Notre Dame Journal of Formal Logic
Volume XIV, Number 3, July 1973
NDJFAM

THE COMPLETENESS OF COMBINATORY LOGIC WITH DISCRIMINATORS

JOHN T. KEARNS

1.0 In [2] I introduced a system of combinatory logic with discriminators. Basically this is a system like those presented in [1], modified by the addition of discriminators, or discrimination functions. In this system, the reduction relation $>$ is somewhat different from the reduction relations considered in [1]. The relation $>$ is characterized by transitivity and left monotony—i.e.,

$$(\tau)\ X_1 > X_2,\ X_2 > X_3 \rightarrow X_1 > X_3$$
$$(\nu)\ X_1 > X_2 \rightarrow X_1 Y > X_2 Y.$$

In addition, there is a basic schema for $>$ corresponding to each basic combinator.

1.1 *Pure Combinators*. The pure combinators are the same as those studied in [1]; these are the combinators which do not involve discriminators. The basic pure combinators and their reduction schemata are:

$$\begin{array}{cc} IX > X & WXY > XYY \\ BXY_1Y_2 > X(Y_1Y_2) & SX_1X_2Y > X_1Y(X_2Y) \\ CXY_1Y_2 > XY_2Y_1 & \varphi XX_1X_2Y > X(X_1Y)(X_2Y) \\ KXY > X & \psi X_1X_2Y_1Y_2 > X_1(X_2Y_1)(X_2Y_2) \end{array}$$

Here, as in [1] and [2], parentheses associated to the left are omitted, so that $X_1X_2 \ldots X_n$ is an abbreviation for $(\ldots(X_1X_2)\ldots X_n)$.

It is unnecessary to adopt so many basic pure combinators. For S, K, and C provide a sufficient basis for constructing the rest, as shown below:

$$\begin{array}{ll} I \equiv SKS & \varphi \equiv BBBSB \\ B \equiv S(KS)K & \psi \equiv B\{B[BW(BC)]B\}(BB) \\ W \equiv S(CI) & \end{array}$$

1.2 *Some Definitions*. A *regular* combinator is one whose reduction leaves its first argument unchanged. All of the basic pure combinators are regular. It is sometimes desirable to employ combinators which leave

Received September 25, 1970

their first n arguments unchanged. These are *deferred* combinators; the number of arguments left unchanged is one greater than the numerical subscript enclosed in parentheses. Thus, for any regular combinator X, $X_{(n)}$ is a deferred combinator which operates like X. For instance, $C_{(n)}$ reduces as illustrated:

$$C_{(n)}X_0X_1 \ldots X_nY_1Y_2 > X_0X_1 \ldots X_nY_2Y_1.$$

For any regular combinator X, the deferred combinator is constructed

$$\begin{aligned} X_{(0)} &\equiv X \\ X_{(n+1)} &\equiv BX_{(n)}. \end{aligned}$$

There are two other combinators related to C which have numerical subscripts enclosed in square brackets. These combinators and their reductions are illustrated:

$$\begin{aligned} C_{[n]}XY_0 \ldots Y_n &> XY_nY_0 \ldots Y_{n-1} \\ C'_{[n]}XY_0 \ldots Y_n &> XY_1 \ldots Y_nY_0. \end{aligned}$$

Their constructions are given in [2].

For any combinators X, Y, the abbreviation '$X \cdot Y$' is defined by $X \cdot Y \equiv BXY$. This can be extended to $X_1 \cdot X_2 \cdot X_3 \equiv (X_1 \cdot X_2) \cdot X_3$, and so on. This gives rise to[1]

$$\begin{aligned} X^0 &\equiv I \\ X^1 &\equiv X \\ X^n &\equiv \underbrace{X \cdot X \cdot \ldots \cdot X}_{n} \end{aligned}$$

Combinatory numbers will be designated by ordinary numerals which are underlined. These numbers are constructed:

$$\begin{aligned} \underline{0} &\equiv KI \\ \underline{n+1} &\equiv SB\underline{n} \end{aligned}$$

1.3 *Discriminators*. The basic discriminator is Z. Its reduction is illustrated:[2]

$$\begin{aligned} ZXfgY &> fY, \text{ if } X \text{ is the same as } Y \\ &> gY, \text{ if } X \text{ is not the same as } Y \end{aligned}$$

