omega$ for $1 \leq i \leq n$

ii) for all $1 \leq i \neq j \leq m$ the occurrence ρ_i is not a part of the occurrence ρ_j in C

iii) all type variables which occur in $\rho_1, \ldots, \rho_m$ don't occur in C outside $\rho_1, \ldots, \rho_m$

iv) for all $1 \leq i \leq m$ either $\rho_i \equiv \tau_j$ or ρ_i is contained in some sequence which occurs in τ_j for some $j (1 \leq j \leq n)$. □

Example 1. Let

$$C = [\phi][\underline{\phi'}]\psi, \underline{[\psi']\ \phi'};[\phi, \underline{\psi'}]\psi \ .$$

The multiset E_1 of subtypes underlined with a continous line is a nucleus of C, but the multiset E_2 of subtypes underlined with a dashed line is not. In fact E_2 does not satisfy:

– point ii) since the first occurrence of ϕ is a subtype of $[\phi][\phi']\psi$

– point iii) since ψ' occurs in E_2 and outside E_2

point iv), since the second occurrence of ψ is neither the occurrence of a type which belongs to C nor the occurrence of a type which is contained in a sequence. □

Let's observe that, for all C, C is a nucleus of itself.

DEFINITION 5. A multiset of type schemes C' is an *immediate expansion* of a multiset C iff there are a nucleus E of C and an integer n ($n \geq 1$) such that C' can be obtained from C in the following way:
Let $\phi_1, \ldots, \phi_m$ be all the variables which occur in E and $\phi_j^{(i)}$ ($1 \leq j \leq m$; $1 \leq i \leq n$) a new set of variables disjoint from all variables in C. For each τ which belongs to E, replace the occurrence τ in C by $\tau_1, \ldots, \tau_n$ where $\tau_i \equiv s_i(\tau)$ and $s_i = \{(\phi_j, \phi_j^{(i)}) \mid 1 \leq j \leq m\}$ for $1 \leq i \leq n$.
We will say that E is the *nucleus* expanded. □

Example 2. An immediate expansion of C as defined in example 1 (if E_1 is the nucleus expanded and $n = 2$) gives

$$C' = [\phi][\phi'_1, \phi'_2]\,\psi, [\psi'_1]\phi'_1, [\psi'_2]\phi'_2, [\phi, \psi'_1, \psi'_2]\psi. \qquad \square$$

Let's notice that condition iv) of Definition 4 is essential to assure that the result of an immediate expansion of a multiset of type schemes is in its turn a multiset of type schemes according to Definition 1.
We say that multiset C' is an expansion of another multiset C iff we can obtain C' from C by means of a finite number (possibly none) of successive immediate expansions.
If $[\bar{\tau}]$ is a sequence scheme and B is a basis scheme, a nucleus of $\langle B;[\bar{\tau}]\rangle$ and an expansion of $\langle B; [\bar{\tau}]\rangle$ are defined on the multiset of type schemes which are contained in $[\bar{\tau}]$ and which are predicates in B.
Obviously, if $\sigma x \in B$, σ belongs to the nucleus to be expanded and σ would be replaced by $\sigma_1, \ldots, \sigma_n$, then the result of this expansion is obtained by replacing $\sigma_1 x, \ldots, \sigma_n x$ for σx in B.
$\langle B; [\bar{\tau}]\rangle \rightarrow \langle B'; [\bar{\sigma}]\rangle$ and $\langle B; [\bar{\tau}]\rangle \overset{*}{\rightarrow} \langle B''; [\bar{\mu}]\rangle$ will mean respectively that $\langle B'; [\bar{\sigma}]\rangle$ is an immediate expansion of $\langle B; [\bar{\tau}]\rangle$ and $\langle B''; [\bar{\mu}]\rangle$ is an expansion of $\langle B; [\bar{\tau}]\rangle$. Let's notice that, if B and $[\bar{\tau}]$ are normalized, then B'', $[\bar{\mu}]$ are normalized too.
It is clear that $\sim$ commutes with normalization and substitution, i.e. $\sigma \sim \tau$ implies both $\sigma_N \sim \tau_N$ and $s(\sigma) \sim s(\tau)$ (also for sequences, bases and pairs). Moreover this relation commutes also with expansion.
We need consider only deductions of sequence schemes owing to the following property :

PROPERTY 1. $B \vdash [\tau]X$ iff $B \vdash \tau X$. □

Lastly we can give the definition of complete sets of pairs. If S is a complete set for a term X then, starting from the pairs of S , we can generate (through expansions and substitution) all and only the pairs $<B; [\bar{\tau}]>$ such that $B \vdash [\bar{\tau}]X$ (modulo normalization).

DEFINITION 6. Let X be a term and S a set of pairs basis scheme, sequence scheme. We say that S is *complete* for X iff for all pairs $<B^*;[\bar{\sigma}]>$:

$B^* \vdash [\bar{\sigma}]$ X iff there exist $<B; [\bar{\tau}]> \in S$, $<B'; [\bar{\nu}]>$ and a substitution s such that $<B; [\bar{\tau}]> \overset{*}{\rightarrow} <B'; [\bar{\nu}]>$ and $s(<B'; [\bar{\nu}]>) \Rightarrow <B^*; [\bar{\sigma}]>$. □

In the particular case $S = \{ <B; [\bar{\tau}]> \}$, we will say that the pair $<B; [\bar{\tau}]>$ is complete for X.

3. COMPLETE PAIRS FOR Ω-NORMAL FORMS

In this section we shall prove some results about the existence of complete pairs for a particular class of terms, i.e. terms in Ω-normal form in that extension of λ-calculus usually referred as λ-Ω-calculus.

The λ-Ω-calculus is defined (Lévy, 1978) by adjoining to the set of terms of λ-calculus a new constant Ω, whose reduction rules are: for all terms X and variables x, ΩX and $\lambda x.\Omega$ are Ω-redexes and both reduce to Ω

A Ω-normal form is a β-normal form which does not contain Ω-redexes. Let N be the set of Ω-normal forms. As in (Lévy, 1978) N can be defined, inductively, in the following way:

i) Ω and all formal variables belong to N

ii) if $A_1, \ldots, A_p \in N$ $(p \geq 0)$, then $\lambda x_1 \ldots x_n .zA_1 \ldots A_p \in N$, where $n \geq 0$ and $z, x_1, \ldots, x_n$ are variables.

We define now, for each Ω-normal form A, a set of (normalized) pairs $<B; [\bar{\tau}]>$ that we will prove to be complete for A. We will call them *ground pairs* .

DEFINITION 7. If A is a Ω-normal form, a pair $<B; [\bar{\tau}]>$ is a *ground pair* for A iff $B, [\bar{\tau}]$ satisfy the following conditions:

1) If $[\bar{\tau}] \equiv [\tau_1, \ldots, \tau_m]$ and $m > 1$ then $\tau_i \not\leq \omega$, $B = \bigcup_{i=1}^{m} B_i$ where $<B_i; [\tau_i]>, <B_j; [\tau_j]>$ are pairwise disjoint for $1 \leq i \neq j \leq m$, and $<B_i; [\tau_i]>$ are ground pairs for A $(1 \leq i \leq m)$.

2) If $[\bar{\tau}] \equiv [\tau]$ then:

i) if $A \equiv \Omega$ then $B = \Phi$ and $\tau \equiv \omega$.

ii) if $A \equiv x$ then $B = \{\psi x\}$ and $\tau \equiv \psi$ where ψ is a type variable.

iii) if $A \equiv \lambda x.A'$ then $\tau \equiv [\bar{\tau}_1]\tau_2$ where $[\bar{\tau}_1] \not\leq [\omega]$ implies that

$< B \cup \{\bar\tau_1 x\}; [\tau_2] >$ is a ground pair for A' and $[\bar\tau_1] \equiv [\omega]$ implies that $< B; [\tau_2] >$ is a ground pair for A'.

iv) if $A \equiv xA_1 \ldots A_p$ $(p \geq 1)$ then τ is a type variable ψ and $B = \bigcup_{i=1}^{p} B_i \cup \{[\bar\sigma_1] \ldots [\bar\sigma_p]\, \psi x\}$ where:

a) ψ does not occur in $B_1, \ldots, B_p$

b) $<B_i ; [\bar\sigma_i]>$, $< B_{i'} ; [\bar\sigma_{i'}] >$ are pairwise disjoint for $1 \leq i \neq i' \leq p$

c) $<B_i ; [\bar\sigma_i]>$ are ground pairs for A_i $(1 \leq i \leq p)$. □

It is easy to see that, if $< B ; [\bar\tau] >$ is a ground pair for A then there is a (unique) deduction of $[\bar\tau]$ A from the statements in B, that we call a *ground deduction* for A. For each Ω-normal form, moreover, there are infinite ground pairs, except in the case of Ω, whose only ground pair is $< \Phi ; [\omega] >$.

Example 3.

$$< \Phi; [[[\omega]\ [[\phi_0]\phi_0, [\phi_1]\phi_1]\ \psi]\,\psi]>$$

is a ground pair for $A \equiv \lambda x.x\, \Omega\, (\lambda y.y)$. In fact, $< \Phi ; [\omega]>$ is a ground pair for Ω and $<\Phi;[[\phi_0]\phi_0, [\phi_1]\phi_1] >$ is a ground pair for $\lambda y.y$. □

To shorten notation we understand, in the following proofs, that $\{\bar\tau x\}$ represents the empty set when $\bar\tau \equiv \omega$.

We shall first prove (in Lemmas 2 and 4) that, for an arbitrary Ω-normal form A, if $B \vdash [\bar\sigma]$ A then $<B ; [\bar\sigma] >$ can be obtained from any arbitrary ground pair for A through expansions, substitution and normalization.

In Lemma 2, in particular, we will prove that for each $<B ; [\bar\sigma] >$ such that $B \vdash [\bar\sigma]$ A there exists a ground pair $<B^*; [\bar\tau] >$ for A such that $<B ; [\bar\sigma] >$ is an instance of $<B^*; [\bar\tau] >$ (modulo normalization). Lemma 3 shows that also the viceversa holds true. In Lemma 4 we will prove that all ground pairs for A can be obtained from any arbitrary ground one through expansions, substitution and normalization.

LEMMA 2. If $A \in N$ and $B \vdash [\bar\sigma]$ A then there are B^*, $[\bar\tau]$ and s such that $< B^*; [\bar\tau] >$ is a ground pair for A and $s(<B^*;[\bar\tau]>) \Rightarrow <B; [\bar\sigma] >$.

Proof. We will prove something more, i.e. that if $[\bar\sigma] \equiv [\sigma_1, \ldots, \sigma_m]$ then $[\bar\tau] \equiv [\tau_1, \ldots, \tau_m]$, i.e. that $[\bar\sigma]$ and $[\bar\tau]$ contain the same number (m) of types. This is useful (with m = 1) in the proof of case 1 of the inductive step.

We first prove that, for an arbitrary A, if this Lemma is true for all sequence schemes which contain only one type scheme then it is true for arbitrary sequence schemes (let's call it Property S).

Proof of Property S. If $[\bar{\sigma}] \equiv [\sigma_1,\ldots,\sigma_m]$ with $m > 1$ then $\sigma_i \not\equiv \omega$ and (by rule Si) there exist $B_1,\ldots, B_m$ such that $B = \bigcup_{i=1}^{m} B_i$, and $B_i \vdash \sigma_i A$. Then, by Property 1, $B_i \vdash [\sigma_i] A$ for $1 \leq i \leq m$.
By hypotesis there are ground pairs $< B_i^* ; [\tau_i] >$ for A and substitutions s_i such that $s_i(< B_i^* ; [\tau_i] >) \Rightarrow < B_i ; [\sigma_i] >$ $(1 \leq i \leq m)$. We can choose (via the notion of trivial variant) $< B_i^* ; [\tau_i] >$ such that they are pairwise disjoint for all i $(1 \leq i \leq m)$. Then, by Definition 7, if $B^* = \bigcup_{i=1}^{m} B_i^*$, $< B^*; [\tau_1,\ldots, \tau_m] >$ is a ground pair for A and it is easy to verify that $s_1 \circ \ldots \circ s_m(< B^*;[\tau_1,\ldots, \tau_m]>) \Rightarrow < B ; [\bar{\sigma}] >$.

Proof of the Lemma. We can suppose $[\bar{\sigma}] \not\equiv [\omega]$ for, in this case, B is empty and we can choose arbitrarily a ground pair $< B^*; [\bar{\tau}] >$ for A and s as the substitution that replaces each type variable with ω. Now we prove the Lemma by induction on the structure of A. Thanks to Property S we need consider in each step only the case $[\bar{\sigma}] \equiv [\sigma]$.

First step. The case $A \equiv \Omega$ is trivial since we must have $B = \Phi$ and $[\sigma] \equiv [\omega]$. If $A \equiv x$ we must have $B = \{\sigma x\}$. Then the desired ground pair is $< \psi x; [\psi] >$ where ψ is a type variable and $s = \{(\psi,\sigma)\}$.

Inductive step. By cases:

1) $A \equiv \lambda x.A'$. In this case, we must have $\sigma \equiv [\bar{\sigma}_1]\sigma_2$ where $\sigma_2 \not\equiv \omega$. By Property 1 $B \vdash [\bar{\sigma}_1]\sigma_2\, \lambda x.\, A'$ and $B \cup \{\bar{\sigma}_1\, x\} \vdash [\sigma_2]A'$. By inductive hypothesis, there exists a ground pair for A', $< B^* \cup \{\bar{\tau}_1 x\} ; [\tau_2] >$ such that B^* does not contain type assignments to x and

$$s(< B^* \cup \{\bar{\tau}_1 x\} ; [\tau_2] >) \Rightarrow < B \cup \{\bar{\sigma}_1 x\} ; [\sigma_2] >$$

(for some substitution s), which implies that $< B^*; [[\bar{\tau}_1]\tau_2] >$ is a ground pair for A and

$$s(< B^*; [[\bar{\tau}_1]\tau_2] >) \Rightarrow < B; [[\bar{\sigma}_1]\sigma_2] > .$$

2) $A \equiv xA_1 \ldots A_p$. In this case we must have $B = \bigcup_{i=1}^{p} B_i \cup \{[\bar{\sigma}_1]\ldots[\bar{\sigma}_p]\sigma x\}$ and $B_i \vdash [\bar{\sigma}_i]\, A_i$ $(1 \leq i \leq p)$. By inductive hypothesis and Property S there are $B_1^*,\ldots, B_p^*, [\bar{\tau}_1],\ldots, [\bar{\tau}_p]$ such that $< B_i^* ; [\bar{\tau}_i] >$ are ground pairs for A_i and $s_i (< B_i^*; [\bar{\tau}_i] >) \Rightarrow < B_i ; [\bar{\sigma}_i] >$ for some suitable s_i $(1 \leq i \leq p)$. We can assume that $< B_i^*; [\bar{\tau}_i] >$ and $< B_{i'}^*; [\bar{\tau}_{i'}] >$ are pairwise disjoint for $1 \leq i \neq i' \leq p$. Therefore if ψ is a type variable which does not occur in $B_1^*,\ldots, B_p^*, [\bar{\tau}_1],\ldots, [\bar{\tau}_p]$ we can choose:

$$B^* = \bigcup_{i=1}^{p} B_i^* \cup \{[\bar{\tau}_1]\ldots [\bar{\tau}_p]\, \psi x \} \quad \text{and} \quad \tau \equiv \psi .$$

By Definition 7, $< B^*; [\psi] >$ is a ground pair for A. Moreover it is easy to see that $s_1 \circ \ldots \circ s_p \circ \{(\psi,\sigma)\}(< B^*; [\psi] >) \Rightarrow < B; [\sigma] >$. □

Lemma 3 states that the viceversa of Lemma 2 is also true.

LEMMA 3. If $B^* \vdash [\bar{\tau}]$ A and, for some substitution s, s($\langle B^*; [\bar{\tau}] \rangle$)$\Rightarrow$ $\langle B; [\bar{\sigma}] \rangle$, then $B \vdash [\bar{\sigma}]$ A.

Proof. Let's first observe that, if s is any substitution, there always exist a proper substitution s' and an ω-substitution s_ω such that $s = s_\omega \circ s'$. Moreover, if s is proper, since B^* and $[\bar{\tau}]$ are normalized by hypothesis, s($\langle B^*; [\bar{\tau}] \rangle$) is normalized and it is easy to prove (via the notion of instance of a deduction (Hindley, 1969)) that s(B^*) $\vdash$ s($[\bar{\tau}]$) A. So it's enough to prove Lemma 3 only in the case of s being an ω-substitution s_ω. The proof, in this case, can be performed with a technique similar to the one used to prove Lemma 2; i.e. proving that if the Lemma is true for $[\bar{\tau}] \equiv [\tau]$ then it is true also for arbitrary sequence schemes and then using this property in a standard proof by structural induction on A. □

LEMMA 4. If $\langle B; [\bar{\tau}] \rangle$ and $\langle B^*; [\bar{\sigma}] \rangle$ are any two ground pairs for A then there are B', $[\bar{\nu}]$ such that $\langle B; [\bar{\tau}] \rangle \overset{*}{\rightarrow} \langle B'; [\bar{\nu}] \rangle$ and s($\langle B'; [\bar{\nu}] \rangle$) $\Rightarrow \langle B^*; [\bar{\sigma}] \rangle$ for some substitution s.

Proof. We will prove something more, i.e. that if both $[\bar{\tau}]$ and $[\bar{\sigma}]$ contain only one type scheme then also $[\bar{\nu}]$ will contain only one type scheme. This is necessary in the proof of case 1 in the inductive step. We first prove that, for an arbitrary A, if we suppose the Lemma true in the case that both $[\bar{\tau}]$ and $[\bar{\sigma}]$ contain only one type scheme it is true also in the general case (let's call it Property S').

Proof of Property S'. By cases.

