Molecular Combinator Reference Manual

UPIM Report 2
Technical Report UT-CS-02-489

Bruce J. MacLennan*

Department of Computer Science
University of Tennessee, Knoxville
www.cs.utk.edu/~mclennan

November 5, 2002

Abstract

This report contains, in summary form, definitions, schematic reactions, and equivalences of all combinators in use by this project. It will be updated as new combinators, equivalences, etc. are used.

*This research is supported by Nanoscale Exploratory Research grant CCR-0210094 from the National Science Foundation. It has been facilitated by a grant from the University of Tennessee, Knoxville, Center for Information Technology Research. This report may be used for any non-profit purpose provided that the source is credited.
Introduction

1. Most of the combinator definitions and equivalences (beyond those peculiar to molecular computation, such as \(R, D, \) and \(V\)) are from Curry and Feys [CFC58].

2. We follow the usual convention in combinatory logic of omitting parentheses that associate to the left. For example, \(XYZ\) means \(((XY)Z)\), and \(B(W(B(C)))B(B(B))\) means \(((B((B(W))(B(C))))((B(B))(B(B))))\).

3. In the definitions of the operators, variables are marked with primes (e.g., \(X'\)) and parenthesized superscripts (e.g., \(X^{(\ell)}\)) to indicate shared complexes. See the description of the \(V\) (Sharing) Primitive (Section 17).

4. Notice that the following are distinct and have different meanings: \(X^n\) (powers of combinators), \(X_n\) (polyadic combinators), \(X^{(n)}\) (sharing), \(X_{(n)}\) (deferred combinators), \(X^{[n]}\) (left reduction), \(X_{[n]}\) (polyadic extension); see Other Notation (p. 14). \(X_n\) is also used in the usual way to denote an element in a series \(X_1, X_2, X_3, \ldots\). When subscripts and superscripts of any kind are combined, the subscripts take precedence; thus \(\Phi_n\) means \((\Phi_n)^m\).

5. The size \(|X|\) of a nonprimitive combinator \(X\) is expressed in terms of the number of \(S, K, \) and \(A\) nodes that it contains. Since nonprimitive combinator definitions are binary trees, if they contain no other nodes besides \(S, K, \) and \(A\), then the counts satisfy \(A = S + K - 1\), and the total nodes are \(T = 2A + 1 = 2(S + K) - 1\).

6. A combinator is called regular if it does not affect its first argument, thus,

\[ FXY_1\cdots Y_n \rightarrow XZ_1\cdots Z_m. \]

Most combinators (e.g., \(B, B', C, I, K, S, W, Y, \Phi, \Phi_n, \Psi\)) are regular.
Definitions of Combinators

1 A Primitive (Application Complex)

The application (A) complex represents the application of a combinator to its argument. The application of $F$ to $X$, written $FX$, is represented by a molecular complex $UAFX$, in which the “operator” binding site of $A$ is linked to $F$, the “operand” binding site is linked to $X$, and the “result” site is linked to $U$, the complex into which the result of $FX$ will be linked.

All (or most) of the non-terminal (interior) nodes of a combinator tree are A nodes; the terminals (leaves) are primitive combinators (e.g., $S$ and $K$). If the network is not a tree, but has shared nodes or cycles, then (most of) the non-terminal nodes are A and V (sharing) nodes. (We say “most” because later we may want to define additional interior node types.)

2 B Combinator (Elementary Compositor)

Definition:

\[ \text{BXYZ} \implies X(YZ) \]  

Reduction to SK:

\[ \text{B} = \text{S}(\text{KS})\text{K} \]  

Size: $2S + 2K + 3A = 7$ total.

Equivalences:

\[ \text{B} = \text{CB}' \]  

\[ \text{B} = \text{C}(\text{JI})\text{C}(\text{JI}) \]  

\[ \text{B}'nFX_1\cdots X_n \implies F(GX_1\cdots X_n) \]  

Notes: If $F$ is regular (p. 2), $FXY_1\cdots Y_n \implies XZ_1\cdots Z_m$, then

\[ \text{B}'nFGXY_1\cdots Y_n \implies GXZ_1\cdots Z_m. \]

That is, $G$ is applied to the result of applying $F$ to the arguments $XY_1\cdots Y_n$.