With Z it is possible to construct Z^n which reduces

$$\begin{aligned} Z^nXfgY_0 \ldots Y_n &> fY_0 \ldots Y_n, \text{ if } X \text{ is the same as } Y_n \\ &> gY_0 \ldots Y_n, \text{ if } X \text{ is not the same as } Y_n \end{aligned}$$

1. The numerical superscripts do not have this significance with the discriminators Z, $Z_{()}$.

2. For X to be the same as Y, X and Y must either be the same simple symbol, or they must both be complex combinations constructed in the same manner from the same simple components.

(It is important to note that numerical superscripts attached to discriminators have a different significance than is given by the definition in 1.2.) The construction of Z^n is given in [2].

A second discriminator that will be taken as basic is $Z_{()}$. It reduces

$$Z_{()}fgX > \begin{cases} fX, & \text{if } X \text{ is a complex combination} \\ gX, & \text{if } X \text{ is a simple symbol} \end{cases}$$

A combinator $Z_{()}^n$ can be constructed, which reduces

$$Z_{()}^n fgX_0 \ldots X_n > \begin{cases} fX_0 \ldots X_n, & \text{if } X_n \text{ is complex} \\ gX_0 \ldots X_n, & \text{if } X_n \text{ is simple} \end{cases}$$

The combinator $Z_{()}$ need not be adopted as basic if the "supply" of basic combinators and simple non-combinatory symbols is fixed (and finite).

Another combinator to be taken as basic which is related to discriminators is R.[3] This reduces

$$RX(Y_1Y_2) > XY_1Y_2$$
$$RXY > XY, \text{ if } Y \text{ is simple}$$

The combinator R can be constructed under the same conditions as $Z_{()}$.

1.4 *Symbolic Systems.* A symbolic system consists of the basic combinators, a finite number of simple non-combinatory symbols, and *combinations* constructed from these by *application*. In the symbolic system(s) considered in this paper, the following three non-combinatory symbols will be employed: #, 1, 2. These symbols are dispensible, and could be replaced by complex combinations formed from basic combinators.

1.5 *The Combinators X''.* These combinators are related to the basic pure combinators, but they can operate on an arbitrary number of arguments. For these combinators, '#' serves as a termination symbol. The combinators, and their reductions are given:

$B''XY_1 \ldots Y_n\# > X(Y_1 \ldots Y_n\#)$, where none of $Y_1, \ldots, Y_n$ is '#'
$W''XY_1 \ldots Y_n\# > XY_1 \ldots Y_n\#Y_1 \ldots Y_n\#$, where none of $Y_1, \ldots, Y_n$ is '#'
$C''XY_1 \ldots Y_n\#Y_1' \ldots Y_m'\# > XY_1' \ldots Y_m'\#Y_1 \ldots Y_n\#$, where none of $Y_1, \ldots, Y_n, Y_1', \ldots, Y_m'$ is '#'
$K''XY_1 \ldots Y_n\# > X$, where none of $Y_1, \ldots, Y_n$ is '#'
$S''X_1X_2Y_1 \ldots Y_n\# > X_1Y_1 \ldots Y_n\#X_2Y_1 \ldots Y_n\#$, where none of $X_2, Y_1, \ldots Y_n$ is '#'
$\varphi''XX_1X_2Y_1 \ldots Y_n\# > XX_1Y_1 \ldots Y_n\#X_2Y_1 \ldots Y_n\#$, where none of $X_2, Y_1, \ldots, Y_n$ is '#'
$\psi''X_1X_2Y_1 \ldots Y_n\#Y_1' \ldots Y_m'\# > X_1X_2Y_1 \ldots Y_n\#X_2Y_1' \ldots Y_m'\#$, where none of $X_2, Y_1, \ldots, Y_n, Y_1', \ldots, Y_m'$ is '#'

3. For 'R' the numerical superscript has the sense given in 1.2. So that

$$R^n \equiv \underbrace{R \cdot R \cdot \ldots \cdot R}_{n}$$

$R''X(Y_1 \ldots Y_n \# Y_1' \ldots Y_m') > X(Y_1 \ldots Y_n) \# Y_1' \ldots Y_m'$, where none of $Y_1', \ldots, Y_m'$ is '#'