Case 1. If $[\bar{\tau}] \equiv [\tau]$ and $[\bar{\sigma}] \equiv [\sigma_1, \ldots, \sigma_m]$ with $m > 1$ we have by Definition 7 that $\sigma_i \not\equiv \omega, B^* = \bigcup_{i=1}^{m} B_i^*$ where $\langle B_i^*; [\sigma_i] \rangle$ are (pairwise disjoint) ground pairs for A ($1 \leq i \leq m$). Then let's first do an immediate expansion of $\langle B; [\tau] \rangle$ according to the nucleus defined by all type schemes in $\langle B; [\tau] \rangle$. We obtain so $\langle B; [\tau] \rangle \rightarrow \langle B_1 \cup \ldots \cup B_m; [\tau_1, \ldots, \tau_m] \rangle$, where $\langle B_i; [\tau_i] \rangle$ are trivial variants of $\langle B; [\tau] \rangle$. Moreover, by hypothesis, we have that, for all i ($1 \leq i \leq m$), there exists $\langle B_i'; [\nu_i] \rangle$ such that $\langle B_i; [\tau_i] \rangle \overset{*}{\rightarrow} \langle B_i'; [\nu_i] \rangle$ and $s_i(\langle B_i'; [\nu_i] \rangle) \Rightarrow \langle B_i^*; [\sigma_i] \rangle$ for some substitution s_i. Since each nucleus of $\langle B_i; [\tau_i] \rangle$ is also a nucleus of $\langle B_1 \cup \ldots \cup B_m; [\tau_1, \ldots, \tau_m] \rangle$ and this property remains true also through successive expansions (if we take care that all variables introduced by any new expansion do not occur in the current multiset of type schemes) we have immediately:

$$\langle B; [\tau] \rangle \rightarrow \langle B_1 \cup \ldots \cup B_m; [\tau_1, \ldots, \tau_m] \rangle \overset{*}{\rightarrow} \langle B_1' \cup \ldots \cup B_m'; [\nu_1, \ldots, \nu_m] \rangle .$$

Moreover

$$s_1 \circ \ldots \circ s_m(\langle B_1' \cup \ldots \cup B_m'; [\nu_1, \ldots, \nu_m] \rangle) \Rightarrow \langle B^*; [\bar{\sigma}] \rangle .$$

Case 2. If $_n[\bar{\tau}] \equiv [\tau_1, \ldots, \tau_n]$ with $n > 1$ we have again, from Definition 7, $\tau_i \not\equiv \omega$, $B = \bigcup_{i=1}^{n} B_i$, where $<B_i; [\tau_i]>$ are (pairwise disjoint) ground pairs for A $(1 \leq i \leq n)$.
Let's choose an arbitrary value of $i(1 \leq i \leq n)$. From case 1, there exist $B'', [\bar{\mu}]$ such that $<B_i; [\tau_i]> \xrightarrow{*} <B''; [\bar{\mu}]>$ and $s'(<B''; [\bar{\mu}]>) \Rightarrow <B^*; [\bar{\sigma}]>$ for some substitution s'. Then we have that, by expanding the same nucleuses,

$$<B; [\bar{\tau}]> \xrightarrow{*} <B_1 \cup \ldots \cup B'' \cup \ldots \cup B_n; [\tau_1, \ldots, \mu_1, \ldots, \mu_p, \ldots, \tau_n]>$$

where $[\mu_1, \ldots, \mu_p] \equiv [\bar{\mu}]$ $(p > 0)$. Let now

$$B' = \bigcup_{1 \leq j \neq i \leq n} B_j \cup B'', \; [\bar{\nu}] \equiv [\tau_1, \ldots, \mu_1, \ldots, \mu_p, \ldots, \tau_n],$$

$s = s' \circ s''$ where $s'' = \{(\psi, \omega) \mid$ for all ψ which occur in $<B_h; [\tau_h]>$ $(1 \leq h \leq n, h \neq i)\}$. It is easy to see that $s(<B'; [\bar{\nu}]>) \Rightarrow <B^*; [\bar{\sigma}]>$ (in fact, all types which do not occur in $<B''; [\bar{\mu}]>$ are reduced to ω and then eliminated by normalizing).

Proof of the Lemma. By structural induction on A. Thanks to Property S', we need prove each step only for the case $[\bar{\sigma}] \equiv [\sigma]$ and $[\bar{\tau}] \equiv [\tau]$.

First step. If $A \equiv \Omega$ we must have $B^* = B = \Phi$ and $\sigma \equiv \tau \equiv \omega$.
If $A \equiv x$ then $<B; [\tau]>$ is a trivial variant of $<B^*; [\sigma]>$ and the proof is immediate.

Inductive step. By cases.

1) $A \equiv \lambda x.A'$. Then $\sigma \equiv [\bar{\sigma}_1]\sigma_2$ and $\tau \equiv [\bar{\tau}_1]\tau_2$ where $\sigma_2 \not\equiv \omega$ and $\tau_2 \not\equiv \omega$. From Definition 7 we have that $<B \cup \{\bar{\tau}_1 x\}; [\tau_2]>$ and $<B^* \cup \{\bar{\sigma}_1 x\}; [\sigma_2]>$ are ground pairs for A' and, by inductive hypothesis, there exist $B', [\bar{\nu}_1], [\nu_2]$ such that B' does not contain type assignments to x,

$<B \cup \{\bar{\tau}_1 x\}; [\tau_2]> \xrightarrow{*} <B' \cup \{\bar{\nu}_1 x\}; [\nu_2]>$ and

$$s(<B' \cup \{\bar{\nu}_1 x\}; [\nu_2]>) \Rightarrow <B^* \cup \{\bar{\sigma}_1 x\}; [\sigma_2]>,$$

for some substitution s. Since each nucleus of $<B \cup \{\bar{\tau}_1 x\}; [\tau_2]>$ is also a nucleus of $<B; [[\bar{\tau}_1]\tau_2]>$ this implies that $<B; [[\bar{\tau}_1]\tau_2]> \xrightarrow{*} <B'; [[\bar{\nu}_1]\nu_2]>$.
Moreover, since $\sigma_2 \not\equiv \omega$,

$$s(<B'; [[\bar{\nu}_1]\nu_2]>) \Rightarrow <B^*; [[\bar{\sigma}_1]\sigma_2]>.$$

2) $A \equiv xA_1 \ldots A_p$. In this case Definition 7 implies that:

a) $B = \bigcup_{i=1}^{p} B_i \cup \{[\bar{\tau}_1] \ldots [\bar{\tau}_p] \psi\, x\}$ for some sequence schemes $[\bar{\tau}_1], \ldots, [\bar{\tau}_p]$ such that $<B_i; [\bar{\tau}_i]>$ are ground pairs for A_i, $<B_i; [\bar{\tau}_i]>$ are all pairwise disjoint and ψ does not occur in $<B_i; [\bar{\tau}_i]>$ for $1 \leq i \leq p$.

b) $B^* = \bigcup_{i=1}^{p} B_i^* \cup \{[\bar{\sigma}_1] \ldots [\bar{\sigma}_p] \phi\, x\}$ for some sequence schemes $[\bar{\sigma}_1], \ldots, [\bar{\sigma}_p]$ such that $<B_i^*; [\bar{\sigma}_i]>$ are ground pairs for A_i, $<B_i^*; [\bar{\sigma}_i]>$ are all pairwise disjoint and ϕ does not occur in $<B_i^*; [\bar{\sigma}_i]>$ for $1 \leq i \leq p$.

By inductive hypothesis and Property S', for each i ($1 \le i \le p$) there exist $B'_i, [\bar{\nu}_i]$ such that $<B_i;[\bar{\tau}_i]> \overset{*}{\rightarrow} <B'_i;[\bar{\nu}_i]>$ and $s_i(<B'_i;[\bar{\nu}_i]>) \Rightarrow <B^*_i;[\bar{\sigma}_i]>$ for some substitution s_i. From point iv) of Definition 7 and Definition 4 we have that each nucleus of $<B_i;[\bar{\tau}_i]>$ is also a nucleus of $<B;[\psi]>$ and it is easy to verify that this property remains true also through successive expansions. Then, for each i we have:

$$<B_1 \cup \ldots \cup B_i \cup \ldots \cup B_p \cup \{[\bar{\tau}_1]\ldots[\bar{\tau}_i]\ldots[\bar{\tau}_p]\psi\, x\};[\psi]> \overset{*}{\rightarrow}$$
$$<B_1 \cup \ldots \cup B'_i \cup \ldots \cup B_p \cup \{[\bar{\tau}_1]\ldots[\bar{\nu}_i]\ldots[\bar{\tau}_p]\psi\, x\};[\psi]>$$

(where we suppose, according to Definition 5, that all variables introduced in the expansions of $<B_i;[\bar{\tau}_i]>$ ($1 \le i \le p$) do not occur in the relevant type schemes). Then by expanding (in any order) all $<B_i;[\bar{\tau}_i]>$ we obtain: $<B_1 \cup \ldots \cup B_p \cup \{[\bar{\tau}_1]\ldots[\bar{\tau}_p]\psi\, x\};[\psi]> \overset{*}{\rightarrow} <B'_1 \cup \ldots \cup B'_p \cup \{[\bar{\nu}_1]\ldots[\bar{\nu}_p]\psi\, x\};[\psi]>$.
Then if we define $s = s_1 \circ \ldots \circ s_p \circ \{(\psi,\phi)\}$ and $B' = \bigcup_{i=1}^{p} B'_i \cup \{[\bar{\nu}_1]\ldots[\bar{\nu}_p]\psi\, x\}$ we have $s(<B';[\psi]>) \Rightarrow <B^*;[\phi]>$. □

We have now to prove (Lemma 6) that also a weakened form of the converse of Lemma 4 is true, i.e. that starting from arbitrary ground pairs for any Ω-normal form A we obtain, by successive expansions, only ground pairs for A.

Lemma 5 contains a preliminary technical result.

LEMMA 5. If $<B;[\bar{\tau}]>$ is a ground pair for some $A \in N$, E is a nucleus of $<B;[\bar{\tau}]>$ and all type schemes which are contained in $[\bar{\tau}]$ belong to E, then all type schemes which are predicates in B must belong to E.

Proof. It is easy to prove, similarly as in Lemmas 2 and 4, that if this Lemma is true for a sequence which contains only one type scheme, then it is true for arbitrary sequences. Then we can apply this property in a proof by structural induction on A.

First step. If $A \equiv \Omega$ the Lemma is trivial. If $A \equiv x$ then we must have $\tau \equiv \psi$ and $B = \{\psi x\}$.

Inductive step. By cases:

1) $A \equiv \lambda x.A'$. By Definition 7 we must have $\tau \equiv [\bar{\tau}_1]\tau_2$ and $<B \cup \{\bar{\tau}_1 x\};[\tau_2]>$ must be a ground pair for A'. So, by inductive hypothesis, each nucleus E of $<B \cup \{\bar{\tau}_1 x\};[\tau_2]>$ which contains τ_2 must contain all type schemes of $B \cup \{\bar{\tau}_1 x\}$ and, therefore, all the more reason for the Lemma being true.

2) $A \equiv xA_1\ldots A_p$. By Definition 7, $\tau \equiv \psi$ is a variable, $B = \bigcup_{i=1}^{p} B_i \cup \{[\bar{\tau}_1]\ldots[\bar{\tau}_p]\psi\, x\}$ and $<B_i;[\bar{\tau}_i]>$ are ground pairs for A_i ($1 \le i \le p$). If ψ belongs to the nucleus E then by conditions iii) and iv) of Definition 4 also $[\bar{\tau}_1]\ldots[\bar{\tau}_p]\psi$ must belong to E.

Let E_i ($1 \leq i \leq p$) be the submultiset of E such that σ belongs to E_i iff σ occurs as a subtype of some predicate in B_i. We want prove that all the predicates in B_i belong to E_i. If $[\bar{\tau}_i] \equiv [\omega]$ then $B_i = \Phi$ and the proof is trivial. Otherwiselet $[\bar{\tau}_i] \equiv [\rho_1,..., \rho_q]$; we call E'_i the multiset obtained as the multi-union of $\rho_1,..., \rho_q$ and E_i.
It is easy to verify, looking at Definition 4, that E'_i is a nucleus of $<B_i; [\bar{\tau}_i]>$. In fact conditions i), iv) are true for the types in E_i since E_i is a submultiset of E and E satisfies i), iv) by hypothesis, while they are true for $\rho_1,..., \rho_q$, respectively, by Definition 7 i) and by construction. Condition ii) is true since all types in E_i belong to E and occur in B_i, while $\rho_1,...$ $..., \rho_q$ are contained in $[\bar{\tau}_i]$. Condition iii) is true since $<B_i; [\bar{\tau}_i]>$ is disjoint from $<B_{i'}; [\bar{\tau}_{i'}]>$ for $i \neq i'$. By inductive hypothesis, then, since all type schemes which are contained in $[\bar{\tau}_i]$ belong to E'_i, all type schemes which are predicates in B_i belong to E'_i. Since this is true for all i, all predicates of statements in $\bigcup_{i=1}^{p} B_i$ must belong to E.

LEMMA 6. If $<B; [\bar{\tau}]>$ is a ground pair for some $A \in N$, and $<B; [\bar{\tau}]> \to$ $<B^*; [\bar{\sigma}]>$ then $<B^*; [\bar{\sigma}]>$ is, in its turn, a ground pair for A.

Proof. Let E be the nucleus which is expanded in $<B; [\bar{\tau}]> \to <B^*; [\bar{\sigma}]>$ We prove the Lemma only in the case in which E cannot be split into $m > 1$ multisets $E_1,..., E_m$ such that each E_i ($1 \leq i \leq m$) is, in its turn, a nucleus. In fact, otherwise, we could achieve the same effect of the expansion of E by expanding $E_1, ..., E_m$ one at a time. If the Lemma is true for each E_i ($1 \leq i \leq m$) then it is true also for E.

As in other proofs, we show first that, for an arbitrary A, if we suppose the Lemma true in the case that $[\bar{\tau}]$ contains only one type scheme then it is true also in the general case (Property S").

Proof of Property S". If $[\bar{\tau}] \equiv [\tau_1,..., \tau_m]$ and $m > 1$ then, by Definition 7, we have that $\tau_i \not\equiv \omega$, $B = \bigcup_{i=1}^{m} B_i$, $<B_i; [\tau_i]>$ for $1 \leq i \leq m$ are ground pairs for A and $<B_i; [\tau_i]>$, $<B_j; [\tau_j]>$ must be pairwise disjoint for $1 \leq i \neq j \leq m$. Since we suppose that E cannot be split into different subnucleus, there must exist an index j such that E is a nucleus of $<B_j; [\tau_j]>$. I.e. $<B_j; [\tau_j]> \to <B_j^*; [\bar{\nu}]>$ by expanding E. Then $B^* = \bigcup_{1 \leq i \neq j \leq m} B_i \cup B_j^*$ and $[\bar{\sigma}] \equiv [\tau_1,..., \nu_1,..., \nu_p,..., \tau_m]$ where $[\nu_1, ..., \nu_p] \equiv [\bar{\nu}]$ $(p > 0)$. It is immediate to verify that $<B^*; [\bar{\sigma}]>$ is a ground pair for A.

Proof of the Lemma. We consider two cases, according as all type schemes contained in $[\bar{\tau}]$ belong to E or not. By property S" we need consider only the case $[\bar{\tau}] \equiv [\tau]$.

Case 1. τ belongs to E. In this case, by Lemma 5, all predicates of B

must belong to E and, therefore, we have $B^* = \biguplus_{i=1} B_i$ and $[\bar{\sigma}] \equiv [\tau_1,...,\tau_p]$ where $< B_i; [\tau_i] >$ is a trivial variant of $<B; [\tau] >$ for $1 \leq i \leq p$. Therefore, by Definition 7, $< B^*; [\bar{\sigma}] >$ is a ground pair for A.

Case 2. τ does not belong to E. Then the expansion in question produces trivial variants only of proper subtypes of τ and therefore $[\bar{\sigma}] \equiv [\sigma]$. We prove this case by induction on the structure of A. By Property S" in the proof of each step we need consider only the case $[\bar{\tau}] \equiv [\tau]$.

First step. If $A \equiv \Omega$ the Lemma is trivially true (we have $B = \Phi$ and $\tau \equiv \omega$). If $A \equiv x$ we have $B = \{\psi x\}$ and $\tau \equiv \psi$. The only nucleus of $< B; [\tau] >$ is ψ , ψ and we are in case 1.

Inductive step. By cases:

1) $A \equiv \lambda x.A'$. By Definition 7, $\tau \equiv [\bar{\tau}_1]\tau_2$ and $<B \cup \{\bar{\tau}_1 x\}; [\tau_2]>$ is a ground pair for A'. Since neither $[\bar{\tau}_1]\, \tau_2$ (by hypothesis) nor τ_2 (by condition iv) of Definition 4) can belong to E , E is also a nucleus of $< B \cup \{\bar{\tau}_1 x\}; [\tau_2]>$ and $< B \cup \{\bar{\tau}_1 x\} ; [\tau_2]> \to < B^* \cup \{\bar{\sigma}_1 x\} ; [\sigma_2] >$ by expanding E . By inductive hypothesis we have that $< B^* \cup \{\bar{\sigma}_1 x\} ; [\sigma_2]>$ is a ground pair for A' and so $< B^*; [[\bar{\sigma}_1]\sigma_2]>$ is a ground pair for A. Moreover, by expanding E , we have $<B;[[\bar{\tau}_1]\tau_2]> \to <B^*; [[\bar{\sigma}_1]\sigma_2] >$.

