Notre Dame Journal of Formal Logic
Volume XV, Number 1, January 1974
NDJFAM

COMBINATORY AND PROPOSITIONAL LOGIC

DAVID MEREDITH

The relationship between combinatory and propositional logic is dealt with at length in [1] and tangentially in [2]. The present paper adds nothing essentially new to previous results. It does, however, offer a straightforward procedure, which for any λ-expression in normal form will either lead to its propositional correspondent or determine that this is null. Section 1 presents the hypothesis upon which the correspondence between λ-expressions and propositional formulae is based; our translation procedure is described in section 2, and illustrated in section 3.

1 Hypothesis. In dealing with λ-expressions we assume Church's rules and conventions as given in [3]. With respect to propositional formulae, 'Λ' denotes the null class of formulae, and 'Γ' is used for C. A. Meredith's operator D: 'ΓPQ' denotes the most general result that can be obtained when *Modus Ponens* is applied with P, or some substitution in it, as major premiss, and Q, or some substitution in it, as minor premiss. '$\sim$' denotes correspondence between a λ-expression and a propositional formula. Our basic hypothesis is the following.

Hypothesis *Where L, M and N are λ-expressions, P, Q and R are propositional formulae, and Σ is an operation under the substitution rule which may be null.*

1. *Let $L \sim P$, then for M with no free variables in common with L, and all N, Q, R. If $M \sim Q$, $LM = N$, and $N \sim R$, then either $\Gamma PQ = \Sigma R$ or $\Gamma PQ = \Lambda$.*

2. *Let $N \sim \Lambda$, then for L, M with no free variables in common, and all P, Q. If $L \sim P$, $M \sim Q$, and $LM = N$, then $\Gamma PQ = \Lambda$.*

The need for the two cases under the first section

(a) $L \sim P$, $M \sim Q$, $LM = N$ and $\Gamma PQ = \Sigma R$ for Σ non-null
(b) $L \sim P$, $M \sim Q$, $LM = N$, $N \sim R$ and $\Gamma PQ = \Lambda$

is unfortunate but unavoidable. With respect to (a): if for $L \sim P$ we take

$$\lambda abcd\,.\,ac(bd) \sim CCpCqrCCsqCpCsr$$

Received March 9, 1971

and for $M \sim Q$ we take

$$\lambda ab \,.\, a \sim CpCqp$$

then while for N we have

$$\lambda abc \,.\, b \sim CrCpCqp$$

ΓPQ gives us

$$\Gamma PQ = CCqrCpCqp.$$

With respect to (b): if for $L \sim P$ we take

$$\lambda ab \,.\, a(ab) \sim CCppCpp$$

and for $M \sim Q$ we take

$$\lambda ab \,.\, a \sim CpCqp$$

then while for N we have

$$\lambda abc \,.\, a \sim CpCqCrp$$

ΓPQ gives us

$$\Gamma PQ = \Lambda\,.$$

2 λ-Translation Procedure. Our procedure requires three definitions and a set of derivation rules.

Definition 1 Let $E = \lambda x_1 x_2 \,.\,.\, x_n \,.\, X_1 X_2 \,.\,.\, X_m$ or $E = X_1 X_2 \,.\,.\, X_m$ (n, $m \geqslant 1$) be a λ-expression in principal normal form; let $F = \lambda y_1 y_2 \,.\,.\, y_p \,.\, Y_1 Y_2 \,.\,.\, Y_q$ (p, $q \geqslant 1$) be a well-formed component of E: then each part of E yields a *result* and an *equality* determined as follows.

1. For $\lambda x_1 x_2 \,.\,.\, x_n \,.$ or the null prefix, and for E, P_k is the result of $\lambda x_1 x_2 \,.\,.\, x_n \,.$ or the null prefix, P_l is the result of E, and $P_l = Cx_1 Cx_2 \,.\,.\, Cx_n P_k$ or $P_l = P_k$.

2. For $\lambda y_1 y_2 \,.\,.\, y_p \,.$ and for F, P_k is the result of $\lambda y_1 y_2 \,.\,.\, y_p \,.$, P_l is the result of F, and $P_l = Cy_1 Cy_2 \,.\,.\, Cy_p P_k$.