The construction of these combinators can be found in [2].[4]

2. *Completeness of the System Without Discriminators*. In (1), *combinatory completeness* is explained as follows. Let $\mathfrak{X}$ be a combination containing 0 or more occurrences of the simple, non-combinatory symbol x. Then, using the simple symbols occurring in $\mathfrak{X}$ (other than x) and the basic combinators, it is possible to construct a combinator X such that $Xx > \mathfrak{X}$. The combinator X is the x-abstract of $\mathfrak{X}$, and is sometimes written '$[x].\mathfrak{X}$.' In [1] it is shown that several different algorithms can be used to produce an x-abstract of a combination. In this section, I will show that the system of basic pure combinators (i.e., without discriminators) is combinatorily complete. The addition of discrimination functions does not alter this completeness; but they are not needed to provide completeness.

In [1] (pp. 190-1) there is a list of specifications given that provides the basis for different abstraction algorithms. This list can be adapted to the present system; the modified list is below. Each specification in the list is such that $[x].\mathfrak{X}$ has the form XI, and XYx reduces $XYx > Y\mathfrak{X}$ for all Y. When $[x].\mathfrak{X} \equiv XI$, '$[x]^{-I}.\mathfrak{X}$' will designate X. (In the list below it is understood that italic capitals do not contain occurrences of x.)

(a′) $[x].X \equiv C(BK)XI$
(b′) $[x].x \equiv II$
(c′) $[x].Xx \equiv CBXI$
(d′) $[x].X_1\mathfrak{X}_2 \equiv C(B^2X_2B)X_1I$, where $X_2 \equiv [x]^{-I}.\mathfrak{X}_2$
(e′) $[x].\mathfrak{X}_1X_2 \equiv C(BC(BX_1B))X_2I$, where $X_1 \equiv [x]^{-I}.\mathfrak{X}_1$
(f′) $[x].\mathfrak{X}_1\mathfrak{X}_2 \equiv B(B(BW(BX_2))X_1)BI$, where $X_1 \equiv [x]^{-I}.\mathfrak{X}_1$, $X_2 \equiv [x]^{-I}.\mathfrak{X}_2$

Lemma 1. *Let x be a simple symbol that is not a combinator. Let $\mathfrak{X}$ be a combination containing at most one occurrence of x. Then the algorithm* (a′), (d′), (e′), (b′)[5] *will produce a combination XI not containing x such that $XIx > \mathfrak{X}$ and $XYx > Y\mathfrak{X}$ for all Y.*

Proof. If $\mathfrak{X}$ does not contain x, then (a′) produces an appropriate XI. If $\mathfrak{X}$ contains an occurrence of x, the lemma is proved by induction on the structure of $\mathfrak{X}$.

The next lemma is a consequence of the specifications (a′), (d′), (e′), (b′) and the nature of the reduction relation.

4. The combinator R'' is not given in [2]. It can be constructed

$$R'' \equiv WI\{B^2R[Z^3\#(KI)(WI)]\}$$

5. When the specification (a') is used for a combination $\mathfrak{X}$ not containing x, it will be used just once for the whole combination, and not for the simple components of $\mathfrak{X}$.

Lemma 2. *Let x, y be distinct simple symbols which are not combinators. Let $\mathfrak{X}$ be a combination containing at most one occurrence of x, and let $\mathfrak{X}'$ be obtained from $\mathfrak{X}$ by replacing x by y. Let $[x].\mathfrak{X} \equiv XI$ be obtained by* (a′), (d′), (e′), (b′). *Then $XIy > \mathfrak{X}'$ and $XYy > Y\mathfrak{X}'$ for all Y. And if $\mathfrak{X}'$ contains an occurrence of y just where $\mathfrak{X}$ contains x (i.e., $\mathfrak{X}'$ contains no additional occurrences of y), then $[y].\mathfrak{X}' \equiv [x].\mathfrak{X} \equiv XI$.*