2) $A \equiv xA_1...A_p$. We must have, by Definition 7, $\tau \equiv \psi$, $B = \biguplus_{i=1} B_i \cup \{[\bar{\tau}_1]... [\bar{\tau}_p]\psi\ x\}$ where $<B_i; [\bar{\tau}_i] >$ are ground pairs for A and $< B_i ;[\bar{\tau}_i]>, < B_{i'} ; [\bar{\tau}_{i'}] >$ are pairwise disjoint for $1 \leq i \neq i' \leq p$. Since we consider the case in which E does not containt ψ and E does not split in two or more subnucleus, then condition iii) of Definition 4 assures us that there is an index j such that E is a nucleus of $<B_j; [\bar{\tau}_j] >$. Then if $< B'; [\bar{\nu}] >$ is obtained by applying the expansion in question to $< B_j; [\bar{\tau}_j] >$ we have, by Property S", inductive hypothesis and Case 1 of this proof, that $<B'; [\bar{\nu}] >$ is a ground pair for A_j . Then if $B^* = \bigcup_{1 \leq i \neq j \leq p} B_i \cup B' \cup \{[\bar{\tau}_1]... [\bar{\nu}] ... [\bar{\tau}_p]\psi\ x\}$ and $\sigma \equiv \psi$ we have $< B;[\psi] > \to <B^*; [\psi] >$ and, by Definition 7, $<B^*; [\psi]>$ is a ground pair for A. □

Lastly we can state the following:

THEOREM 4. If $A \in N$ then each ground pair for A is complete for A.

Proof. If $< B; [\bar{\tau}]>$ is a ground pair for A and $<B; [\bar{\tau}] > \xrightarrow{*} < B^*; [\bar{\sigma}]>$ then, by Lemma 6, $< B^*; [\bar{\sigma}] >$ is, in its turn, a ground pair for A. Moreover, if s is a substitution, and $s(< B^*; [\bar{\sigma}] >) \Rightarrow <B'; [\bar{\nu}]>$ then, by Lemma 3, $B' \vdash [\bar{\nu}]$ A.

Viceversa, if $B' \vdash [\bar{\nu}]$ A, Lemma 2 implies that there are a ground pair $< B^*; [\bar{\sigma}] >$ for A and a substitution s such that $s(<B^*; [\bar{\sigma}] >) \Rightarrow <B'; [\bar{\nu}]>$.

Moreover, if $<B;[\bar{\tau}]>$ is any ground pair for A, then, by Lemma 4, there are B'', $[\bar{\mu}]$ and s' such that $<B;[\bar{\tau}]> \overset{*}{\Rightarrow} <B'';[\bar{\mu}]>$ and $s'(<B'';[\bar{\mu}]>) \Rightarrow$ $\Rightarrow <B^*;[\bar{\sigma}]>$.

By Lemma 1, we can postpone normalizations after substitutions and this completes the proof. □

Although all ground pairs for a given Ω-normal form A are complete according to Definition 6, none of them, unfortunately, is complete if we consider only the operation of substitution. It can be proved, in fact, that, given an arbitrary ground pair $<B;[\bar{\tau}]>$, it is always possible to find another one $<B^*;[\bar{\sigma}]>$ such that $<B^*;[\bar{\sigma}]>$ is not an instance of $<B;[\bar{\tau}]>$. As a positive property, however, we have that given any two ground pairs $<B;[\bar{\tau}]>$, $<B^*;[\bar{\nu}]>$ for a term there always exists a ground pair $<B';[\bar{\sigma}]>$ such that both $<B;[\bar{\tau}]>$ and $<B^*;[\bar{\nu}]>$ are instances of it (modulo normalization). The simplest way to build it is to choose $B' = B \cup B^*$ and $[\bar{\sigma}] = [\bar{\tau}, \bar{\nu}]$ but there are, in general, more elegant choices. This implies also that for each finite set F of pairs $<B;[\bar{\tau}]>$ such that $B \vdash [\bar{\tau}]$ A there exists some (not unique) ground pair $<B^*;[\bar{\sigma}]>$ such that all pairs in F are instances of it.

4. THE PRINCIPAL BASIS AND TYPE SCHEMES OF A TERM

Among the pairs of basis and sequence schemes which we have proved to be complete for a given Ω-normal form A we will choose, in this section, a particular one that we will call the principal for A. Since, in this pair, the sequence scheme will contain only one type scheme, we call it the principal type scheme of A. Then we will prove some properties about principal basis and type schemes which allow us to generalize these concepts to arbitrary terms.

DEFINITION 8. A type scheme π is the *principal type scheme (p.t.s.)* of an Ω-normal form A and a basis scheme B is the *principal basis scheme (p.b.s.)* of A iff B, π satisfy the following conditions:

i) if $A \equiv \Omega$ then $B = \Phi$ and $\pi \equiv \omega$

ii) if $A \equiv x$ then $B = \{\psi x\}$ and $\pi \equiv \psi$ where ψ is a type variable

iii) if $A \equiv \lambda x.A'$, and B', π' are the p.b.s. and p.t.s. of A' then if x occurs in A' $B = B' - \{\bar{\sigma}\, x\}$, $\pi \equiv [\bar{\sigma}]\, \pi'$ (x does not occur in B) otherwise $B = B'$, $\pi \equiv [\omega]\pi'$.

iv) if $A \equiv xA_1 \ldots A_p$ and B_i, π_i are the p.b.s.s and p.t.s.s of A_i for $1 \le i \le p$ (we choose a suitable trivial variant of them such that they are all pairwise disjoint) then $B = \bigcup_{i=1}^{p} B_i \cup \{[\pi_1] \ldots [\pi_p]\psi\, x\}$

and $\pi \equiv \psi$ where ψ is a type variable which does not occur in B_i, π_i for $1 \le i \le p$. □

It is immediate, from Definitions 7 and 8, that if B, π are the p.b.s. and p.t.s. of A then $<B; [\pi]>$ is a ground pair for A. In particular, it is easy to prove, by structural induction on A, that if $<B'; [\bar{\sigma}]>$ is an arbitrary ground pair for A then there is an ω-substitution s_ω such that $s_\omega(<B'; [\bar{\sigma}]>) \Rightarrow$ $\Rightarrow <B; [\pi]>$. In this sense the p.b.s. and p.t.s. of a term define the simplest ground pair. Moreover, $<B; [\pi]>$ is the unique ground pair for A such that the deduction $B \vdash [\pi]$ A requires only one application of rule Si (to obtain $[\pi]$ A from πA). In this deduction, in fact, exactly one type scheme is assigned to each component of A. In the deductions associated to all other ground pairs for A, on the contrary, there is at least one component of A to which it is assigned more than one type scheme.

Definitions 7, 8 don't allow to define any complete pair and then any p.b.s. and p.t.s. for terms which are not Ω-normal forms. We can, however, define, for an arbitrary term X, a set of pairs which is complete for X according to Definition 6. We will do this through the notion of approximant of a term.

We recall that (Wadsworth, 1976) an Ω-normal form A is a *direct approximant* of a term X iff A and X differ (modulo Ω-reductions) only for components which are all Ω in A. The set A(X) of approximants of a term X is defined as:

A(X)={A |∃X', such that X = X', A is a direct approximant of X' and A is in Ω-normal form }.

The following Lemma characterizes the relation between the types of an arbitrary term and those of its approximants.

LEMMA 7 (Coppo *et al.*, 1979b). $B \vdash \tau$X iff there exists A ∈ A(X) such that $B \vdash \tau$A. □

DEFINITION 9. If X is a term, Π(X) is the set of pairs <p.b.s.;[p.t.s.]> of all approximants of X, i.e.:

Π (X) = { $<B; [\pi]>$ | ∃ A∈ A(X) such that B, π are, respectively, the p.b.s. and p.t.s. of A } □

It is obvious from Lemma 7 and Theorem 4 that, for all arbitrary terms X, Π(X) is a complete set of types for X, i.e.:

THEOREM 5. Let X be an arbitrary term. Then the set Π (X) is complete for X. □

Let's define P as the set of pairs $<B; [\pi]>$ such that B and π are, respectively, the p.b.s. and p.t.s. of some Ω-normal form A.

Let's now define, on N and P, two partial order relations.
The elements of N are ordered according to $\leq$ so defined (Lévy, 1978):

i) $\Omega \leq A$ for all $A \in N$

ii) Let $A \equiv \lambda x_1 \ldots x_n .zA_1 \ldots A_p$, $A' \equiv \lambda x_1 \ldots x_{n'} .z'A'_1 \ldots A'_{p'}$.
$A \leq A'$ iff $n = n'$, $p = p'$, $z \equiv z'$ and $A_i \leq A'_i$ $(1 \leq i \leq p)$.

In P we define $<B;[\pi]> \sqsubseteq <B';[\pi']>$ iff there exists an ω-substitution s_ω such that $s_\omega(<B';[\pi']>) \Rightarrow <B;[\pi]>$.
We can prove that the relation $\leq$ on N corresponds to $\sqsubseteq$ on P.

LEMMA 8. Let $A, A' \in N$ and B, π, B', π', respectively, their p.b.s.s and p.t.s.s. Then $A \leq A'$ iff $<B;[\pi]> \sqsubseteq <B';[\pi']>$.

Proof. The proof that $A \leq A'$ implies $<B;[\pi]> \sqsubseteq <B';[\pi']>$ can be done by a simple structural induction on A.
We will prove that $<B;[\pi]> \sqsubseteq <B';[\pi']>$ implies $A \leq A'$ by structural induction on A'.

First step. If $A' \equiv \Omega$ then $B' = \Phi$ and $\pi' \equiv \omega$. The proof is trivial.
If $A' \equiv x$ then $B' = \{\psi\, x\}$ and $\pi' \equiv \psi$. We must have $<B;[\pi]> = <\psi\, x;[\psi]>$ or $<B;[\pi]> = <\Phi;[\omega]>$ and the proof is again trivial.

Inductive step. By cases.

1) $A' \equiv \lambda x.\overline{A}'$. Then, by Definition 8, $\pi' \equiv [\pi'_1]\pi'_2$. In this case, since $\pi \sqsubseteq \pi'$ we can have either $\pi \equiv \omega$ or $\pi \equiv [\pi_1]\pi_2$. If $\pi \equiv \omega$ then $B = \Phi$ and $A \equiv \Omega$. The case is trivial. Else we have, from Definition 8, $A \equiv \lambda x.\overline{A}$. Again from Definition 8 we have that $B \cup \{\pi_1\, x\}, \pi_2$ and $B' \cup \{\pi'_1\, x\}, \pi'_2$ are, respectively, the p.b.s.s and p.t.s.s of $\overline{A}$ and $\overline{A}'$. Moreover it is easy to verify that $<B \cup \{\pi_1\, x\};[\pi_2]> \sqsubseteq <B' \cup \{\pi'_1 x\};[\pi'_2]>$.
By inductive hypothesis, $\overline{A} \leq \overline{A}'$ and then $\lambda x.\overline{A} \leq \lambda x.\overline{A}'$.

2) $A' \equiv xA'_1 \ldots A'_p$. By Definition 8,

$$<B';[\pi']> = <\bigcup_{i=1}^{p} B'_i \cup \{[\nu'_1] \ldots [\nu'_p]\psi\, x\};[\psi]>,$$

where B'_i, ν'_i are the p.b.s.s and p.t.s.s of A'_i $(1 \leq i \leq p)$. Let's notice that either $<B;[\pi]> = <\Phi;[\omega]>$, if ψ is substituted by ω or

$$<B;[\pi]> = <\overline{B} \cup \{[\nu_1] \ldots [\nu_p]\psi\, x\};[\psi]>,$$

where, if s_ω is the ω-substitution that proves $\sqsubseteq$, $s_\omega(\bigcup_{i=1}^{p} B'_i) \Rightarrow \overline{B}$ and $s_\omega(\nu'_i) \Rightarrow \nu_i$ for $1 \leq i \leq p$.
In the first case $A \equiv \Omega$ and the proof is trivial.
In the second case we must have, from Definition 8, $A \equiv xA_1 \ldots A_p$ and $\overline{B} = \bigcup_{i=1}^{p} B_i$ where B_i, ν_i are, respectively, the p.b.s.s and p.t.s.s of A_i and they are all pairwise disjoint $(1 \leq i \leq p)$. Let now $s_\omega(B'_i) \Rightarrow B^*_i$, we want prove that $B^*_i = B_i$ for all i $(1 \leq i \leq p)$. Let C, E_i, E'_i, E^*_i be the multisets of types which occur as predicates, respectively, of the statements

of $\bar{B}$, B_i, B'_i, B^*_i $(1 \leq i \leq p)$. By hypothesis E_i, ν_i and E_j, ν_j are disjoint for all $1 \leq i \neq j \leq p$ and, since s_ω is an ω-substitution, also E^*_i, ν_i and E^*_j, ν_j are such *(Property D).* Moreover C is both the multi-union of all E_i $(1 \leq i \leq p)$ and the multiunion of all $E^*_i (1 \leq i \leq p)$ *(Property U).* We have, from the proof of Lemma 5 (case 2 of the inductive step), that E_i, ν_i is a nucleus of $<B;[\psi]>$ and E'_i, ν'_i is a nucleus of $<B';[\psi]> (1 \leq i \leq p)$. It is so easy to verify, from Definition 4, that also E^*_i, ν_i is a nucleus of $<B;[\psi]>$. Now let $\bar{E}_i$ be the multi-intersection of E_i and E^*_i. We prove that $\bar{E}_i, \nu_i$ is a nucleus of $<B;[\psi]>$. In fact conditions i), ii), iv) of Definition 4 are trivially satisfied. For condition iii) we notice that all types σ which do not occur in $\bar{E}_i$ are disjoint from the ones which occur in $\bar{E}_i$, ν_i. In fact, if σ does not belong to E_i, by Property U it must belong to E_j for some $j \neq i$ and then by Property D σ is disjoint from E_i, ν_i. A similar argument applies if σ does not belong to E^*_i. Since $<B_i;[\nu_i]>$ is a ground pair for A_i, by Lemma 5 we must have $\bar{E}_i = E_i$. Then for all i, E_i is a submultiset of E^*_i. This implies, by Properties U and D, that $E^*_i = E_i$ and, so, since $\bar{B} = \bigcup_{i=1}^{p} B^*_i = \bigcup_{i=1}^{p} B_i$, $B^*_i = B_i$.
Then we have $<B_i;[\nu_i]> \sqsubseteq <B'_i;[\nu'_i]>$ and, by inductive hypothesis, $A_i \leq A'_i$ $(1 \leq i \leq p)$ which implies $xA_1 \ldots A_p \leq xA'_1 \ldots A'_p$. □

In (Lévy, 1978) it is proved that N builds a meet-semilattice with respect to $\leq$. From this fact and the previous Lemma, the following Theorem follows immediately.

THEOREM 6. P, $\sqsubseteq$ is a meet-semilattice isomorphic to N, $\leq$.
Moreover we have the following properties: □

THEOREM 7.

i) For all terms X, $\Pi(X)$ is an ideal in P.

ii) For all $A \in N$, $\Pi(A)$ is a principal ideal in P.

iii) $\Pi(X) = \bigsqcup \{\Pi(A) \mid A \in A(X)\}$.

Proof. All these properties are true in N with $A(X)$ instead of $\Pi(X)$ (Lévy, 1978). So they are true also in P for the isomorphism proved in Theorem 6. □

From Lemma 7 and Theorem 7 i), ii) it follows that, for each term X such that $\Pi(X)$ is finite, $\bigsqcup \Pi(X)$ is a complete pair for X.

COROLLARY 1. For any term X with a finite set of approximants there exists a pair $<B;[\pi]>$ which is complete for X. □

In this case we call B and π, respectively, the p.b.s. and p.t.s. of X.
But, in general, $\Pi(X)$ is not finite and $\bigsqcup \Pi(X)$ does not exist since P, as N, is not complete. However if we consider the set $\hat{P}$ of all ideals of P we have that $\hat{P}$, ordered by the relation of inclusion, is a complete

lattice in which the sublattice of all principal ideals is isomorphic to P. Moreover, from Theorem 7i), iii), the correspondence $X \to \Pi(X)$ defines a mapping from the set of all terms to $\hat{P}$ which extends in a natural way the correspondence between terms with a finite set of approximants and their p.b.s.s, p.t.s.s. It is then consistent to suppose that P can be completed with "infinite" pairs to represent the p.b.s.s and p.t.s.s of terms with an infinite set of approximants.

For example, in the case of $Y \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ we have

$A(Y) = \{\Omega, \lambda f.f\,\Omega, \lambda f.f(f\,\Omega),\ldots\}$

and $\Pi(Y) = \{<\Phi;[\omega]>, <\Phi;[[[\omega]\psi]\psi]>, <\Phi;[[[\omega]\psi_2,[\psi_2]\psi_1]\psi_1]>,\ldots\}$.

So we can suppose the p.t.s. of Y to be the "infinite" type scheme obtained as the limit for $n \to \infty$ of $[[\omega]\psi_n,[\psi_n]\psi_{n-1},\ldots,[\psi_2]\psi_1]\psi_1$.

This completion of P allows us to say that there exists a (possibly "infinite") complete pair for any term (if we extend in an obvious way the notions of substitution and normalization).

5. A MODEL OF λ-CALCULUS

In this section we will show that the power set of the set T_N of normalized type schemes is a suitable domain to build a λ-calculus model. Let Υ be obtained by adjoining ω to each element of the power set of $T_N - \{\omega\}$. In other words, Υ is built by all subsets of T_N which contain ω. We define a binary operation $\bullet : \Upsilon^2 \to \Upsilon$ and a map $[\![\ ]\!] : \Lambda \to \Upsilon$ (Λ represents the set of terms) in such a way that the conditions on λ-structures given in (Hindley *et al.*, 1980) are satisfied by $<\Upsilon, \bullet, [\![\ ]\!]>$.