3. For $Z_i Z_j$ any well-formed component of E such that $Z_i Z_j \neq X_{m-1} X_m$ and $Z_i Z_j \neq Y_{q-1} Y_q$:

   a. If Z_i and Z_j are both elementary, P_k is the result, and $Z_i = CZ_j P_k$.
   b. If Z_i is elementary, and Z_j is not elementary, P_k is the result, and $Z_i = CP_l P_k$ where P_l is the result of Z_j.
   c. If Z_i is not elementary, and Z_j is elementary, P_k is the result, and $P_l = CZ_j P_k$ where P_l is the result of Z_i.
   d. If Z_i and Z_j are neither of them elementary, P_k is the result, and $P_l = CP_m P_k$ where P_l and P_m are the results of Z_i and Z_j respectively.

4. For $Y_{q-1} Y_q$ where P_k is the result of $\lambda y_1 y_2 \,.\,.\, y_p$:

   a. If Y_{q-1} and Y_q are both elementary, P_k is the result, and $Y_{q-1} = CY_q P_k$.

b. If Y_{q-1} is elementary and Y_q is not elementary, P_k is the result, and $Y_{q-1} = CP_lP_k$ where P_l is the result of Y_q.
c. If Y_{q-1} is not elementary and Y_q is elementary, P_k is the result, and $P_l = CY_qP_k$ where P_l is the result of Y_{q-1}.
d. If Y_{q-1} and Y_q are neither of them elementary, P_k is the result, and $P_l = CP_mP_k$ where P_l and P_m are the results of Y_{q-1} and Y_q respectively.

5. For Y_q $(q = 1)$ elementary, where P_k is the result of $\lambda y_1 y_2 \,.\,.\, y_p .$, P_k is the result, and $Y_q = P_k$.

6. For $X_{m-1}X_m$ where P_k is the result of $\lambda x_1 x_2 \,.\,.\, x_n$. or the null prefix:

a. If X_{m-1} and X_m are both elementary, P_k is the result, and $X_{m-1} = CX_mP_k$.
b. If X_{m-1} is elementary and X_m is not elementary, P_k is the result, and $X_{m-1} = CP_lP_k$ where P_l is the result of X_m.
c. If X_{m-1} is not elementary and X_m is elementary, P_k is the result, and $P_l = CX_mP_k$ where P_l is the result of X_{m-1}.
d. If X_{m-1} and X_m are neither of them elementary, P_k is the result, and $P_l = CP_mP_k$ where P_l and P_m are the results of X_{m-1} and X_m respectively.

7. For X_m $(m = 1)$ elementary, where P_k is the result of $\lambda x_1 x_2 \,.\,.\, x_n$. or the null prefix, P_k is the result, and $X_m = P_k$.

Our derivation rules are the normal rules governing identity, together with three special rules.

Rule 1. Where $I, J, K, L, M \neq \Lambda$, if $I = CJK$ and $I = CLM$, then $J = L$ and $K = M$.

Rule 2. Where I occurs in J, if $I = J$, then $I = \Lambda$.

Rule 3. Where I and J occur in K, if $I = \Lambda$, then $J = \Lambda$ and $K = \Lambda$.

Definition 2 The *equality set belonging to* a λ-expression $=_{df}$, the union of the set of equalities which the expression yields, and the set of equalities derivable from that set.

Definition 3 The *expanded result of* a λ-expression $=_{df}$ J, where for the result I of the expression, $I = J$ is a member of the equality set belonging to the expression, and either J is null, or the set contains no equality $I = K$ where K is longer than J or is of the same length as J and has fewer distinct arguments.

If the expanded result of a λ-expression is null, the expression has no propositional correspondent. If this result is non-null, the desired correspondent is obtainable by relettering.

3 Illustrations In illustrating our procedure we make use of the following conventions:

A. 'Q' 'R' . . . are always used to denote the results of λ-prefixes, of $Y_{q-1}Y_q$, of Y_q, of $X_{m-1}X_m$, and of X_m.
B. 'Φ' 'Ψ' . . . are always used to denote the results of well-formed expressions commencing with a λ-prefix.