Theorem 1. *Let x be a simple symbol that is not a combinator. Let $\mathfrak{X}$ be a combination containing n occurrences ($n \geq 0$) of x. If $n \leq 1$, the algorithm* (a′), (d′), (e′), (b′) *will produce a combination XI not containing x such that $XIx > \mathfrak{X}$. If $1 < n$, the algorithm* (a′), (d′), (e′), (b′) *can be used on each occurrence of x to produce a combination XI not containing x such that $W^{n-1}(XI)x > \mathfrak{X}$.*

Proof. If $n \leq 1$, the theorem is established by Lemma 1. Suppose $n = 2$. Select a simple symbol y not occurring in $\mathfrak{X}$. Let $\mathfrak{X}'$ result from $\mathfrak{X}$ by replacing the first occurrence of x by y. By Lemma 1, the algorithm (a′), (d′), (e′), (b′) will produce a combination $X'I$ not containing y such that $X'Iy > \mathfrak{X}'$. By Lemma 2,

$$X'Ix > \mathfrak{X}.$$

By Lemma 1, the algorithm will produce a combinator X'' not containing x such that $X''Ix > X'I$. By (ν), $X''Ixx > X'Ix$. By (τ) and (i),

$$X''Ixx > \mathfrak{X}.$$

But $W^1(X''I)x > X''Ixx$. The general case is proved by induction on n.

There are other algorithms which can be used to perform abstraction. The algorithm (f′), (a′), (b′) can be used on a combination $\mathfrak{X}$ directly, avoiding the necessity of treating each occurrence of x separately. The more cumbersome procedure in Theorem 1 is presented in view of the result to be obtained in section **3**.

3.0 *Completeness with Discriminators.* The presence of discriminators does not change the completeness results of the preceding section, but these functions are so powerful that one would expect a stronger sort of completeness in a system containing them. It seems plausible to claim that any effective calculation can be represented (expressed?) in a system containing pure combinators and discriminators. But it is difficult to form a precise statement of a stronger completeness; a statement that would be susceptible of proof.

In [2] it is shown that Turing machines can be modelled by combinatory logic with discriminators. So every computable function is expressible with (pure) combinators, discriminators, and various other simple symbols. This does not require that non-numerical symbols be coded as numbers; combinatory logic with discriminators can incorporate any symbols.

In this section, I will show that the addition of discriminators to the pure combinators makes it possible to construct a combinator λ, so that for a simple symbol y that is not a combinator and a combination Y, $\lambda y Y > [y].Y$, where $[y].Y$ is formed as in Theorem 1.

3.1 *Some Definitions.* The combinators constructed in this section will be used in the construction of λ. The combinator $\mathfrak{D}_i$ reduces

$$\mathfrak{D}_i XY_1 \ldots Y_i x > XxY_1 \ldots Y_i x$$

It is constructed

$$\mathfrak{D}_i \equiv B^{i+1} WC_{[i]}$$

The reduction and construction of $\mathfrak{F}_i$ are given

$$\mathfrak{F}_i X_0 \ldots X_i fgY_1 \ldots Y_m\#(\#\#) > X_0 \ldots X_i(fY_1 \ldots Y_m\#)(gY_1 \ldots Y_m\#)(\#\#),$$
where none of Y_i, Y_{i+1} are #, (##)
$$\mathfrak{F}_i \equiv WI[Z^{i+4}\#\{\varphi_{(i+2)}[Z^{i+4}(\#\#)(KI)(WI)]\}\{\varphi_{(i+2)}(WI)\}]$$

$\mathfrak{L}$ is given

$$\mathfrak{L}XY_1^0 \ldots Y_{n_0}^0\#Y_1^1 \ldots Y_{n_1}^1\# \ldots \#Y_1^m \ldots Y_{nm}^m\#(\#\#) > X\underline{m}[Y_1^0 \ldots . Y_{nm}^m\#](\#\#),$$
where none of $Y_j^i \equiv \#$, and no $Y_1^i \equiv (\#\#)$
$$\mathfrak{L} \equiv WI\{B^3B''[Z^4(\#\#)[K(CI)]\{C[BB(WI)](SB)\}]\}\ \underline{0}$$

And finally $\mathfrak{T}_i$ is given

$$\mathfrak{T}_i X_0 \ldots X_i Y_1 \ldots Y_m\#(\#\#) > X_0 \ldots X_i(Y_1 \ldots Y_m\#)(\#\#),$$ where none of X_j, X_{j+1} are #, (##)
$$\mathfrak{T}_i \equiv WI[B''_{(i+5)}Z^{i+3}(\#\#)(KI)(WI)]$$

3.2 *The Combinator* λ.