DEFINITION 10. If $d_1, d_2 \in \Upsilon$ then

$d_1 \bullet d_2 = \{\omega\} \cup \{\tau \mid [\sigma_1,\ldots,\sigma_n]\tau \in d_1$ and $\sigma_i \in d_2$ for $1 \leq i \leq n\}$. □

We notice that the operation $\bullet$ is defined as a particular case of the more general notion of application of (Plotkin, 1972).

To define $[\![\ ]\!]$ let's remember that an environment e is a map from the set of free variables V to the domain, i.e. e: $V \to \Upsilon$. It is natural to associate the environments with bases taking into account that a basis is always finite. Then we define, for each environment e, a set of basis schemes B_e as follows: $B_e = \{B \mid \tau x \in B$ implies $\tau \in e(x)\}$. It is clear that $\Phi \in B_e$ for all e.

DEFINITION 11. The value of a term X in an environment e is

$[\![X]\!]_e = \{\tau \mid \exists B \in B_e$ such that $B \vdash \tau X\}$. □

Obviously $\omega \in [\![X]\!]_e$ for all X and all e.

By Definition 11, we have that $B \vdash [\bar{\sigma}]X$ and $B \in B_e$ implies that the set of type schemes contained in $[\bar{\sigma}]$ is a subset of $[\![X]\!]_e$.

THEOREM 8. $< \Upsilon, \bullet, [\![\ ]\!] >$ is a λ-structure, i.e. it satisfies the followin conditions:

i) $[\![x]\!]_e = e(x)$

ii) $[\![XY]\!]_e = [\![X]\!]_e \bullet [\![Y]\!]_e$

iii) $[\![\lambda x.X]\!]_e \bullet d = [\![X]\!]_{e_x^d}$

where, as usual, e_x^d denotes the environment e' such that

$$e'(y) = \begin{cases} d & \text{if } y \equiv x \\ e(y) & \text{if } y \not\equiv x \end{cases}$$

iv) if $[\![x]\!]_e = [\![x]\!]_{e'}$ for all variables x which occur free in X then $[\![X]\!]_e = [\![X]\!]_{e'}$.

v) $[\![\lambda x.X]\!]_e = [\![\lambda y.X[x/y]]\!]_e$ if y does not occur free in X

vi) if $[\![X]\!]_{e_x^d} = [\![Y]\!]_{e_x^d}$ for all $d \in \Upsilon$ then $[\![\lambda x.X]\!]_e = [\![\lambda x.Y]\!]_e$

where these equalities must hold for all variables x,y, terms X, Y and environments e, e'.

Proof. These conditions are verified almost trivially from the given definitions. In particular:

i) Immediate from Definition 11.

ii) If $\tau \in [\![X]\!]_e \bullet [\![Y]\!]_e$ and $\tau \not\equiv \omega$, then, by Definitions 10 and 11, there exist $B_0, B_1, \ldots, B_n \in \mathrm{B}_e$ such that $B_0 \vdash [\sigma_1, \ldots, \sigma_n]\tau\ X$ and $B_i \vdash \sigma_i\ Y$ $(1 \leq i \leq n)$. Then $B = \bigcup_{i=0}^{n} B_i \in \mathrm{B}_e$ and, by rules F'e and Si, $B \vdash \tau XY$, i.e. $\tau \in [\![XY]\!]_e$. The case $\tau \equiv \omega$ is trivial. The proof of the converse is similar.

iii) If $\tau \in [\![\lambda x.X]\!]_e \bullet d$ and $\tau \not\equiv \omega$ then there is a sequence scheme $[\bar{\sigma}] \equiv [\sigma_1, \ldots, \sigma_m]$ and a basis $B \in \mathrm{B}_e$ such that $\sigma_i \in d$ $(1 \leq i \leq m)$ and $B \vdash [\bar{\sigma}]\tau\ \lambda x.X$. Then $B \cup \{\bar{\sigma}x\} \in \mathrm{B}_{e_x^d}$ and so $\tau \in [\![X]\!]_{e_x^d}$ (this holds also in the case $[\bar{\sigma}] \equiv [\omega]$). The case $\tau \equiv \omega$ is trivial. The proof of the converse is similar.

iv) v) and vi) hold trivially. □

The above semantics induces in a natural way a partial order relation between terms: $X \leq^* Y$ iff $[\![X]\!]_e \subseteq [\![Y]\!]_e$ for all environments e, that is $X \leq^* Y$ iff, for all B and τ, $B \vdash \tau X$ implies $B \vdash \tau Y$.

We can define another partial order relation between terms, i.e. $X \lesssim Y$ iff $\Pi(X) \subseteq \Pi(Y)$.

The relation $\lesssim$ coincides with the semantic relation $\sqsubseteq_L$ induced by Lévy's syntactic model of λ-calculus (that is $X \sqsubseteq_L Y$ iff $A(X) \subseteq A(Y)$).

THEOREM 9. $X \sqsubseteq_L Y$ iff $X \lesssim Y$.

Proof. The proof is obvious from the correspondence between N and P. □

This means that $\lesssim$ is also equivalent to the inclusion of Böhm trees and, hence, to the partial order relation induced by Plotkin's model T_ω (Barendregt *et al.*, 1980).

In contrast, the relation $\leq^*$ is an extension of these relations:

THEOREM 10. $X \lesssim Y$ implies $X \leq^* Y$.

Proof. By Theorem 9, $X \lesssim Y$ implies $\mathcal{A}(X) \subseteq \mathcal{A}(Y)$. Then $X \leq^* Y$ follows immediately from Lemma 7. □

However, the viceversa is not always true. For example, $\lambda xy.xy \leq^* \lambda x.x$ but $\lambda xy.xy \not\lesssim \lambda x.x$.

Lastly, let's consider the semantic relation $X =^* Y$ iff $X \leq^* Y \leq^* X$. In (Coppo *et al.*, 1979 b) it has been proved that this equality of terms coincides with the equality induced by the models P_ω, T_ω. So we have that all the above relations split the set of terms into the same equivalence classes.

CONCLUSION

The results of this paper show that, given the p.t.s. and p.b.s. of any term with a finite set of approximants, it is possible to define a procedure to generate all type schemes which can be deduced for that term. As a consequence of this it turns out that, given any two arbitrary terms in normal form, it is possible to define a procedure which generates all possible type schemes of their application. Now, since each term is convertible to the application of two normal forms, we have that the set of all type schemes which can be assigned to an arbitrary term is recursively enumerable. This property is confirmed, in fact, by the results of section 4.

ACKNOWLEDGEMENTS

We are very grateful to Prof. R. Hindley for his helpful suggestions.

REFERENCES

Barendregt, H. and Longo, G. (1980). Equality of Lambda Terms in the Model T_ω, *This Volume.*

Coppo, M. and Dezani-Ciancaglini, M. (1978). An Extension of Basic Functionality Theory for λ-calculus, *Notre Dame Journal of Formal Logic,* (to appear).

Coppo, M., Dezani-Ciancaglini, M. and Venneri, B. (1979a). Functional Characracters of Solvable Terms, *Zeitschrift für Mathematische Logik und Grund-*

lagen der Mathematik, (to appear).

Coppo, M., Dezani-Ciancaglini, M. and Sallé, P. (1979b). Functional Characterization of some Semantic Equalities Inside λ-calculus. *In* "Automata, Languages and Programming" (ed. E. Maurer), Lecture Notes in Computer Science, vol. 71, pp. 133-146, Springer-Verlag, Berlin.

Curry, H.B. (1969). Modified Basic Functionality in Combinatory Logic, *Dialectica* 23, 83-92.

Curry, H.B. and Feys, R. (1958). "Combinatory Logic", Vol. 1, North-Holland, Amsterdam.

Curry, H.B., Hindley, R. and Seldin, J.P. (1972). "Combinatory Logic", Vol. 2, North-Holland, Amsterdam.

Hindley, R. (1969). The Principal Type Scheme of an Object in Combinatory Logic, *Transactions of the American Mathematical Society*, 146, 29-60.

Hindley, R. and Longo, G, (1980). Lambda Calculus Models and Extensionality, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 26, 2, (to appear).

Lévy, J.J. (1978). "Reductions Correctes et Optimales dans le λ-calcul". Thèse d'Etat, Paris.

Plotkin, G. (1972). A Set Theoretical Definition of Application, *Memo* MIP-R-95, School of Artificial Intelligence, University of Edinburgh.

Scott, D. (1972). Data Types as Lattices, *SIAM Journal on Computing*, 5, 3, 522-587.

Wadsworth, C.P. (1976). The Relation between Computational and Denotational Properties for Scott's D_∞-Models of the Lambda Calculus, *SIAM Journal on Computing*, 5, 3, 488-521.

A TYPE ASSIGNMENT FOR THE STRONGLY NORMALIZABLE λ-TERMS

Garrel Pottinger

Department of History and Philosophy
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213, USA.

Dedicated to H.B. Curry on the occasion of his 80th birthday

1. INTRODUCTION

This paper provides an assignment of type symbols to the λK-terms which are βη-strongly normalizable. The assignment can be viewed as an extension of the formulas-as-types approach to the study of intuitionist logic and, consequently, may be said to have its ultimate origins in the remarks of Curry and Feys (1958, pp. 313-315).

The type symbols employed are formulas built up from grouping indicators, propositional parameters, $\supset$, and the new connective $\hat{\&}$. The intuitive meaning of $\hat{\&}$ can be explained by saying that to assert $A \mathbin{\hat{\&}} B$ is to assert that one has a reason for asserting A which is <u>also</u> a reason for asserting B. Taken together with the usual intuitionist understanding of $\supset$, this reading of $\hat{\&}$ provides a nice motivation for all but one of the rules used in the first system defined below to establish the promised assignment of type symbols. The exception is a rule designed to allow for as full a treatment of η-conversion as possible.

In Curry and Feys (1958, pp. 313-315) it is pointed out that the functional characters assigned to closed terms in the basic theory of functionality are in effect the theorems of intuitionist propositional logic in which the only logical constant which occurs is $\supset$. Given this, it is natural in the present setting to define A to be a theorem iff it is the type symbol of a closed term and then consider the properties of $\hat{\&}$ in the light of this definition. Although the bulk of this paper is devited to questions which arise from considering the systems presented here as providing as assignment of type symbols to λ-terms, the consequences of this definition are explored briefly in section 6. In particular, it will be shown that the behavior of $\supset$ and $\hat{\&}$ is quite different from the behavior of $\supset$ and &. This is to be expected, since, according to the usual intuitionist understanding of &, to assert A & B is to assert that one has a <u>pair</u> of reasons, the first of which is a reason for asserting A and the second of which is a reason for asserting B. Evidently, this is quite different from the reading for $\hat{\&}$ given above. The point of section 6 is to show how this intuitive difference manifests itself formally.

The results proved here should be compared with the recent work on extended type assignments for λ-terms contained in Coppo and Dezani-Ciancaglini (1978), Sallé (1978), Coppo, Dezani-Ciancaglini, and Sallé (1979), and Coppo and Dezani-Ciancaglini (1980). They are especially similar to the results presented in Coppo and Dezani-Ciancaglini (1980) but go beyond the latter in that they cover the λK-terms and include a treatment of η-conversion. This similarity will be described precisely after the necessary for mal machinery has been introduced.

In what follows systems will be thought of concatenatively, but, in accordance with Curry's policy, no expression of these systems will be written down. Only U-language expressions will appear in this paper. Curry's punctuational conventions will be adopted, and notations used without explanation are to be understood according to Curry's definitions of them. '=' expresses identity.

2. THE SYSTEMS S_1 AND S_2

<u>Statements</u> of these systems have the form $t \models A$, where t is a λK-term and A is a formula built up in the way described above. $t \models A$ may be interpreted as saying that t is a reason for asserting A.[1] $P, Q, R\ldots$ are to be lists of the form $x_1 \models A_1, \ldots, x_n \models A_n$ $(n \geq 0)$, where for all $i, j (1 \leq i < j \leq n)$, $x_i \neq x_j$. <u>Sequents</u> of S_1 and S_2 have the form $P \vdash t \models A$. P is the <u>antecedent</u> and $t \models A$ is the <u>succedent</u> of of $P \vdash t \models A$. (Note that, according to these definitions, a variable may not occur twice in the antecedent of a sequent. This restriction applies everywhere in what follows, and it is important. For example, it is easy to see that the rule $\supset$I, which will be stated momentarily, would be unsound without it.)

Intuitively, $x_1 \models A_1, \ldots, x_n \models A_n \vdash t \models A$ is supposed to mean that if $x_1, \ldots, x_n$ were replaced by reasons for asserting $A_1, \ldots, A_n$, respectively, in t, then the result would be a reason for asserting A.

<u>Derivations</u> of S_1 are finite, ordered trees of sequents built up according to the following specifications.

Axioms

$$P, x \models A, Q \vdash x \models A$$

Rules

$\supset$E
$$\frac{P \vdash t \models A \supset B \qquad P \vdash u \models A}{P \vdash tu \models B}$$

$\supset$I
$$\frac{P, x \models A, Q \vdash t \models B}{P, Q \vdash \lambda y[y/x]t \models A \supset B}$$
provided $y \neq x$ only if y is not free in t

$\hat{\&}$E
$$\frac{P \vdash t \models A \mathbin{\hat{\&}} B}{P \vdash t \models A} \qquad \frac{P \vdash t \models A \mathbin{\hat{\&}} B}{P \vdash t \models B}$$

$\hat{\&}$I
$$\frac{P \vdash t \models A \qquad P \vdash t \models B}{P \vdash t \models A \mathbin{\hat{\&}} B}$$

η
$$\frac{P \vdash \lambda x.tx \models A}{P \vdash t \models A}$$
provided x is not free in t

It should be clear that the axioms and rules of S_1 other than η accord with the intended meaning of sequents. η is a rule of type inclusion which allows for the treatment of η-conversion. It will turn out that η-conversion must be restricted, despite the presence of η.

To try to treat η-conversion as fully as possible is obviously reasonable from the point of view of combinatory logic, but it is also clear that this motive is independent of the motivation given above for the rules of S_1 other than η. This independence can be worked out formally. Let $S_1-\eta$ be the system obtained from S_1 by deleting η. It is not difficult to see that $S_1-\eta$ assigns type symbols to the same set of terms as S_1 (though it does not assign the same type symbols), that deleting η does not disturb the treatment of β-conversion given in section 4, and that the set of formulas which are theorems according to the definition given in section 1 is not changed by passing from S_1 to $S_1-\eta$. It also happens that the assignment of type symbols provided by $S_1-\eta$ is essentially the same as that given in Coppo and Dezani-Ciancaglini (1980) -- if one simply rewrites the notation '$[\sigma_1,\ldots,\sigma_n]$' used there as '$A_1 \hat{\&} \ldots \hat{\&} A_n$' and rewrites '$F[\sigma_1,\ldots,\sigma_n]\tau$' as '$A_1 \hat{\&} \ldots \hat{\&} A_n \supset B$', then it is almost trivial to prove that the two assignments are the same.[2] From this it follows that these authors could have extended their treatment of β-conversion to the λK-terms by proceeding in the manner of section 4, below.

Although S_1 expresses the motivation given above in a very clear way, the presence of the rules for $\hat{\&}$ and η make it hard to prove things about S_1.[3] It will now be shown that these rules can be avoided by enlarging the stock of axioms and altering the form of $\supset$I. The resulting system will be called 'S_2'. First, the auxiliary system CL must be defined.

$\Gamma, \Delta, \Theta, \Gamma_1, \ldots$ are to be finite sequences of formulas. Sequents of CL have the form $\Gamma \Vdash A$.

Axioms

$$\Gamma, A, \Delta \Vdash A$$

Rules

$\hat{\&}$E
$$\frac{\Gamma \Vdash A \hat{\&} B}{\Gamma \Vdash A} \qquad \frac{\Gamma \Vdash A \hat{\&} B}{\Gamma \Vdash B}$$

$\hat{\&}$I
$$\frac{\Gamma \Vdash A \qquad \Gamma \Vdash B}{\Gamma \Vdash A \hat{\&} B}$$

$\supset\hat{\&}$ $$\frac{\Gamma \mid\vdash A \supset B \qquad \Gamma \mid\vdash A \supset C}{\Gamma \mid\vdash A \supset B \mathbin{\hat{\&}} C}$$

$\mid\vdash\supset$ $$\frac{C \mid\vdash A \qquad \Gamma \mid\vdash A \supset B}{\Gamma \mid\vdash C \supset B}$$

$\supset\mid\vdash$ $$\frac{\Gamma \mid\vdash A \supset B \quad B \mid\vdash C}{\Gamma \mid\vdash A \supset C}$$

Simple arguments by induction on the length of CL derivations show that the following rules are admissible in CL.

$K\mid\vdash$ $$\frac{\Gamma, \Delta \mid\vdash B}{\Gamma, A, \Delta \mid\vdash B}$$

$C\mid\vdash$ $$\frac{\Gamma, A, B, \Delta \mid\vdash C}{\Gamma, B, A, \Delta \mid\vdash C}$$

$W\mid\vdash$ $$\frac{\Gamma, A, A, \Delta \mid\vdash B}{\Gamma, A, \Delta \mid\vdash B}$$

$\mathrm{Cut}\mid\vdash$ $$\frac{\Gamma \mid\vdash A \qquad \Delta, A, \Theta \mid\vdash B}{\Gamma, \Delta, \quad \mid\vdash B}$$

$X, Y, Z, X_1 \ldots$ are to be sets of formulas. Let Γ^* be the set of formulas occurring in Γ, and define

$cl(X) = \{A:$ for some Γ, $\Gamma^* \subseteq X$ and $\Gamma \mid\vdash A$ is derivable in CL$\}$. Where $\mathfrak{F}$ is the set of formulas, it is easy to see that cl is a closure operation on $\wp\mathfrak{F}$.[4] $\mathrm{Cut}\mid\vdash$ yields the conclusion that $cl(cl(X)) = cl(X)$, and the other conditions are immediate from the definitions of CL and cl.