3 B’ Combinator (Permuting Compositor)

Definition:

\[ \text{B}'XYZ \implies X(ZY') \]  

Equivalences:

\[ \text{B}' = \text{CB} \]  

Size: $7S + 6K + 12A = 25$ total.
4 C Combinator (Elementary Permutator)

Definition:

\[ CXYZ \implies XZY \] (8)

Equivalences:

\[
\begin{align*}
C &= B(BS) \\
C &= S(BBS)(KK) \\
C &= JC_\ast(JC_\ast)(JC_\ast)
\end{align*}
\] (9) (10) (11)

Size: \(5S + 4K + 8A = 17\) total (Def. 9).

5 C* Combinator (Pure Permutator)

Definition:

\[ C_\ast XY \implies YX \] (12)

Equivalences:

\[
\begin{align*}
C_\ast &= CI \\
C_\ast &= JIII
\end{align*}
\] (13) (14)

Size: \(6S + 6K + 11A = 23\) total (Def. 13).

6 D Primitive (Elementary Deleter)

Reaction:

\[
\begin{align*}
D_P + PQ &\longrightarrow P_P + DQ \\
DAXY + DQ + PQ &\longrightarrow DX + DY + PAQ_2 \\
DURX + 2PQ &\longrightarrow UX + P_2RQ + DQ \\
DUVX + PQ &\longrightarrow PUVX + DQ
\end{align*}
\] (15) (16) (17) (18)

Notes: In Eq. 15, \(p\) represents any primitive combinator (e.g., S or K). Notice that in Eq. 17, a deletion cancels a replication in progress. However, in Eq. 18, a deletion does not affect a shared complex, except to cap the deleted sharing link.

Reaction Specification:

\[
\begin{align*}
d &: D, a &: A, x, y, d &: D, p &: P, q &: Q, q &: Q.
d, a_1, x, a_2, y, d', q', p, q
=> (\text{DeleteApplication})
\end{align*}
\] (DeleteApplication)

\[
\begin{align*}
d &x, d &y, p &a, a_1 &q, a_2 &q'.
\end{align*}
\]
Notes: In the last (DeletePrimitive) rule, ‘Prim’ stands for any primitive combinator. Therefore, at least at the present time, that rule must be repeated with ‘Prim’ replaced by each primitive combinator species in use (e.g., ‘K’, ‘S’).

7 | Combinator (Elementary Identifier)

Definition:
\[ \text{l}X \Rightarrow X \]  
\[ (19) \]

Reduction to SK:
\[ \text{l} = \text{SK}X \]  
\[ (20) \]

Size: 1S + 2K + 2A = 5 total (taking \( \text{l} = \text{SKK} \)).

Equivalences:
\[ \text{l} = \text{CK}X \]  
\[ (21) \]
\[ \text{l} = \text{WK} \]  
\[ (22) \]
8 J Combinator

Definition:

\[ JUXYZ \rightarrow UX(UYZ) \]  
(23)

9 K Combinator (Elementary Cancellator)

Definition:

\[ KXY \rightarrow X \]  
(24)

Reaction:

\[ UA_2KXY + DQ \rightarrow UX + DA_2QY \]  
(25)

Equivalences:

\[ K^nXY_1 \cdots Y_n \rightarrow X \]  
(26)

Reaction Specification:

\[ a: A, b: A, k: K, d: D, q: Q, u, x, y. \]
\[ u a, a_1 b, b_1 k, b_2 x, a_2 y, d q \rightarrow \text{(Kreaction)} \]
\[ u x, d a, a_1 b, b_1 k, b_2 q, a_2 y. \]

10 N Combinator (Inert Complex)

The N (inert) combinator is used when we want to prevent reduction, generally when we are intending to produce a static structure. For example, if the structure \( FX_1 \cdots X_n \) is generated, then there is a risk that the reduction rules for \( F \) will destroy the structure. This is avoided by using the inert combinator, e.g. \( NX_1 \cdots X_n \). Since it is inert, there are no reduction or reaction rules for it. Of course, in practice, there need not be just one inert combinator, and any molecular species that does enter into the computational reactions could be used.