Theorem 2. *Let y be a simple symbol that is not a combinator. Let Y be a combination (containing 0 or more occurrences of y). Then there is a combinator λ such that $\lambda yY > [y] . Y$, where $[y] . Y$ is as described in Theorem 1, except that W^{n-1} is replaced by $(\underline{n-1})W$.*

The proof of this theorem is the construction of λ. I will not show here that the construction given is adequate, for showing this is a straightforward but quite lengthy task.

$$\lambda \equiv C_{[3]}\{B^2_{(6)}B^3_{(4)}B_{(2)}BC\Delta(WI)B^2\Delta'\underline{1}\}\Delta''(WI)\#$$

λyY reduces

$$\lambda yY > \Delta\{WI[B^2(\Delta'\underline{1}y)\Delta''(WI)]\}Y\#.$$

If $Y \equiv y$, $\Delta XY\# > II$. If Y is a simple symbol distinct from y, $\Delta XY\# > C(BK)YI$. If Y is complex,

$$\Delta XY\# > X\#x_1^1 \ldots x_{n_1}^1\# \ldots \#x_1^m \ldots x_{nm}^m\#(\#\#)S_1 \ldots S_m\#(\#\#)\#$$

Here $S_1, \ldots, S_m$ are the simple symbols occurring in Y, in the order of their occurrence. The position of S_i in Y is given the n-tuples of '1's and '2's, $x_1^i \ldots x_{ni}^i$. The construction of Δ is given

$$\Delta \equiv Z^1_{()}\Gamma_1\Gamma_2$$
$$\Gamma_1 \equiv C_{[8]}\{B[R_{(8)}(WI\Gamma_3)]\}\#1\#2\#(\#\#)\#(\#\#)$$
$$\Gamma_2 \equiv K_{(6)}R_{(3)}K_{(2)}R^2_{(1)}K^2_{(1)}\{R_{(3)}K_{(2)}R_{(1)}K[C_{[2]}Z[K(II)]\{C[C(BK)]I\}]\}$$

$\Gamma_3 \equiv \mathfrak{F}_1\mathfrak{F}_0 Z^1_{()}\Gamma_4\Gamma_5(WI)$
$\Gamma_4 \equiv B^3W''(\Gamma_6\Gamma_7)$
$\Gamma_5 \equiv B''_{(2)}[Z^3(\#\#)\Gamma_8\Gamma_9]$
$\Gamma_6 \equiv B''\{R_{(3)}C_{(1)}B''[B''_{(1)}R(CI2)]\,1\}$
$\Gamma_7 \equiv \mathfrak{T}_1\mathfrak{T}_0 R_{(1)}$
$\Gamma_8 \equiv K^2_{(1)}\mathfrak{T}_0(RC_{[2]})$
$\Gamma_9 \equiv \mathfrak{T}_2\mathfrak{T}_1 RC_{[2]}$

When Y is complex

$\Delta\{WI[B^2(\Delta'\,\underline{1}y)\Delta''(WI)]\}Y\# >$
$\quad \Delta'\,\underline{1}y\,[\Delta''(WI)\{WI[B^2(\Delta'\,\underline{1}y)\Delta''(WI)]\}]\#x^1_1 \,.\,.\,.\,.\, x^m_{nm}\#(\#\#)S_1 \,.\,.\,.\, S_m\#(\#\#)\#$

And

$\Delta'\,\underline{1}yX\#x^1_1 \,.\,.\,.\,.\, x^m_{nm}\#(\#\#)S_1 \,.\,.\,.\, S_m\#(\#\#)\# >$
$\quad X\#x^1_1 \,.\,.\,.\,.\, x^m_{nm}\#(\#\#)\underline{i}S_1 \,.\,.\,.\, S_m\#(\#\#)1\#$, where $S_i \equiv y$ and $S_j \neq y$ for all $j < i$
$\quad X\#x^1_1 \,.\,.\,.\,.\, x^m_{nm}\#(\#\#)S_1 \,.\,.\,.\, S_m\#(\#\#)\#$, where $S_i \neq y$ for $1 \leq i \leq m$