Where X is a non-empty set of formulas, $P \vdash t \models X$ is to be a sequent having P as its antecedent and having some member of X as the formula on the right side of $\models$ in its succedent. S_2 is defined by the following specifications.

Axioms $\quad P, x \models A, Q \vdash x \models cl(\{A\})$

Rules

$\supset$E As in S_1

$$\supset\mathrm{I}\quad \frac{P, x \models A_1, Q \vdash t \models B_1 \quad \ldots \quad P, x \models A_n, Q \vdash t \models B_n}{P, Q \vdash \lambda y[y/x]t \models \mathrm{cl}(\{A_1 \supset B_1, \ldots, A_n \supset B_n\})}$$

provided $y \neq x$ only if y is not free in t

3. THE EQUIVALENCE OF S_1 AND S_2

LEMMA 3.1 If $P \vdash t \models A_1, \ldots, P \vdash t \models A_n$ are derivable in S_2 and $A \in \mathrm{cl}(\{A_1, \ldots, A_n\})$, then $P \vdash t \models A$ is derivable in S_2.

Proof. Induction on the complexity of t. If t is a variable or t begins with λ, the required argument is trivial. Otherwise, for all i ($1 \leq i \leq n$), $P \vdash A_i$ arises via an inference of the form

$$\frac{P \vdash t_1 \models B_i \supset A_i \qquad P \vdash u_1 \models B_i}{P \vdash t_1 u_1 \models A_i}\ \supset\mathrm{E}$$

It is easy to show that $B_1 \hat{\&} \ldots \hat{\&} B_n \supset A_1 \hat{\&} \ldots \hat{\&} A_n \in \mathrm{cl}(\{B_1 \supset A_1, \ldots, B_n \supset A_n\})$ and that $A \in \mathrm{cl}(\{A_1 \hat{\&} \ldots \hat{\&} A_n\})$. Hence, $B_1 \hat{\&} \ldots \hat{\&} B_n \supset A \in \mathrm{cl}(\{B_1 \supset A_1, \ldots, B_n \supset A_n\})$. Also, $B_1 \hat{\&} \ldots \hat{\&} B_n \in \mathrm{cl}(\{B_1, \ldots, B_n\})$. By Hyp. Ind. $P \vdash t_1 \models B_1 \hat{\&} \ldots \hat{\&} B_n \supset A$ and $P \vdash u_1 \models B_1 \hat{\&} \ldots \hat{\&} B_n$ are derivable in S_2, so $P \vdash t_1 u_1 \models A$ is derivable in S_2 by $\supset$E.

LEMMA 3.2 η is admissible in S_2.

Proof. Suppose $P \vdash \lambda x.tx \models A$ is derivable in S_2, and suppose x is not free in t. $P \vdash \lambda x.tx \models A$ is derived by an inference of the form

$$\frac{Q, y \models B_1, R \vdash ty \models C_1 \quad \ldots \quad Q, y \models B_n, R \vdash ty \models C_n}{Q, R \vdash \lambda x.tx \models cl(\{B_1 \supset C_1, \ldots, B_n \supset C_n\})} \supset I$$

In turn, for each $i (1 \leq i \leq n)$, $Q, y \models B_i, R \vdash ty \models C_i$ arises through an inference of the form

$$\frac{Q, y \models B_i, R \vdash t \models D_i \supset C_i \qquad Q, y \models B_i R \vdash y \models D_i}{Q, y \models B_i, R \vdash ty \models C_i} \supset E$$

Since y is not free in t, it can be shown by induction on the length of S_2 derivations that, for all $i (1 \leq i \leq n)$, $Q, R \vdash t \models D_i \supset C_i$ is derivable in S_2. Also, for all $i (1 \leq i \leq n)$, $Q, y \models B_i, R \vdash y \models D_i$ is an axiom of S_2, and, hence, $D_i \in cl(\{B_i\})$. Lemma 3.1 implies that, for all $i (1 \leq i \leq n)$, $Q, R \vdash t \models B_i \supset C_i$ is derivable in S. The desired conclusion follows from this and lemma 3.1.

COROLLARY 3.3. If $P \vdash t \models A$ is derivable in S_1, then $P \vdash t \models A$ is derivable in S_2.

Proof. Immediate from lemmas 3.1 and 3.2.

By induction on the length of S_1 and S_2 derivations it can be shown that the following rules are admissible in S_1 and S_2.

$$K\vdash \quad \frac{P, Q \vdash t \models B}{P, x \models A, Q \vdash t \models B}$$

$$C\vdash \quad \frac{P, x \models A, y \models B, Q \vdash t \models C}{P, y \models B, x \models A, Q \vdash t \models C}$$

$$W\vdash \quad \frac{P, x \models A, y \models A, Q \vdash t \models B}{P, x \models A, Q \vdash [x/y]t \models B}$$

$$Cut\vdash \quad \frac{P \vdash u \models A \qquad P, x \models A, Q \vdash t \models B}{P, Q \vdash [u/x]t \models B}$$

LEMMA 3.4. If $B \in cl(\{A\})$, then $P, x \models A, Q \vdash x \models B$ is derivable in S_1.

Proof. By induction on the length of CL derivations ending with $A \mid\vdash B$. ($K\mid\vdash$ and $W\mid\vdash$ imply that there is no loss of generality.) Let $\underset{\sim}{D}$ be the given derivation. If $\underset{\sim}{D}$ is an axiom or ends with $\hat{\&}E$ or $\hat{\&}I$, the required argument is trivial. In the $\supset\hat{\&}$ case one proceeds via Hyp. Ind., $K\vdash$, $\supset E$, $\hat{\&}I$, $\supset I$, and η. Hyp. Ind., $K\vdash$, $\supset E$, $\supset I$ and η suffice in the $\mid\vdash\supset$ case, and Hyp. Ind., $K\vdash$, $\supset E$, $Cut\vdash$, $\supset I$, and η yield the desired conclusion in the $\supset\mid\vdash$ case.

LEMMA 3.5. If $P \vdash t \models A_1, \ldots, P \vdash t \models A_n$ are derivable in S_1 and $A \in cl(\{A_1, \ldots, A_n\})$, then $P \vdash t \models A$ is derivable in S_1.

Proof. $P \vdash t \models A_1 \hat{\&} \ldots \hat{\&} A_n$ can be derived in S_1 by means of $\hat{\&}I$, and lemma 3.4 implies that $P, x \models A_1 \hat{\&} \ldots \hat{\&} A_n \vdash x \models A$ is derivable in S_1. $Cut\vdash$ yields the desired conclusion.

THEOREM 3.6. $P \vdash t \models A$ is derivable in S_1 iff $P \vdash t \models A$ is derivable in S_2.

Proof. Immediate from corollary 3.3 and lemmas 3.4 and 3.5.

From now on 'derivable' will often be written instead of 'derivable in S_1' and 'derivable in S_2'. Also, if Q is a result of permuting elements of P, P and Q may be identified in view of $C\vdash$ and $W\vdash$. This will be done in what follows.

4. REDUCTION AND CONVERSION

Define:

$X_{t,P} = \{A : P \vdash t \models A \text{ is derivable}\}$

$X_t = \underset{\sim}{\cup} X_{t,P}$

t_1 $RED_{1\beta}$ t_2 iff there exist t, x, and u s.t. x is not free in t only if $X_u \neq 0$,[5] and t_2 is a result of replacing an occurrence of $(\lambda xt)u$ in t_1 by an occurrence of $[u/x]t$.

$\mathfrak{F}_{\supset}=\{A \supset B: A,B \in \mathfrak{F}\}$

t_1 $RED_{1\eta}$ t_2 iff there exist t and x s.t. $X_t \subseteq cl(\mathfrak{F}_{\supset})$, x is not free in t, and t_2 is a result of replacing an occurrence of $\lambda x.tx$ in t_1 by an occurrence of t.

t_1 $RED_{1\beta\eta}$ t_2 iff t_1 $RED_{1\beta}$ t_2 or t_1 $RED_{1\eta}$ t_2.

$=_\alpha$ is the usual relation of α-conversion.

t_1 RED_β t_2[t_1 $RED_{\beta\eta}$ t_2] iff there exist $v_1,\ldots v_n$ $(1 \leq n)$ s.t. $v_1=t_1$, $v_n=t_2$, and, for all $i < n$, $v_1=_\alpha v_{i+1}$ or v_i $RED_{1\beta}$ v_{i+1} [v_i $RED_{1\beta\eta}$ v_{i+1}].

t_1 $CONV_\beta$ t_2 [t_1 $CONV_{\beta\eta}$ t_2] iff there exist $v_1,\ldots v_n$ $(1 \leq n)$ s.t. $v_1=t_1$, $v_n=t_2$, and for all $i < n$, t_1 RED_β t_2 or t_2 RED_β t_1 [t_1 $RED_{\beta\eta}$ t_2 $RED_{\beta\eta}$ t_1].

$TERM=\{t: X_t \neq 0\}$.

It will now be shown that if t $CONV_{\beta\eta}$ u, then $X_t=X_u$, and a fortiori, that TERM is closed under $CONV_{\beta\eta}$.

A CL derivation $\underset{\sim}{D}$ is normal iff no sequent occurrence in $\underset{\sim}{D}$ is both the conclusion of a $\hat{\&}$I and the premiss of a $\hat{\&}$E. It can be shown by induction on the length of CL derivations that if $\Gamma \,||\!\!-\, A$ is derivable in CL, then there is a normal CL derivation which ends with $\Gamma \,||\!\!-\, A$. If $\underset{\sim}{D}$ is a normal CL derivation which ends with $A_1 \supset B_1,\ldots, A_n \supset B_n \,||\!\!-\, A \,\hat{\&}\, B$, induction on the length of $\underset{\sim}{D}$ yields the conclusion that the last inference of $\underset{\sim}{D}$ is a $\hat{\&}$I. It follows that the last inference of a normal CL derivation ending with $A_1 \supset B_1,\ldots,A_n \supset B_n \,||\!\!-\, A \supset B$ is not a $\hat{\&}$E.

LEMMA 4.1. If $A \supset B \in \mathrm{cl}(\{A_1 \supset B_1,\ldots,A_n \supset B_n\})$, then there exist $C_1 \supset D_1,\ldots,C_m \supset D_m \in \{A_1 \supset B_1,\ldots, A_n \supset B_n\}$ s.t. $A \supset B \in \mathrm{cl}(\{C_1 \supset D_1,\ldots,C_m \supset D_m\})$, $C_1,\ldots,C_m \in \mathrm{cl}(\{A\})$, and $B \in \mathrm{cl}(\{D_1,\ldots,D_m\})$.

Proof. By induction on the length of normal CL derivations ending with $A_1 \supset B_1,\ldots,A_n \supset B_n \Vdash A \supset B$. ($K\Vdash$, $C\Vdash$, and $W\Vdash$ imply that there is no loss of generality.)

LEMMA 4.2. If $P, x \models A, Q \vdash t \models B$ is derivable and $A \in \mathrm{cl}(\{C\})$, then $P, x \models C, Q \vdash t \models B$ is derivable.

Proof. By induction on the length of S_2 derivations.

LEMMA 4.3. If $P \vdash t_1 \models A$ is derivable and $t_1 \ \mathrm{RED}_{1\beta}\ t_2$, then $P \vdash t_2 \models A$ is derivable.

Proof. Induction on the complexity of t_1. Hyp. Ind. suffices if a proper part of t_1 is replaced. Otherwise, lemmas 4.1, 4.2, 3.1, and Cut$\vdash$ yield the desired conclusion.

LEMMA 4.4. If $\underset{\sim}{D}$ is an S_2 derivation ending with $P \vdash [u/x]t \models A$, $X = \{B\colon$ for some Q and v, $v =_\alpha u$ and $Q \vdash v \models B$ occurs in $\underset{\sim}{D}\}$, and $X \subseteq \mathrm{cl}(\{C\})$, then $P, y \models C \vdash [y/x]t \models A$ is derivable in S_2.

Proof. Induction on the complexity of t.

LEMMA 4.5. If $P \vdash t \models A$ is derivable and x is free in t, then P has the form $Q, x \models B, R$.

Proof. Induction on the length of S_2 derivations

LEMMA 4.6. If $\underset{\sim}{D}$ is an S_2 derivation which ends with $P, x \models A, Q \vdash t \models B$ and $P_1, x \models C, Q_1 \vdash t_1 \models B_1$ occurs in $\underset{\sim}{D}$, then $A = C$.

<u>Proof</u>. Induction on the length of $\underset{\sim}{D}$.

LEMMA 4.7. If x is free in t, $\underset{\sim}{D}$ is an S_2 derivation which ends with $P \vdash [u/x]t \models A$, and $X = \{B\colon$ for some Q and v, $v =_\alpha u$ and $Q \vdash v \models B$ occurs in $\underset{\sim}{D}\}$, then $X \neq O$.

<u>Proof</u>. Induction on the length of $\underset{\sim}{D}$.

For $P = x_1 \models A_1, \ldots, x_n \models A_n, y_1 \models B_1, \ldots, y_m \models B_m$ and $Q = x_1 \models C_1, \ldots, x_n \models C_n, z_1 \models D_1, \ldots, z_k \models D_k$, where $y_1, \ldots y_m$ are distinct from $z_1, \ldots, z_k$, let $P + Q = x_1 \models A_1 \mathbin{\hat{\&}} C_1, \ldots, x_n \models A_n \mathbin{\hat{\&}} C_n, y_1 \models B_1, \ldots, y_m \models B_m, z_1 \models D_1, \ldots, z_k \models D_k$.

LEMMA 4.8. If $t_1 \ \mathrm{RED}_{1\beta}\ t_2$ and $P \vdash t_2 \models A$ is derivable, then there is an R s.t. $R \vdash t_1 \models A$ is derivable.

<u>Proof</u>. Induction on the complexity of t_2. If a proper part of t_2 is replaced, Hyp. Ind. suffices. Suppose $t_2 = [u/x]t$.

If x is not free in t, then $X_u \neq O$. Let $Q \vdash u \models B$ be derivable. By $K\vdash$ and $\supset I$ $P \vdash \lambda xt \models B \supset A$ is derivable. By lemma 4.2 and $K\vdash$ $P + Q \vdash u \models B$ and $P + Q \vdash \lambda xt \models B \supset A$ are derivable, so $P + Q \vdash (\lambda xt)u \models A$ is derivable by $\supset E$.

If x is free in t, let $\underset{\sim}{D}$ be an S_2 derivation ending with $P \vdash [u/x]t \models A$ and let X be as in lemma 4.7. Lemma 4.7 implies that $X \neq O$. By lemma 4.4 $P, y \models B_1 \mathbin{\hat{\&}} \ldots \mathbin{\hat{\&}} B_n \vdash [y/x]t \models A$ is derivable, where $X = \{B_1, \ldots, B_n\}$. Hence, $P \vdash \lambda xt \models B_1 \mathbin{\hat{\&}} \ldots \mathbin{\hat{\&}} B_n \supset A$ is derivable by $\supset I$. Let $Q_1 \vdash v_1 \models B_1, \ldots, Q_n \vdash v_n \models B_n$ be the sequents of the form $Q \vdash v \models B$ s.t. $v =_\alpha u$ and $Q \vdash v \models B$ occurs in $\underset{\sim}{D}$. Since for all i $(1 \leq i \leq n)$, $v_i =_\alpha u$ it can be shown by induction on the complexity of $v_1,\ldots,v_n$ that $Q_1 \vdash u \models B,\ldots,Q_n \vdash u \models B$ are derivable. By suppressing elements of $Q_1,\ldots,Q_n$ which involve variables not free in u and applying lemmas 4.5 and 4.6, $K\vdash$, and $C\vdash$, it follows that $P \vdash u \models B_1,\ldots,$ $P \vdash u \models B_n$ are derivable. By lemma 3.1 $P \vdash u \models B_1 \hat{\&} \ldots \hat{\&} B_n$ is derivable, so $P \vdash (\lambda xt)u \models A$ is derivable by $\supset E$.

COROLLARY 4.9. If t_1 $RED_{1\beta}$ t_2, then $X_{t_1} = X_{t_2}$.

__Proof__. Immediate from lemmas 4.3 and 4.8.

LEMMA 4.10. If t_1 $RED_{1\eta}$ t_2, then $X_{t_1,P} = X_{t_2,P}$.

__Proof__. Induction on the complexity of t_1. If a proper part of t_1 is replaced, Hyp. Ind. suffices. Suppose $t_1 = \lambda x.tx$. Then $t_2 = t$.

$X_{\lambda x.tx,P} \subseteq X_{t,P}$ by η. Suppose $A \in X_{t,P}$. It will be shown by induction on the complexity of A that $A \in X_{\lambda x.tx,P}$. Since $X_{t,P} \subseteq cl(\mathfrak{J}_{\supset})$, it can be shown by induction on the length of CL derivations that A is not a propositional parameter.

If $A = A_1 \supset A_2$, the desired conclusion follows __via__ $K\vdash$, $\supset E$, and $\supset I$. If $A = A_1 \hat{\&} A_2$, apply lemma 3.1, Hyp. Ind., and lemma 3.1 again in order to complete the argument.

COROLLARY 4.11. If t_1 $RED_{1\eta}$ t_2, then $X_{t_1} = X_{t_2}$.

__Proof__. Immediate from lemma 4.10.

THEOREM 4.12. If t_1 $CONV_{\beta\eta}$ t_2, then $X_{t_1} = X_{t_2}$.

__Proof__. As was remarked in the proof of lemma 4.8, $=_\alpha$ causes no trouble, so the theorem follows from corollaries 4.9 and 4.11.

5. TERM IS THE SET OF $\beta\eta$-STRONGLY NORMALIZABLE TERMS

x is the *head of* x. λxt is the *head of* λxt. The *head of* tu is the head of t.