11 P Primitive (Result Cap)

The result cap is inert; it is a place-holder for the “result” binding-site of any group.

12 Q Primitive (Argument Cap)

The argument cap is inert; it is a place-holder for the “argument” binding-site of any group (in particular, for the “operator” and “operand” sites of an A complex).
13 **R Primitive (Elementary Replicator)**

Reaction:

\[
UV R_p + P_p + P Q \rightarrow U p + V p + P_2 R Q
\]  
\[
UV R A X Y + PA Q_2 + P_2 R Q \rightarrow (U A)(V A)(R X)(R Y) + 3 P Q
\]

**Notes:** In Eq. 27, \( p \) represents any primitive combinator (e.g., \( S \) or \( K \)).

**Reaction Specification:**

\[
\begin{align*}
  r: & \ R, \ a: \ A, \ u, \ v, \ x, \ y, \ r': \ R, \ a': \ A, \\
  p: & \ P, \ p': \ P, \ p'': \ P, \ q: \ Q, \ q': \ Q, \ q'': \ Q, \\
  u \ r_1, & \ v \ r_2, \ r \ a, \ a_1 \ x, \ a_2 \ y, \\
  p \ r'_1, & \ p' \ r''_2, \ r' \ q, \ p'' \ a', \ a'_1 \ q', \ a'_2 \ q'' \\
\end{align*}
\]

\( \Rightarrow \) (ReplicateApplication)

\[
\begin{align*}
  u \ a, & \ v \ a', \\
  a_1 \ r_1, & \ a'_1 \ r_2, \\
  a_2 \ r'_1, & \ a'_2 \ r''_2, \\
  r \ x, & \ r' \ y, \\
  p \ q, & \ p' \ q', \ p'' \ q''.
\end{align*}
\]

\[
\begin{align*}
  r: & \ R, \ pc: \ Prim, \ u, \ v, \ pc': \ Prim, \ p: \ P, \ p': \ P, \ q: \ Q, \\
  u \ r_1, & \ v \ r_2, \ r \ pc, \ p \ q, \ p' \ pc' \\
\end{align*}
\]

\( \Rightarrow \) (ReplicatePrimitive)

\[
\begin{align*}
  u \ pc, & \ v \ pc', \\
  p \ r_1, & \ p' \ r_2, \ r \ q.
\end{align*}
\]

**Notes:** In the last (ReplicatePrimitive) rule, ‘Prim’ stands for any primitive combinator. Therefore, at least at the present time, that rule must be repeated with ‘Prim’ replaced by each primitive combinator species in use (e.g., ‘\( K \)’, ‘\( S \)’).

14 **S Combinator (Elementary Formalizer, Replicating)**

**Definition:**

\[
S X Y Z \rightarrow X Z(Y Z)
\]

**Reaction:**

\[
U A_3 S X Y Z + P_2 R Q \rightarrow U A(A X)(A Y) R Z + P S + P Q
\]

**Reaction Specification:**

\[
\begin{align*}
  a: & \ A, \ a': \ A, \ a'': \ A, \ s: \ S, \ r: \ R, \ p: \ P, \ p': \ P, \ q: \ Q, \\
  u, & \ x, \ y, \ z.
\end{align*}
\]
\[
\begin{array}{l}
u, a, a_1 a', a'_1 a'', a''_1 s, a''_2 x, a'_2 y, a_2 z, \\
p r_1, p' r_2, r q
\end{array}
\]

\[\Rightarrow \text{(Sreaction)}\]

\[
\begin{array}{l}
u, a, a_1 a', a'_1 x, a'_2 r_1, \\
a_2 a'', a''_1 y, a''_2 r_2, \\
r z, \\
p s, p' q.
\end{array}
\]