Δ' is constructed

$\Delta' \equiv \mathfrak{T}_2(WI\Sigma_1)$
$\Sigma_1 \equiv \mathfrak{D}_7 I\varphi(Z^6\#)\Sigma_2 I\Sigma_3(WI)$
$\Sigma_2 \equiv \mathfrak{D}_7 I\varphi[Z^7(\#\#)](K^5I)I$
$\Sigma_3 \equiv \mathfrak{D}_5[B(\mathfrak{D}_6 I\varphi)Z^5]\Sigma_4\Sigma_5$
$\Sigma_4 \equiv K^2_{(1)}C'_{[3]}(K_{(3)}\mathfrak{T}_2C'_{[2]}I1)$
$\Sigma_5 \equiv C'_{[2]}B_{(1)}(SB)$

When the reduction of Δ' is completed, the resulting combination begins

$$\Delta''(WI)\,\{B^2(\Delta'\,\underline{1}y)\Delta''(WI)\}$$

This reduces

$\Delta''(WI)\,\{B^2(\Delta'\,\underline{1}y)\Delta''(WI)\}\,\#x^1_1 \,.\,.\,.\,.\, x^m_{nm}\#(\#\#)\underline{i}S_1 \,.\,.\,.\, S_m\#(\#\#)\underbrace{1\,.\,.\,.\,1}_{r}\# >$

$\quad \Delta\{WI[B^2(\Delta'\,\underline{1}y)\Delta''(WI)]\}\,Y'1 \,.\,.\,.\, 1\#$, where Y' is the y-abstract for the first occurrence of y in the formula analyzed by Δ in the preceding stage.

$\Delta''(WI)\{B^2(\Delta'\,\underline{1}y)\Delta''(WI)\}\#x^1_1 \,.\,.\,.\,.\, x^m_{nm}\#(\#\#)S_1 \,.\,.\,.\, S_m\#(\#\#)\underbrace{1\,.\,.\,.\,1}_{r}\# >$

$\quad C(BK)YI$, if $r = 0$; $[y]\,.\,Y$, as formed by algorithm (a′), (d′), (e′), (b′) if $r = 1$; and $(r - \mathit{1})\,W(Y'I)$, if $r > 1$, where $(Y'I)$ is produced as described in Theorem 1.

Δ'' is constructed

$\Delta'' \equiv \mathfrak{L}(Z^3_{()}\Pi_1\Pi_2)(B\Delta)$
$\Pi_1 \equiv C_{[3]}(C_{[3]}R''_{(3)}R''_{(1)}\Pi_3)$
$\Pi_2 \equiv C_{[2]}R''_{(2)}R''\,[B^2(WI\Pi_{24}\underline{0})\Pi_{25}]$
$\Pi_3 \equiv \mathfrak{D}_3\mathfrak{T}_2B(C'_{[2]}KI)(WI\Pi_4)$
$\Pi_4 \equiv \mathfrak{D}_4B_{(5)}\,B''_{(3)}\,R^2_{(2)}\,[Z^32(WI\Pi_5)(WI\Pi_6)]$