LEMMA 5.1. If t is β-normal, then the head of t is a variable or t.

Proof. Induction on the complexity of t.

LEMMA 5.2. If the head of t is a variable and $t \in$ TERM, then $X_t = \mathfrak{I}$.

Proof. Induction on the complexity of t.

THEOREM 5.3. If t is β-normal, then $t \in$ TERM.

Proof. Induction on the complexity of t, using lemmas 5.1 and 5.2 as required.

THEOREM 5.4. If t is $\beta\eta$-strongly normalizable (in the usual sense), then $t \in$ TERM.

Proof. The $\beta\eta$-strongly normalizable terms are the same as the β-strongly normalizable terms. Proceed by induction on the maximum number of β-contractions in a reduction of t to a β-normal term, using theorem 5.3 and theorem 4.12 as required.

In order to prove the converse of theorem 5.4, it suffices to show that every member of TERM is β-strongly normalizable. A method for proving this will now be explained.

$s, s_1, \ldots$ are to be sequents of S_1 and S_2. s_1 β-*reduces to* s_2 iff, for some P, t, u, and A, $s_1 = P \vdash t \models A$, $s_2 = P \vdash u \models A$, and t β-reduces to u (in the ordinary sense). s is β-*strongly normalizable* iff every β-reduction of s contains only finitely many β-contractions.

Let $s = P \vdash t \models A$ be derivable. If A is a propositional parameter and s is β-strongly normalizable, then s is computable. If $A = A_1 \supset A_2$ and, for every computable sequent $Q \vdash u \models A_1$, $P + Q \vdash tu \models A_2$ is computable, then s is computable. If $A = A_1 \hat{\&} A_2$ and $P \vdash t \models A_1$ and $P \vdash t \models A_2$ are computable, then s is computable.

Given this definition, it is easy to modify the arguments of Stenlund (1972, pp. 126-131) so as to prove that every derivable sequent is β-strongly normalizable. The converse of theorem 5.4 follows.

THEOREM 5.5. TERM = $\{t:\ t$ is $\beta\eta$-strongly normalizable$\}$

Proof. By theorem 5.4 and the method for proving the converse of theorem 5.4 which has just been described.

6. $\hat{\&}$ AS A CONNECTIVE

It was remarked in section 1 that $\hat{\&}$ behaves quite differently from $\&$. This will now be made apparent.

A is a theorem iff, for some t, $\vdash t \models A$ is derivable. This amounts to saying that A is a theorem iff A is realized by a closed member of TERM.

Given theorem 4.12 and 5.5, it is easy to show that the following formulas are not theorems: $p \supset. q \supset p \hat{\&} q$, $p \supset q \supset. p \supset r \supset. p \supset q \hat{\&} r$, $p \hat{\&} q \supset r \supset. p \supset. q \supset r$. On the other hand, the following sequents are derivable.

$$\vdash \lambda x.xx \models A \hat{\&} (A \supset B) \supset B$$

$$\vdash \lambda x \lambda y.xy \models (A \supset B) \hat{\&} (A \supset C) \supset. A \supset B \hat{\&} C$$

$$\vdash \lambda x \lambda y.xy \models A \supset B \hat{\&} C \supset. (A \supset B) \hat{\&} (A \supset C)$$

$$\vdash \lambda x \lambda y.xy \models A \supset C \supset. A \hat{\&} B \supset C$$

$\vdash \lambda x\lambda yx \models A \mathbin{\hat{\&}} B \supset . A \supset B$

$\vdash \lambda x\lambda y.xyy \models A \supset (B \supset C) \supset . A \mathbin{\hat{\&}} B \supset C$

$\vdash \lambda xx \models A \mathbin{\hat{\&}} B \supset A$

$\vdash \lambda xx \models A \supset A \mathbin{\hat{\&}} A$

$\vdash \lambda xx \models A \mathbin{\hat{\&}} B \supset B \mathbin{\hat{\&}} A$

$\vdash \lambda xx \models A \mathbin{\hat{\&}} (B \mathbin{\hat{\&}} C) \supset (A \mathbin{\hat{\&}} B) \mathbin{\hat{\&}} C$

Since the meaning of $\hat{\&}$ is reasonably clear (to claim that $A \mathbin{\hat{\&}} B$ is to claim that one has a reason for asserting A which is also a reason for asserting B), it would obviously be of interest to figure out how to add $\hat{\&}$ to intuitionist logic and then consider the analysis of intuitionist mathematical reasoning in the light of the resulting system.

FOOTNOTES

1. This is crude, but it will suffice to motivate the rules and axioms of the system S_1. Clearly, it would be nice to be able to replace this sort of talk by a pleasant realizability interpretation. For those who believe that all is syntax the results proved here will in effect do that. It is in fact possible to produce a set theoretically based realizability interpretation for the formal machinery employed in this paper, which should be some comfort to those who do not believe that all is syntax. But that interpretation is far from pleasant, and this paper is too small to contain it.

2. One needs a lemma to the effect that in an S_1-η derivation no sequent need ever be both the conclusion of an $\hat{\&}I$ and the premiss of an $\hat{\&}E$ (cf. the remarks preceding lemma 4.1 and the proof of that lemma), but it is easy to prove this by induction on the length of S_1-η derivations.

3. η is the real culprit here. If attention were restricted to S_1-η, then it would suffice to control $\hat{\&}I$ and $\hat{\&}E$ in the way explained in note 2.

4. $\wp\mathfrak{J}$ = the powerset of $\mathfrak{J}$.

5. O = the empty set.

ACKNOWLEDGEMENTS

The research reported on here was supported in part by NSF grants SOC 7707548 and SOC 7906386.

Earlier drafts of this paper were greatly improved due to the suggestions of the editors and the referee and to correspondence with M. Coppo. Peter Andrews, Don Cohen, Eve Cohen, David Jefferson, Dale Miller, and Oswald Wyler also provided much help in the preparation of the final draft by pointing out various sins of commission and ommision in the course of a seminar presentation of the first draft.

REFERENCES

M. Coppo and M. Dezani-Ciancaglini, "A New Type Assignment for λ-terms", Archiv für Mathematische Logik und Grundlagen Forschung, vol. 19 (1978), pp. 139-156.

M. Coppo and M. Dezani-Ciancaglini, "An Extension of Basic Functionality Theory for λ-calculus", Notre Dame Journal of Formal Logic (forthcoming July 1980).

M. Coppo, M. Dezani-Ciancaglini, and P. Sallé, "Functional Characterization of some Semantic Equalities inside λ-calculus", Lecture Notes in Computer Science, vol. 71 (1979), pp. 133-146.

H. B. Curry and R. Feys, Combinatory Logic vol. 1, North Holland, 1958.

P. Sallé, "Une Extension de la Théorie des Types en λ-calcul", Lecture Notes in Computer Science, vol. 62 (1978), pp. 398-410.

S. Stenlund, Combinators, λ-terms and Proof Theory, D. Reidel, 1972.

A SURVEY OF THE PROJECT AUTOMATH

N.G. de Bruijn

Department of Mathematics,
Eindhoven University of Technology
P.O.Box 513, 5600MB Eindhoven
The Netherlands

Dedicated to H.B. Curry on the occasion of his 80th Birthday

1. PURPOSE OF THIS SURVEY

Thus far, much about AUTOMATH was written in separate reports. Most of this work has been made available upon request, but only a small part was published in journals, conference proceedings, etc. Unfortunately, a general survey in the form of a book is still lacking. A short survey was given in de Bruijn (1973), but the present one will be much more extensive. Naturally, this survey will report about work that has been done, is going on, or i planned for the future. But it will also be used to explain how various parts of the project are related. Moreover we shall try to clarify a few points which many outsiders consider as uncommon or even weird. In particular we spend quite some attention to our concept of types and the matter of "propositions as types" (section 14). Finally the survey will be used to ventilate opinions and views in mathematics which are not easily set down in more technical reports.

Some further material of a general nature can be found e.g. in de Bruijn (1968b); Jutting (1977). For those who have not read anything about the project, this survey cannot pretend to give more than a vague idea of the languages. For getting a better idea, de Bruijn (1971b), van Daalen (1973) may be recommended; van Daalen (1973) gives a very precise definition of AUT-QE, one of the most prominent members of the family (see sections 9,13).

2. PURPOSES OF THE PROJECT AUTOMATH

The project was conceived in 1966; the first report was de Bruijn (1968a). The idea was to develop a system of writing entire mathematical theories in such a precise fashion that verification of the correctness can be carried out by formal operations on the text. Here "formal" means: without "understanding" the "meaning", and therefore it has to be possible to instruct computers how to check the correctness. Indeed, the fact that we do have computers will be one of the reasons why our generation has better chances than those who tried to have similar claims in the past, like Leibniz, Peano and Hilbert. Even if we do not actually *use* computers, they are there to set the standard of what is "formal" verification.

In the next three sections we discuss motivations for the project: checking, understanding and processing. The first two motives seem to favour the choice of a system of a very general nature, not necessarily tied to today's ideas of formalizing mathematics in terms of classical logic and set theory.

3. CHECKING

Most mathematicians can very well check themselves what they read and write. Nevertheless only a small portion of mathematical literature is absolutely flawless. Moreover, human checking seems to be a social affair too: mathematicians put trust in something since they think or know that other mathematicians have checked it.

Very meticulous checking is definitely unpopular. The thing we have in mind puts quite a burden on those who write the mathematics to be checked. They have to justify every little step extensively. It is only after this that a computer can do the final checking and guarantee the correctness.

We mention two cases were checking may be important. The first one is for things which are very hard and condensed, and where there is little intuitive or experimental support. The second one is for long and tedious proofs which form very long chains of very elementary steps. Such things may occur in combinatorial arguments, but, more important, in the large amount of work that has to be done to check the semantic correctness of large computer programs or machine designs.

Checking may actually be carried out in man-machine cooperation. This may also mean that, at least temporarily, parts of the checking may be omitted if they refer to things we are absolutely sure of.

Many errors in mathematics are made at the interfaces between theories. Therefore, we want to do the checking in a system that embraces all the theories involved. For example, if we want to check that the regular 17-gon can be constructed by ruler and compass, we have to be able to formulate the rules of geometric constructions into our system.

4. UNDERSTANDING

Formal systems help us to understand mathematics already by the mere fact that they force us to subdivide mathematical discussion into (i) language (ii) metalanguage and (iii) interpretation. The role of the latter is often underestimated. Those who say that mathematics *is* set theory, usually disregard the fact that they handle an extensive system of interpretation which is almost completely intuitive. Quite often it is just the interpretation that means "understanding" mathematics. Therefore we want a system that checks as much as possible of what we can actually say. (This is as far as we can go: we cannot expect a machine or a person to check what is in the back of our minds). Our system should check a kind of language that comes as close as possible to what we write in ordinary mathematics.

If we want to understand mathematics we also have to get insight into the roles of axioms, definitions, proofs, theorems. We cannot expect to get such an insight from a basic theory that has been built up itself with axioms, definitions, proofs and theorems. It is much better to have a foundation that is nothing but a set of rules for manipulating language. On such a foundation we can build logic and mathematics, possibly with the use of axioms. There is nothing against axioms, but we should be free to accept them or to reject them. Axioms should not be tied to the fundamentals of our system.

Another thing that a good language may help us to understand is the structure or the complexity of an argument. The text may reveal analogies in the structure of arguments, and classification of their inherent difficulty. As to the classification of difficulty we mention that a very useful borderline between "elementary" and "higher" mathematics is that elementary mathematics is the part of mathematics that can be expressed without lambda calculus. In other words: "elementary" is what can be said in PAL (see section 11).

5. PROCESSING

The fact that a machine can read, check and store the mathematics we produce, can have several advantages. One of these is that we can be absolutely sure that two mathematicians use the same theorem with exactly the same conditions. But a machine can also process its contents for answering questions. Examples: (i) produce a glossary of a text, (ii) find out in a given argument whether a given axiom does or does not play a (direct or indirect) role, (iii) print all notions and arguments that are needed to understand a given theorem, omit everything that is irrelevant to it.

6. WHAT KIND OF MATHEMATICS CAN WE DO?

The AUTOMATH system is like a big restaurant that serves all sorts of food: vegetarian, kosher, or anything else the customer wants. The languages are not tied to any logical system: hardly any logic has been built in. Admittedly there are basic notions of functionality and typing, but these need not be used the way they seem to be intended for. Those who want to say that a function is a subset in a certain cartesian product, can say it in AUTOMATH, but the restaurant also caters for those who want to describe mathematical functions by means of the functionality available in the language itself. Those who reject the axiom of choice or the excluded middle can use the system, as well as adepts in "New Math" and those who see "truth" as a matter of checking zeros and ones in truth tables.

Nevertheless some customers are better served than others. The best-served are those who try to keep close to the way mathematicians actually talk and think. They can use the types for doing typed set theory, the context structure to represent their ordinary way of reasoning (natural deduction), and the built-in functionality for describing their functions.

For typed set theory and natural deduction in relation to AUTOMATH, see de Bruijn (1975a), Nederpelt (1977). Formal Zermelo-Fränkel set theory was written in AUT-68 (cf. section 9) by van Daalen (1970). For a large piece of mathematics described in the "natural" style, we refer to the Landau translation (see sections 20 and 25).

7. BOOKS AND CONTEXTS

We write our mathematics in books, consisting of sequences of lines. Each line is written in some context.

We use the word "context" in a restricted sense. At each point of a mathematical discussion we can consider

(i) The set of assumptions which are considered to be valid at that point.

(ii) The set of variables which are "alive" at that point.

(iii) The set of all notions that have been developed previously (either by definitions or by taking them as primitives). Many people will say that the context is (i)+(iii), and disregard (ii) (their idea is that there is an infinite pool of variables which are always available).

We shall use the word context differently, taking it to be described by (i)+(ii). There is no reason for us to specify (iii), since it follows from the given order of the lines in the book. This is not true for (i): assumptions can be both introduced and discarded. And as to (ii): our point to take this as part of the context, is the fact that the variables will be typed. These types may be expressed by means of "older" variables but their construction may also depend on the fact that the assumptions of the context are valid (i.e. the types may be defined by expressions containing things that were defined only under these assumptions).

Similarly, the assumptions may be expressed in terms of variables belonging to the context. In this respect assumptions and variables play the same role in the context. They can appear in any order. Let us give an informal example of a context:

"Let n be a natural number. Let P be a point of R_n.
Let Q be a point of R_n. Assume $d(P,Q) > n$."

This context contains three variables n,P,Q and one assumption. We say that this context has length 4. Things of the kind (iii) are "natural number", "point of R_n", "d", ">".

In a mathematics book we can indicate the context of every line. There is a special kind of lines that serve to define new contexts (these lines are called *block openers*). Examples: "Let n be a natural number". "Assume $d(P,Q) > n$". Block openers are placed in a context too.

A context can be seen as a sequence of block openers, arranged in the order in which they appear in the book. If these context lines are labeled $A_1,\ldots,A_n$, then the context of A_n is $A_1,\ldots,A_{n-1}$. Therefore the context $A_1,\ldots,A_n$ is adequately described by mentioning A_n only: looking up line A_n in the book will reveal A_{n-1}, etc.

The word "block opener" suggests the usual situation that assumptions are taken to be valid during a sequence of consecutive lines, and that validity regions of assumptions are nested intervals. These things will not be generally assumed however. A context can shrink for a while, and be picked up later.

8. DEFINITIONAL LINES AND PN-LINES

What kind of material can be written in a context (apart from block openers that extend the context)? It will turn out that we can get away with two things: definitional lines and PN-lines. In the first case we have a new identifier (symbol or word), and an expression (in terms of old identifiers and material from the context); the line is interpreted as the definition of the new identifier. In a PN-line, however, no expression is given, but the symbol PN is written instead. The interpretation is that the identifier is introduced as a primitive symbol. In section 14 it will be explained how some of the definitional lines can be interpreted as theorems with proofs and some of the PN-lines as axioms.

9. THE LANGUAGE FAMILY

As basic language we take SEMIPAL. It is not able to handle mathematics, but just intended to give a record of how things are expressed in terms of others. The contexts in SEMIPAL are sequences of untyped variables. Apart from the block openers there are definitional lines and PN-lines. The expressions are composed of identifiers and variables. If the context is $x_1,\ldots,x_n$, the new identifier is p, then the line is written as something like

$$x_1,\ldots,x_n * p := f(g(x_1,a),h(x_1)). \tag{9.1}$$

On the right we have an example of an expression. In order to explain what we intend with this line, it is better to write $p(x_1,\ldots,x_n)$ instead of p; the interpretation is that p is introduced as a function of n variables. The expression on the right is assumed to be correct, i.e. (i) each non-variable identifier has been introduced previously in the book, with a context length equal to the number of subexpressions it has in (9.1), (ii) the variables occurring in (9.1) all belong to the context $x_1,\ldots,x_n$.

SEMIPAL can be extended in two ways.

(i) by admitting lambda expressions (λ-SEMIPAL).

(ii) by attaching a type to every expression, taken from a fixed finite set of types. Let us call this PAL-FT (PAL with fixed types).

We can go beyond (ii):

(iii) by admitting the introduction of type variables and of primitive types. This will be called PAL ("Primitive Automath Language").

The combination of PAL and λ-SEMIPAL leads to AUT-68 (for a long time this was called AUTOMATH), and, a little beyond it, AUT-QE. Let us write A ≤ B if every correct book in language A is also correct in language B. Then we have PAL ≤ AUT-68 ≤ AUT-QE.

A different extension of PAL is J.Zucker's AUT-PI (see section 22).

The language AUT-SL (single line AUT, see de Bruijn (1971a), Nederpelt (1973) has been created mainly in order to get a streamlined language theory. It is a very general higher-order language, obtained by giving up all restrictions on abstraction, and admitting all numbers 0,1,2,... as degrees (see section 11). Once this has been done, we can write PN's as block openers (cf. section 16), eliminate all definitional lines, and thus obtain a complete book in the form of a single line.