C. 'P_1' 'P_2' . . . are always used to denote the results of $Z_i Z_j$.
D. An asterisk marks the first derived equality.

In addition, for equalities yielded by an expression under Definition 1, the section of the definition in virtue of which the quality obtains is noted to its right.

Ex. 1 $\lambda abc\,.\,a(bc)$

$$\begin{aligned} \Phi &= CaCbCcQ && (1)\\ a &= CP_1 Q && (6b)\\ b &= CcP_1 && (3a)\\ {}^*\Phi &= CCP_1 QCCcP_1 CcQ \sim CCqrCCpqCpr && \end{aligned}$$

Ex. 2 $a(bc)$

$$\begin{aligned} \Phi &= Q && (1)\\ a &= CP_1 Q && (6b)\\ b &= CcP_1 && (3a)\\ {}^*\Phi &= Q \sim p && \end{aligned}$$

The expanded result of an expression with no bound variables will always be either elementary or null.

Ex. 3 $\lambda a\,.\,a(bb)$

$$\begin{aligned} \Phi &= CaQ && (1)\\ a &= CP_1 Q && (6b)\\ b &= CbP_1 && (3a)\\ {}^*b &= \Lambda && \\ \Phi &= \Lambda && \end{aligned}$$

The effect of derivation Rule 3 is that the occurrence of a single equality of the form $I = \Lambda$ in the equality set belonging to a λ-expression, ensures that the expanded result of the expression is null.

Ex. 4 $\lambda ab\,.\,a(ab)$

$$\begin{aligned} \Phi &= CaCbQ && (1)\\ a &= CP_1 Q && (6b)\\ a &= CbP_1 && (3a)\\ {}^*b &= P_1 && \\ Q &= P_1 && \\ a &= CP_1 P_1 && \\ \Phi &= CCP_1 P_1 CP_1 P_1 \sim CCppCpp && \end{aligned}$$

Ex. 5 $\lambda ab\,.\,a(b(\lambda c\,.\,a))$

$$\begin{aligned} \Phi &= CaCbQ && (1)\\ a &= CP_1 Q && (6b)\\ b &= C\Psi P_1 && (3b)\\ \Psi &= CcR && (2)\\ a &= R && (5)\\ {}^*R &= CP_1 Q && \end{aligned}$$

$$\Psi = CcCP_1Q$$
$$b = CCcCP_1QP_1$$
$$\Phi = CCP_1QCCCcCP_1QP_1Q \sim CCpqCCCrCpqpq$$

Ex. 6 $\lambda ab\,.\,a(\lambda cd\,.\,c(bd))$

$$\Phi = CaCbQ \quad (1)$$
$$a = C\Psi Q \quad (6b)$$
$$\Psi = CcCdR \quad (2)$$
$$c = CP_1R \quad (4b)$$
$$b = CdP_1 \quad (3a)$$
$$*\Psi = CCP_1RCdR$$
$$a = CCCP_1RCdRQ$$
$$\Phi = CCCCP_1RCdRQCCdP_1Q \sim CCCCqrCprsCCpqs$$

Ex. 7 $\lambda abc\,.\,b(abc)$

$$\Phi = CaCbCcQ \quad (1)$$
$$b = CP_1Q \quad (6b)$$
$$a = CbP_2 \quad (3a)$$
$$P_2 = CcP_1 \quad (3c)$$
$$*a = CCP_1QCcP_1$$
$$\Phi = CCCP_1QCcP_1CCP_1QCcQ \sim CCCpqCrpCCpqCrq$$

REFERENCES

[1] Curry, Haskell B., and Robert Feys, *Combinatory Logic*, Vol. I, North-Holland Publishing Co., Amsterdam (1958).

[2] Meredith, C. A., and A. N. Prior, "Notes on the axiomatics of the propositional calculus," *Notre Dame Journal of Formal Logic*, vol. IV (1963), pp. 171-181.

[3] Church, Alonzo, *The Calculi of Lambda-Conversion*, Princeton University Press (1941).

RCA Corporation
Dayton, New Jersey