$$
\begin{aligned}
\Pi_5 &\equiv R''_{(7)} R_{(6)} Z^4 1 \Pi_{16}(WI\Pi_{17}\underline{0})\\
\Pi_6 &\equiv B^2_{(5)} B''_{(4)} R^2_{(3)} [Z^4 2 \Pi_7(WI\Pi_8\underline{0})]\\
\Pi_7 &\equiv K_{(3)} K_{(5)} R''^2_{(3)} K''_{(4)} [\mathfrak{D}_4(B^6 \mathfrak{L} B^4_{(1)}) B \Pi_9 \Pi_{10}]\\
\Pi_8 &\equiv B^2_{(10)} C_{(3)} B_{(3)} B''_{(6)} R^2_{(5)} [Z^6 2 \Pi_{13}(WI)](SB)\\
\Pi_9 &\equiv C'_{[8]} \{C'_{[6]} [B^2_{(7)} C'_{[5]}(B_{(7)} C'_{[5]} B^2_{(4)} C)]\} BB(BC)\\
\Pi_{10} &\equiv WI \{R_{(7)} Z^5(\#\#)\Pi_{11}(WI)\}\\
\Pi_{11} &\equiv C_{[5]} K_{(2)} R''_{(2)}(\mathfrak{D}_3 IR''_{(2)} \Pi_{12})\\
\Pi_{12} &\equiv Z^4\# \{C'_{[8]} B_{(6)} I [K^2(K^3_{(1)} K^2_{(1)})]\}(WI)\\
\Pi_{13} &\equiv K \{K_{(9)} R''^2_{(7)} K''_{(8)} B^4 \mathfrak{L} [\mathfrak{D}_6 IB_{(6)}(\mathfrak{D}_3 B^6_{(1)} BB_{(6)})\Pi_{14}]\}\\
\Pi_{14} &\equiv \mathfrak{D}_2 IR_{(5)}(\mathfrak{D}_5 IR_{(5)} \Pi_{15})\\
\Pi_{15} &\equiv C_{[6]} IR''_{(4)} [C_{[3]} IR''_{(4)} R_{(3)}(WI)]\\
\Pi_{16} &\equiv K_{(6)} K_{(8)} K''_{(8)} R_{(4)} K_{(3)} B^2 \mathfrak{L} [\mathfrak{D}_3 B^3_{(1)} B \Pi_{18}]\\
\Pi_{17} &\equiv C_{(3)} B_{(3)} R''_{(6)} R_{(5)} [Z^6 1 \Pi_{21}(WI)](SB)\\
\Pi_{18} &\equiv C_{(4)} \{C'_{[6]} [C'_{[4]} (B^2_{(7)} C'_{[5]} B^2_{(4)} C \Pi_{19})] BB^2\}\\
\Pi_{19} &\equiv WI \{R_{(7)} Z^5(\#\#)\Pi_{20}(WI)\}\\
\Pi_{20} &\equiv C_{[5]} K_{(2)} R''_{(2)}[\mathfrak{D}_3 R''_{(4)} R''_{(2)} \Pi_{12}]\\
\Pi_{21} &\equiv K \{K_{(10)} K''_{(10)} R_{(6)} K_{(5)} B^4 \mathfrak{L} [\mathfrak{D}_6 IB_{(6)}(\mathfrak{D}_3 IR_{(6)} B_{(6)})\Pi_{22}]\}\\
\Pi_{22} &\equiv \mathfrak{D}_2 IB_{(5)} (\mathfrak{D}_5 IR_{(5)} \Pi_{23})\\
\Pi_{23} &\equiv C_{[6]} IR''_{(4)} [C_{[3]} IB''_{(4)} R^2_{(3)}(WI)]\\
\Pi_{24} &\equiv C(B_{(3)} B''_{(3)} R^2_{(2)} \{Z^3 2 \Pi_{26} [B^2_{(2)} (WI)]\})(SB)\\
\Pi_{25} &\equiv K^4 \{Z^3\# [K^3_{(1)} \{C [C(BK)] I\}] [Z^4\# K^4(WI\Pi_{28}\underline{0})]\}\\
\Pi_{26} &\equiv R''_{(5)} R_{(4)} K_{(4)} K''_{(4)} (\mathfrak{D}_5 \mathfrak{T}_3 BBI \Pi_{27})\\
\Pi_{27} &\equiv B_{(5)} C_{[3]} B_{(1)} R'' \{Z^4(\#\#)(KI) [B_{(3)} C(WI)\underline{0}]\}\\
\Pi_{28} &\equiv C_{(2)} B_{(2)} B_{(5)} [Z^6\#(K^4_{(5)} C_{[2]} K_{(1)} W)(WI)](SB)
\end{aligned}
$$

REFERENCES

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

[2] Kearns, John T., "Combinatory logic with discriminators," *The Journal of Symbolic Logic*, vol. 34 (1969), pp. 561-575.

State University of New York at Buffalo
Buffalo, New York