10. ABBREVIATION SYSTEM

In SEMIPAL we have a simple abbreviation system that can be maintained throughout the language family. If p was introduced by (9 1), say, then in later expressions p is allowed to have fewer than n subexpressions. The missing subexpressions are just supplied by adding $x_1, x_2, \ldots$ on the left. For example: if $E_3, E_4, \ldots, E_n$ are expressions, then $p(E_3, \ldots, E_n)$ is an abbreviation for $p(x_1, x_2, E_3, \ldots, E_n)$. (So $p(E_3, \ldots, E_n)$ can only be used in a context containing the first two variables of the context of (9.1)).

Quite a different kind of abbreviation, (again for all languages of the family) lies in the *paragraph system* (for a description see Jutting (1977). It has the practical advantage that names for identifiers (e.g. common letters like x,a,...) can be used over and over again. The book is divided into sections, subsections, sub-subsections,... (all called paragraphs). If we mention an identifier we mean the one that was introduced in the smallest surrounding paragraph; if we want to refer to a different identifier with the same name, we have to mention its paragraph number.

11. TYPING AND DEGREES

We begin with a language with fixed types. Let us call it PAL-FT. We start from SEMIPAL, and we attach a type (taken from the given set) to every variable, to every identifier and to every expression. The rules are obvious: if we form an expression by substituting expressions $E_1,\ldots,E_n$ for $x_1,\ldots,x_n$ in $p(x_1,\ldots,x_n)$, then for each i the type of E_i should equal the one of x_i, and $p(E_1,\ldots,E_n)$ gets the same type as p. The type can be written at the end of each line of the book (including block openers). As a seperation mark we can use the semicolon (we also write $p : \tau$ in the metalanguage in order to say that p has the type τ).

Let us pass to PAL. We introduce a new symbol <u>type</u>, and say that τ : <u>type</u> for every type we had thus far. Let us admit this new kind of typing for block openers as well as for PN-lines. Then by obvious extensions of our rules, we can get the new types in the definitional lines too. We do not need the collection of fixed types anymore: the same effect can be obtained with PN-lines "τ := PN : <u>type</u>".

Since PN-lines can be written inside a context, we can get big expressions typed by <u>type</u> (i.e. we get types depending on a number of parameters).

Let us say that <u>type</u> is an expression of degree 1; if E : <u>type</u> we say that E has degree 2; if F : E and E : <u>type</u> we say that F has degree 3.

In the languages mentioned in section 9 the degrees are restricted to 1,2,3. There would not be any harm in admitting higher degrees, but the description of present-day mathematics does not seem to require more than three degrees. There is a suggestion of using degree 4 in de Bruijn (1974b), but what is done with it might also be done with lower degrees by slight modifications of the language.

The typing rule of PAL-FT is to be modified in PAL: the type of $p(E_1,\ldots,E_n)$ is to be what we get if in the type of $p(x_1,\ldots,x_n)$ we substitute E_1 for $x_1,\ldots,E_n$ for x_n. And we require that the type of E_i is "definitionally equal" (see section 18) to the one we get by that same substitution in the type of x_i (the latter type does not contain $x_i,x_{i+1},\ldots,x_n$).

12. ADDING THE LAMBDA CALCULUS

In section 9 we announced λ-SEMIPAL as what we get from PAL by admitting λ-expressions as expressions. (This language has never been used or studied in the project; it is only mentioned here as a resting-point in the discussion). If E is an expression containing the variable x, then $\lambda_x E$ is an expression in which x is no longer a variable but a dummy. The passage from E to $\lambda_x E$ is called *abstraction*. The interpretation is that $\lambda_x E$ is a function, which at any point p has as its value the expression we get if in E we replace x by p (the result of this substitution is written in the metalanguage as $[\![x \mid p]\!]\, E$).

The counterpart of abstraction is called *application*. We write p{f} for the thing that is interpreted as the value of the function f at the point p. (The usual way of writing fp or f(p) is inconvenient since abstraction is written on the left, and it happens so often that abstractions and applications are tied together in pairs.

A crucial role in the metalanguage is played by *β-reduction*. This means reducing $\{p\}\,\lambda_x E$ to $[\![x \mid p]\!]\, E$, in accordance with the interpretation. Less important is *η-reduction*, reducing $\lambda_x\,\{x\}E$

to E in cases where E does not contain x.

In λ-SEMIPAL we have two different ways to describe the relation between a function f, a value p of the variable, and the value of the function at that point. One way is the application {p}f, the other one is by means of what we shall call *instantiation*. If f is an identifier introduced in the context x (either by a definitional line or by a PN-line or by a block opener) then we can use the expression f(p) in later lines. This feature of the language has disadvantages (two ways of writing, with the same interpretation) and hardly any advantage: instantiation does not do what application cannot do. This will be different in typed languages: the scopes of instantiation and application overlap, but none of the two scopes is contained in the other.

13. ADDING TYPED LAMBDA CALCULUS TO PAL

We first say that the word "typed" in the title does not refer to fixed types like in PAL-FT of section 11. We shall admit type variables, and lambda expressions as types. Therefore we get beyond what is usually called typed lambda calculus.

The typed lambda expressions we want to add to PAL are of the form $\lambda_{x:A}B$. The B may contain x as a variable, and it has to be a legitimate expression under the assumption that the type of x is A. In the metalanguage we speak of "abstraction over A" or "abstraction of B over A".

The subscripted notation $\lambda_{u:U}$ is hard to print in the many cases where U is an expression containing further λ's. Therefore we always write [u : U] instead of $\lambda_{u:U}$.

There are various possibilities to play the game. For a survey we refer to de Bruijn (1974a). In particular we have to decide what degrees for A and B we admit. Both in AUT-68 and AUT-QE we admit abstraction over A's of degree 2 only. In AUT-68 the abstracted expression B can have degree 2 or 3, in AUT-QE B can have degree 1,2 or 3. The typing rule for λ-expressions in AUT-QE is roughly this: if in the context γ extended by x:A we have

B(x):C(x), then in the context γ we have [x:A]B(x):[x:A]C(x). In AUT-68 this is different if B has degree 2. If B(x) : type then AUT-68 obtains [x:A]B(x) : type.

In AUT-QE the "quasi-expressions" (like [x:A]type) seem strange, but once one gets accustomed to them they turn out to be quite natural and enjoyable. They allow applications {a} f if we know f:P,P:[x:A]type and a:A.

There is a rule in AUT-QE that increases the power of the language. The rule is called *type inclusion*. If we have a typing like T : [u:U][v:V]type we say that the typing T : [u:U]type and T : type are also acceptable (acceptable in the sense of the rules for instantiation and application). Expressed superficially: everything we say for arbitrary types can be used for function types too.

Actually we can take three decisions about type inclusion. It can be forbidden (like in AUT-SL), allowed (like in AUT-QE) or prescribed (like in AUT-68). Prescribing type inclusion means that the abstractions in front of type have to be skipped.

In AUT-68 typings are unique in the following sense. If in some context both $A{:}B_1$ and $A{:}B_2$ are correct, then B_1 and B_2 turn out to be definitionally equal (see section 18). In AUT-QE this holds with the exception of type inclusion. But this is just a matter of phrasing the language definition. We can also say that typing is unique but that the typing rule is liberalized (cf. "mock typing" in de Bruijn (1974a)).

If A : type and B : type we are able to say "let f be a mapping of A to B" by means of a block opener "f : [x : A]B". This shows that in the typed language the lambda calculus can do what instantiation cannot do (cf. section 12). On the other hand, by instantiation we are able to handle block openers like "A : type", and the functional relationships expressed in this context cannot be expressed by abstraction, at least not in languages (like AUT-68, AUT-QE) that forbid abstraction over expressions of degree

1 (e.g. over <u>type</u>).

14. USE OF TYPING FOR REASONING

The fact that PAL and its descendants AUT-68 and AUT-QE can be used for mathematical reasoning depends on the idea of *propositions as types*. Roughly it means that if p is a proof for a proposition, we write it as a typing p:P. This principle goes back to Curry and Feys (1958), and was elaborated by Howard (1969), Prawitz(1971), Girard (1972), Martin Löf (1973). Completely independently of these developments it appeared in de Bruijn (1968a,b).

Treating propositions as types is definitely not in the way of thinking of the ordinary mathematician, yet it is very close to what he actually does. We shall try to explain this presently.

Assume that our book contains the following theorem (described informally), for some given functions ϕ,ψ:

"Theorem 1. Let x be a real number. Assume $\psi(x) > 1$. Let n be an integer. Assume $\phi(x) > x^n$. Then $\psi(x) > n$".

We want to apply this later, with x=q, n=5, and want to conclude $\psi(q) > 5$. We have to convince ourselves that the conditions are satisfied. To this end we write a proof for $\psi(q) > 1$ and label this result as (1). And we write a proof for $\phi(q) > q^5$ and label that result as (2). Now we claim to apply the theorem, providing in this order q, (1), 5, (2). So the (1) and (2) are treated on a par with the names (of "objects") q and 5. Is (1) to be considered as a name for the proposition $\psi(q) > 1$? No, the application of the theorem is not legitimate because of the existence of the proposition $\psi(q) > 1$, but because of its being proved. So consider the reference (1) as a reference to a proof of $\psi(q) > 1$. Let us try to explain our application to a machine that knows Theorem 1. The machine wants to check (i) that q is a real number, (ii) that (1) is a statement that $\psi(q) > 1$ has been proved, (iii) that 5 is an integer, (iv) that (2) refers to a proof of $\phi(q) > q^5$. We only need to change a few words in order to get: q is a real number, (1) is a proof of $\psi(q) > 1$, 5 is an integer, (2) is a proof of

$\phi(q) > q^5$. All together, we have a proof of $\psi(q) > 5$.

The parallelism between proofs and "ordinary" mathematical objects gets even stronger if we realize that many objects are defined conditionally only. If we define a function f for x real, $x > 1$ then the use of the value of the function at a point requires (i) that point (a real number), (ii) a proof that the real number is >1. Now the value of the function is an object, and it depends on an object and a proof. So proofs may depend on objects and objects may depend on proofs. One might say that we have been confusing "proofs" with "references to proofs" or "names of proofs". But in informal talk we make the same switchings from "objects" to "names of objects". There is not much of a point in arguing whether proofs are as real or more real than objects. Quoting Wittgenstein's "Don't ask for the meaning, ask for the use", we must say that as far as the use is concerned, the parallelism is complete.

In the above example, the proof of $\psi(q) > 5$ is a single-step proof. In PAL it is expressed in a line

Theorem 2 := Theorem 1(q, (1), 5, (2)) : P

where P in some way represents the proposition $\psi(q) > 5$ (or rather the type of proofs of that proposition). The term "single-step proof" means that we only have to quote. It would become a multi-step proof if (1) was not available directly in the book, but (1) had to be constructed on the spot, again by substituting things in the name of the proof of a theorem, like "lemma 3(q,q)" in

Theorem 1(q, lemma 3(q,q), 5, (2)) : P.

In this way arguments of several steps can be condensed in a single line.

Let us illustrate the principle "propositions as types" by how it works for implications. Let p and q be propositions. Having a proof of the implication $p \to q$ can be interpreted as this: we have a procedure by which we are able to give a proof of q for every customer who might present us a proof of p. That is, our procedure is a function that maps the set of all proofs of p into proofs of q. Using our terminology of context, we can say that in the context "x : proof(p)" (representing "let x be a proof of p") we can write a line

f := ... : proof(q).

By the abstraction rule of AUT-68 or AUT-QE we get, outside the x-context (see section 13 for the notation),

[t : proof(p)] f(t) : [t : proof(p)] proof(q).

Hence [t : proof(p)] proof(q) acts as the proof type of the implication.

15. USING TWO EXPRESSIONS OF DEGREE 1.

For various reasons it is attractive to introduce a symbol _prop_ of degree 1 that behaves exactly like _type_, but with different interpretation. If A : B, B : _type_ then A is the name of an object of type B, and if C : D, D : _prop_ then C is the name of a proof for the proposition expressed by the proof type D.

One reason to make the distinction between _type_ and _prop_ is to give an easier insight into the interpretations, but there are also more essential reasons for making the difference. One of the forms of the logical double negation axiom, written by means of "_prop_", turns into the axiom about Hilbert's ε-operator if we replace _prop_ by _type_. So if we want to do classical logic and do not want to accept the axiom of choice, we need some distinction. It should be mentioned, however, that introduction of _prop_ is not the only way out of this difficulty. (Another way is to create a primitive type called "bool" (for boolean) and for every boolean b a primitive type "proof type of b").

Another suggestion to profit by treating _type_ and _prop_ differently, is "proof irrelevance" (section 24).

We can now give a survey of the various kinds of lines involving _prop_. First, block openers "x : _prop_" introduce propositional variables. PN lines "p := PN : prop" introduce primitive propositions. A definitional line "b := ... : prop" introduces an abbreviation for a more complex expression representing a proposition.

Next we take some P with P : prop. This P is interpreted as the proof type of a proposition. Now the block opener x : P states the proposition as an assumption. The PN-line u := PN : P is interpreted as stating the propositions as an axiom. The definitional line v := E : P states the proposition as theorem. The expression E represents the proof, and v is a name for the proof. The theorems themselves do not get names. In order to quote a theorem it suffices to quote a name for the proof.

Contexts are sequences of block openers like

$$x_1 : A_1, \ldots, x_n : A_n.$$

At the places where A_j : prop the interpretation is that x_j is the name of the assumption, at places where A_j : type the x_j is a variable. And, of course, there can be places where A_j = type or A_j = prop.

Especially in AUT-QE it is attractive to talk "prop-style", i.e. to suppress all propositions and talk about their proof types only. It turns out that there is hardly ever a necessity to talk about the propositions any more (talking about propositions is called "bool-style"). The example at the end of section 14 shows how this works: we can just *define* the proof type of the implication associated with the proof types P and Q by [t : P]Q.

The more often one does this kind of thing, the easier one forgets the original use of the word "proposition". This may explain why the AUTOMATH workers began to say prop instead of proof type. A consequence is that if P : prop they do not pronounce p : P as "p is a P" but as "p proves P".

16. AXIOMS vs. ASSUMPTIONS

If we have a PN-line in an empty context there is no harm in replacing it by an assumption. The name of the assumption will be a part of every context in the sequel. Taking it as an assumption gives more flexibility, since axioms are things we can never get rid of (unless we start a new book) and assumptions can be discarded if we wish.

If a PN-line is written in a non-empty context we can sometimes, but not always, replace it by an equivalent axiom in the empty context (and next replace it by an assumption). Whether this is possible depends on the degrees involved in the context as well as on the degree of the type of the PN-line, both in connection with the abstraction rules of the language. In AUT-SL, the most liberal language of the family, all PN's can be eliminated this way.

17. DERIVATION RULES

In the AUTOMATH family there is no essential difference between logic and mathematics. Logical connectives can be taken as PN's or as defined notions, inference rules can be taken as axioms or as derived rules, and later applications of such rules have the same form as applications of mathematical theorems.

As an example we present the double negation law. Somehow we have an expression CON with CON : <u>prop</u>. It has the following interpretation: if in some context we have an expression p with p : CON, then "we have a contradiction" (one can even say that p *is* a contradiction). In the context P : <u>prop</u> we next define NON(P) (by means of a definitional line) as [x : P]CON. The "double negation law" can now be written as follows:

[P : <u>prop</u>] [y : NON(NON(P))] * dbng := PN : P.

To the left of the asterisk the context is indicated: "let P be a <u>prop</u>, let y be a proof of the double negation". In this context we postulate the truth of P. The identifier "dbng" is chosen as the name of the law.

18. TWO KINDS OF EQUALITY

There is (already in SEMIPAL) a notion of definitional equality between expressions. The notion plays a central role in language theory. In typed languages it is essential already in the language definition (see the end of section 11). Definitional equality is generated by δ-reductions (δ-reduction means elimination of some previously defined identifier, replacing it by its definition given in the definitional line) and the β- and η-reductions of the lambda calculus.

In our languages no facilities have been provided for talking in the book about definitional equality. It is hardly necessary, for if A and B are definitionally equivalent then at every place in the book A may be replaced by B without any argumentation. The kind of equality mathematicians do talk about is what we call *book equality*. It may be introduced by means of a PN (but there are also possibilities to *define* book equality), and its basic properties can be covered by axioms or theorems.

19. LANGUAGE THEORY

Language theory is about reductions (the δ-, β- and η-reductions mentioned in section 18), normal forms (i.e. expressions which do not admit reductions) and about the relation between correct expressions and their types ("correct" means: acceptable in the book). Important parts of the language theory were obtained in Jutting (1971), van Daalen (1973), Nederpelt (1973), de Vrijer (1975). The forthcoming Ph.D. thesis by D.T. van Daalen will cover all aspects of the language theory at least for AUT-68, AUT-QE and AUT-SL. The essential results are (in a rough formulation)

(i) The Church-Rosser theorem: If A and B are definitionally equivalent then there is an expression C such that both A and B can be reduced to C by sequences of reductions.

(ii) The normal form theorem: For every A there is a normal form N to which A can be reduced by a sequence of reductions; N is uniquely determined.

(iii) The strong normal form theorem: Every reduction sequence terminates (and for every A there is an upper bound to the length of the reduction sequences starting at A).

(iv) The closure theorem: If A is correct and if A reduces to B then B is correct.

We note that (ii) is not true for untyped lambda calculus. It is true, however, for the untyped language SEMIPAL (which has no lambdas).

20. VERIFICATION

One of the most important things in the project is that we expect machines to check the correctness of what humans have written. This would be an easy programming job if the language would require of the writer that every little application of the rules of the language should be indicated in the text. But this is out of the question: from experience we know that it would require texts which are hundreds of times longer than they are in our present system. We expect the machine to do much of the checking on its own initiative, not necessarily in the same way the text-writer might have had in mind.

The machine has to find out whether there is a sequence of applications of the language rules that motivates the correctness of a line of the book, once all previous lines have been checked. The results of language theory show (at least for SEMIPAL, PAL, AUT-68, AUT-QE, AUT-SL) that this is automatically decidable. Definitional equivalence of two expressions can be established by reducing both to their normal form and checking whether these are the same. But already in short books this may turn out to give a prohibitive amount of work (in particular it will duplicate much of the work done in checking previous lines). What we really want is a good *strategy* by which the machine can try to find a shorter way from one expression to the other, about as short as what may have been in the writer's mind.

The computer programs whose execution effectuate the verification of a book, are called *verifiers* or *checkers*. For AUT-68 and AUT-QE the verifiers operate satisfactorily. The checkings can be done on-line from a teleprinter. In some cases where the program's strategy seems to run into very much work, the machine may ask whether the writer really wants it. In most cases it turns out that the writer has made a mistake. It would not be sensible to require that the machine *proves* that a line is incorrect: such a proof might require evaluation of normal forms. Therefore, it is better to let the machine report if it has a serious difficulty. And on some rare occasions we may let the machine ask for a hint in what direction to search. Sometimes it may help the machine if we write a few extra lines in the book, just as if we are explaining mathematics to human readers. In general, if we condense two lines into one, then checking the condensed line may require more work then checking the seperate lines one by one.

Most of the work on verifiers was done by I.Zandleven (cf. Zandleven (1973) in the years 1971-1976. Later this work was continued by A.Kornaat and L.S. van Benthem Jutting. A very large part of the effort is just caused by the limitations of today's computer technology. The amount of information involved in handling moderate amounts of mathematics is so big that it has to be distributed efficiently over the various kinds of fast and slow memory, and checking a single line may require consultation of many remote parts of the book. The paragraph system (see section 10) plays a role in coping with these difficulties.

In handling substitution in lambda calculus it is often necessary to re-name dummies in substitution operations, in order to avoid "name clashes". In order to simplify this, namefree lambda calculus was developed de Bruijn (1972b), (1978a), where references to dummies are not indicated by name but by reference depth. This system lies at the root of today's verifiers.

As it was said before, the problem of how to handle large amounts of mathematics requires considerable effort in the design of the verifiers, but the matter of strategies is more essential. It is, of course, closely related to language theory. The closure theorem (section 19) is important: it saves much work, e.g. it saves checking types when doing β-reduction.

The essential difficulty of verification is also the essential difficulty of language theory. It is the fact that definitionally equal expressions are connected by chains $A_1, A_2, \ldots, A_n$ in which the reductions go either way: sometimes A_i reduces to A_{i+1}, sometimes A_{i+1} reduces to A_i.

Jutting (1977) gives some details about experiences with the checking of a relatively large text (viz. the translation of Landau's "Grundlagen", Landau (1930)). The coded version (Jutting (1976)) consists of about 5.10^7 bits. This may seem very large (may be 10 to 50 times as large as a direct encoding of the words and symbols Landau wrote himself), but it is still of the order of what a single cassette tape can contain.

21. AUTOMATIC THEOREM PROVING

Automatic theorem proving is a very hard subject. In order to be efficient it certainly requires clever adaption to the kind of problems it is applied to. Therefore it is very questionable whether it would profit much from AUTOMATH, with its claims for generality and adaptivity to human reasoning. Admittedly, our verifiers do automatic searching, and may establish definitional equalities the writer has not bothered to see through, but this is not the level of what is usually called automatic theorem proving.

Nevertheless one may think of building "attachments" to the verifier which find proofs of little gaps the writer might like to leave. This might be done completely outside the system (e.g. by consulting the computer's arithmetic unit or by checking tautologies by inspection of cases), but it can also be conceived that the machine, after finding its proof, writes it in AUTOMATH and checks it by its own verifier. An attachment of the latter type was built (as a student's exercise) by R.M.A.Wieringa. Given natural numbers p,q,r with pq=r, where p,q,r are presented in the binary number system, his program produces an AUTOMATH text proving pq=r. The number of lines is of the order of the number of digits we write down with ordinary pencil-and-paper multiplication.

Attachments of the first kind, working outside the system, can of course work very much faster, At least some of them will be very profitable, but the AUTOMATH group never worked in this direction. They rather did what others don't than what others do very efficiently already.

22. FURTHER LANGUAGE EXTENSIONS

There is a number of things that mathematicians find so self-evident that they do not see them as part of the structure of axioms, definitions, and theorems, but more as a part of their language. This is deceptive, of course (after all we seem to do a lot of mathematical work subconsciously), but nevertheless one can try to incorporate as much as possible in the language definition. To quote a few unrelated things: pairs, strings, set theoretical operations, equality, commutativity and associativity, mathematical induction. One might say that in AUT-68 and AUT-QE only two things have been implemented: functional relationship and typing. All the rest is left to the book-writers.

We never found much use for building mathematical induction into the language definition (it is done in some other constructive systems). The reason is that in our system we have books to write in, and for a thing like induction it is as easy to quote the rule from a book as to apply a language facility. But for some of the other subjects mentioned above, the use of language facilities would be very much shorter than quoting from the book.

Every extension may seriously complicate both language theory and verifier. It is not clear how far one should go. J. Zucker devised AUT-PI (Zucker (1975)) as a relatively mild adaption and extension of AUT-68. It is much easier to write than AUT-68. Zucker wrote an extensive manuscript "Real Analysis" directly in AUT-PI (it is not a translation of something that was written first in ordinary language on scrap paper). A few chapters were written by A.Kornaat, who also produced some harder material in AUT-PI, viz. the proofs of the equivalence of various forms of the axiom of choice.

AUT-PI uses some proper extensions (like facilities for handling pairs) and a number of things which are more in the line of fast notation. Much of this belongs to a system called AUT-SYNT (partly developed by I.Zandleven) which has facilities for operations on syntactic variables, strings, and telescopes (a telescope is a string of block openers with types, like $[x_1:A_1]\ldots$ $\ldots[x_n:A_n]$; the name comes from hand telescopes with tubes fitting into each other).

The work on language theory and verifier of these languages is unfinished.

One way to look at AUT-SYNT is that it is just an auxiliary language (like in de Bruijn (1972a)) that helps us to prepare an input text in a language like AUT-68 or AUT-QE, where language theory and verifier are on prettv safe grounds. It is likely that on the long run AUT-68 provided with AUT-SYNT input facilities will not be less adequate than some of the fancier languages, at least for classical mathematics.

23. IS THERE A NEED FOR HIGHER ORDER LANGUAGE?

As it was said before, AUT-68 is a first order language since there is no abstraction over expressions of degree 1. Yet this does not seem to be a serious limitation, since a few extra axioms in the book extend the power of the language. As an example, we mention how abstraction over <u>prop</u> can be mimicked. We start with an axiom in the empty context "bool := PN : <u>type</u>", and from now on expressions b with b : bool are interpreted as propositions. Next, in a context [x : bool] we take the axiom "proof := PN : : <u>prop</u>". (In older publications we wrote "TRUE" instead of "proof"). The effect is that for every proposition b the typing u : proof(b) will mean that u is a proof for b. Now we can mimic abstraction over prop. Instead of saying that f(p) holds for all p : prop, we say that f(proof(b)) holds for all b : bool, and the abstraction is now over something of degree 2.

The reports de Bruijn (1976), (1977), (1978 b) give suggestions how slight extensions of AUT-68, and how AUT-QE-NTI (AUT-QE without type inclusion) can be used for mimicking stronger languages.

24. PROOF IRRELEVANCE

This is a feature we might add to our languages if we are interested in classical mathematics only. The classical mathematician would find it even hard to understand what its counterpart "proof relevance" is. We give an example. If x is a real number, then P(x) stands for "proof of x > 0". Now we define "log" (the logarithm) in the context [x : real][y : P(x)], and if we want to talk about log 3 we have to write log(3,p), where p is some proof for 3 > 0. Now the p is relevant, and we have some trouble in saying that log(3,p) does not depend on p. This can be done by means of the general axioms for book equality, with the effect that in this case $\log(3,p_1)$ and $\log(3,p_2)$ are book-equal if both p_1 and p_2 are proofs 3 > 0.

Some time and some annoyance can be saved if we extend the language by proclaiming that proofs of one and the same proposition are always definitionally equal. This extra rule was called "proof irrelevance" in de Bruijn (1974 b). We of course do not want to have the similar feature for type.

25. MATHEMATICS PRODUCED IN AUTOMATH

As a test case for handling larger amounts of mathematics, Jutting (1976) gave a line-by-line translation of Landau's "Grundlagen" (Landau (1930)) into AUT-QE. His experiences are reported in Jutting (1977). Landau's book was chosen because it presents material of different kinds in a very constant style of presentation: the steps do not get bigger towards the end of the book. It would of course have been much easier to rewrite Landau's book first, so as to make it easier to translate, but it was our aim to show that AUTOMATH can cope with any kind of mathematics, not just the mathematics especially designed for it.

Another substantial piece of work is Zucker's "Real Analysis" (mentioned in section 22). And J.T.Udding writes (in AUT-QE) a new theory of real numbers based on an approach that avoids the repeated troublesome embeddings Landau had to go through.

Many smaller pieces of mathematics have been done by students. The experience is that in a period of 2 or 3 weeks a mathematics student (without any training in logic) is able to learn AUT-QE, produce a piece of text (possible using basic material already known to the computer), punch it, have it checked via a teleprinter, correct it, and get a final AUT-QE version. For an account of how a piece of mathematics is translated in several stages see Jutting (1973).

The easiest things to translate are very condensed and very abstract pieces of mathematics (Example: the proofs of equivalence of various forms of the axiom of choice did not become much longer than the original text). Hard subjects are those where (subconscious) "experience" comes in, like in analysis and combinatorics.

A very important thing that can be concluded from all writing experiments is the *constancy of the loss factor*. The loss factor expresses what we loose in shortness when translating very meticulous "ordinary" mathematics into AUTOMATH. This factor may be quite big, something like 10 or 20, but it is constant: it does not increase if we go further in the book. It would not be too hard to push the constant factor down by efficient abbreviations.

26. WORK IN PROGRESS

Apart from things discussed before, we mention a few sub-projects which are studied now or will be studied in the near future.

(i) Programming language semantics. This may become an important customer for AUTOMATH. The idea is, to write in a single book: definition of a programming language and of its semantics, the logic and mathematics involved, particular programs, and proofs for their semantics. The ideal situation is this: a computer that has to execute a program, reads it directly from that book, thus avoiding every kind of interpretation. R.M.A.Wieringa is working on a system proposed in de Bruijn (1975b).

(ii) A far reaching extension of lambda calculus is presented in de Bruijn (1978 c) and studied in Wieringa (1978). In ordinary lambda calculus we can interpret substitution and β-reduction as replacing end-points of trees by branches of trees. The extension in de Bruijn (1978 c) means that we can also break open some edge and paste a segment of a tree into it. These segments might represent strings or telescopes (see section 22). This kind of lambda calculus can be expected to be helpful to simplify both language definitions and verifiers.

(iii) In the spirit of the work of the AUTOMATH project, the project WOT was started. WOT is a dutch abbreviation and stands for the "mathematical vernacular", i.e. the strange mixure of words and formulas mathematicians use. The idea is to get to a purified form of WOT that can be used as a formal system for expressing mathematics. The foundations of mathematics have to become some kind of grammar for WOT. Thus far the only reports on WOT are in dutch, and are used in the training of mathematics teachers.

ADDITIONAL NOTE

After this paper was finished, Professor Dana Scott pointed out that section 14 should have mentioned H. Laüchli's work on the principle of propositions-as-types. Reference should have been made to his abstract "Intuitionistic propositional calculus and definably non-empty terms", J. Symb. Logic 30 (1965) p. 263, and to his paper "An abstract notion of realizability for which intuitionistic predicate calculus is complete" in: Intuitionism and Proof Theory, Proceedings of the Summer Conference at Buffalo N.Y. 1968, North Holland Publ. Comp. 1970, pp. 227-234.

REFERENCES

(Items marked AUT followed by a number, are privately distributed reports).

de Bruijn, N.G. (1968a). AUTOMATH, a language for mathematics. Eindhoven University of Technology, T.H.-Report 68-WSK-05. AUT 1.

de Bruijn, N.G. (1968b). The mathematical language AUTOMATH, its usage and some of its extensions. Symposium on Automatic Demonstration. IRIA, Versailles, Dec. 1968. Lecture Notes in Math., Vol. 125, pp. 29-61, Springer Verlag, 1970.

de Bruijn, N.G. (1971a). AUT-SL, a single line version of AUTOMATH. AUT 20.

de Bruijn, N.G. (1971b). AUTOMATH, a language for mathematics. A series of lectures at the Séminaire de mathématiques supérieures. Université de Montréal, June 1971. Les Presses de l'Université de Montréal, 1973. Lecture Notes by B.Fawcett.

de Bruijn, N.G. (1972a). Some abbreviations in the input language for AUTOMATH. AUT 25.

de Bruijn, N.G. (1972b). Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Kon.Ned.Akad.Wetensch.* Proc.Ser. A 75 (=Indag.Math. 34) 381-392 (1972).

de Bruijn, N.G. (1973). The AUTOMATH Mathematics Checking Project Proc.Symp.APLASM, Vol. I, ed.P.Braffort, Orsay, France. AUT 34

de Bruijn, N.G. (1974a). A framework for the description of a number of members of the AUTOMATH family. AUT 40.

de Bruijn, N.G. (1974b). Some extensions of AUTOMATH: The AUT-4 family. AUT 44.

de Bruijn, N.G. (1975a). Set theory with type restrictions. *In* "Infinite and Finite Sets", ed.A.Hajnal, R.Rado, Vera T. Sós, Vol. I, pp. 205-214, Coll.Math.Soc.János Bolyai 10 (1975).

de Bruijn, N.G. (1975b). The use of the language AUTOMATH for syntax and semantics of programming languages. AUT 41.

de Bruijn, N.G. (1976). Modifications of the 1968 version of AUTOMATH. AUT 45.

de Bruijn, N.G. (1977). Some auxiliary operators in AUT-PI. AUT 51

de Bruijn, N.G. (1978a). Lambda calculus with namefree formulas involving symbols that represent reference transforming mappings. *Kon.Ned.Akad.Wetensch.*Proc.Ser. A 81 (=Indag.Math. 40), 348-356 (1978).

de Bruijn, N.G. (1978b). AUT-QE without type inclusion. AUT 56.

de Bruijn, N.G. (1978c). A namefree lambda calculus with facilities for internal definition of expressions and segments. Eindhoven University of Technology, T.H.-Report 78-WSK-03 (1978).

Curry, H.B. and Feys R. (1958). Combinatory Logic, Vol. 1, North Holland Publishing Comp., Amsterdam 1958.

van Daalen, D.T. (1970). Verzamelingstheorie, de axioma's van Zermelo-Fränkel. AUT 15.

van Daalen, D.T. (1973). A description of AUTOMATH and some aspects of its language theory. Proc.Symp. APLASM, Vol. I, ed.P. Braffort, Orsay, France, 1973; reprinted in Jutting (1977).
Girard, J.Y. (1972). Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur, Thèse, Paris.
Howard, W.A. (1969). The formulae-as-types notion of construction, Manuscript 1969, published in this Volume.
Jutting, L.S. van Benthem (1971). On normal forms in AUTOMATH. AUT 23.
Jutting, L.S. van Benthem (1973). The development of a text in AUT-QE. Proc.Symp.APLASM, Vol. I, ed.P.Braffort, Orsay, France 1973. AUT 37.
Jutting, L.S. van Benthem (1976). A translation of Landau's "Grundlagen" in AUTOMATH, vol.1-5. AUT 60.
Jutting, L.S. van Benthem (1977). Checking Landau's "Grundlagen" in the AUTOMATH system. Doctoral thesis. Eindhoven University of Technology 1977, Mathematical Centre Tracts nr. 83, Math. Centre Amsterdam 1979.
Landau, E. (1930). Grundlagen der Analysis. First ed. Leipzig 1930, Fourth ed. Chelsea Publ.Comp., New York 1965.
Martin-Löf, P. (1973). An intuitionistic theory of types, predicative part. *In* Logic Coll. 1973, Rose and Sheperdson ed., Amsterdam 1975.
Nederpelt, R.P. (1973). Strong normalization in a typed lambda calculus with lambda structured types. Doctoral thesis, Eindhoven University of Technology, 1973.
Nederpelt, R.P. (1977). Presentation of natural deduction. Receuil des Travaux de l'Institut Math., Nouvelle Série, tome 2(10). Symp. Set Theory, Foundations of Math., Beograd (1977), pp. 115-126.
Prawitz, D. (1971). Ideas and results in proof theory. *In* Fenstad (ed), Proc. of the 2nd Scand.Logic.Symp., North Holland, Amsterdam 1971.
de Vrijer, R.C. (1975). Big trees in a lambda-calculus with lambda expressions as types. Symp. on lambda calculus and computer science theory. Rome March 1975. Lecture Notes in Computer Science, Vol. 37, pp. 252-271, Springer-Verlag 1975.
Wieringa, R.M.A. (1978). Een notatie-systeem voor lambda-calculus met definities. Master's thesis. Eindhoven University of Technology.
Zandleven, I. (1973). A verifying program for AUTOMATH. Proc. Symp.APLASM, Vol. I, ed.P.Braffort, Orsay, France, 1973. AUT 36.
Zucker, J. (1975). Formalization of classical mathematics in AUTOMATH. *In* "Colloque International de Logique (Clermont-Ferrand 18-25 Juillet 1975)", pp. 135-145, Ed. CNRS, Paris 1977.