Equivalences:

\[
S = B(B(W)C)(BB) \quad (31)
\]

15 \(\check{S}\) Combinator (Elementary Formalizer, Sharing)

Definition:

\[
\check{S}XYZ \implies XZ'(YZ) \quad (32)
\]

Reaction:

\[
UA_3\check{S}XYZ + P_2VQ \rightarrow UA(AX)(AY)VZ + PS + PQ \quad (33)
\]

Reaction Specification:

\[
\begin{array}{l}
u, x, y, z.
\end{array}
\]

\[
\begin{array}{l}
u, a, a_1 a', a'_1 a'', a''_1 s, a''_2 x, a'_2 y, a_2 z, \\
p v_1, p' v_2, v q
\end{array}
\]

\[\Rightarrow \text{(SharingSreaction)}\]

\[
\begin{array}{l}
u, a, a_1 a', a'_1 x, a'_2 v_1, \\
a_2 a'', a''_1 y, a''_2 v_2, \\
v z, \\
p s, p' q.
\end{array}
\]

Equivalences:

\[
\check{S} = B(B(W)C)(BB) \quad (34)
\]

Notes: See Sec. 19 for a discussion of this definition.

16 \(S_n\) Combinator (Polyadic Elementary Formalizer)

Definition:

\[
S_n XY_1 \cdots Y_n X \implies XZ(Y_1 Z) \cdots (Y_n Z) \quad (35)
\]
Reduction to SK:

\[
S_1 = S \quad (36)
\]

\[
S_{n+1} = BS_n \circ S \quad (37)
\]

**Size:** \((5n - 4)S + 4(n - 1)K + 9(n - 1)A = 18(n - 1) + 1\) total for \(S_n\).

**Notes:** \(S_n\) can be replicating or sharing depending on whether \(S\) or \(\tilde{S}\) is used in its recursive definition. If it is sharing, it produces the following structure:

\[
\tilde{S}_nXY_1 \cdots Y_nZ \rightarrow XZ^{(n)}(Y_1Z^{(n-1)}) \cdots (Y_{n-1}Z')(Y_nZ) \quad (38)
\]

**Equivalences:**

\[
S_n = \Phi_{n+1}I \quad (39)
\]

17  \(\forall\) **Primitive (Sharing Complex)**

The sharing primitive (\(\forall\)) is used for constructing non-tree structures, including cyclic structures. It is produced by sharing combinators such as \(\tilde{S}\), \(\bar{W}\), and \(\bar{Y}\). Note that a \(\forall\) complex between a combinator and its arguments will block reduction of the combinator, so \(\forall\) complexes appear primarily in structures that are being treated as data.

Primes and parenthesized superscripts on variables are used to indicate informally the sharing of structures. Thus, if there is a single sharing complex above \(X\), then the two links to it will be called \(X\) and \(X'\). Notice that both will be "covered" by a sharing complex; if it is necessary to make this explicit, the two links will be written \(X^{(0)}\) and \(X'\). If one of these links is replaced by another sharing complex, then the original link and the two new ones will be called \(X, X', X''\), and so forth. Obviously such a notation cannot capture all the possible structures of sharing complexes, but it allows the convenient expression of chains of \(\forall\) complexes, which is the most common case. To go beyond this, diagrams should be used.

18  \(\forall\) **Combinator (Elementary Duplicator, Replicating)**

**Definition:**

\[
WXY \rightarrow XYY \quad (40)
\]

**Equivalences:**

\[
W = CSI \quad (41)
\]

\[
W = S(CI) \quad (42)
\]

\[
W = SS(KI) \quad (43)
\]

**Size:** \(7S + 6K + 12A = 25\) total (Def. 41 or 42).
19 \( \tilde{W} \) Combinator (Elementary Duplicator, Sharing)

**Definition:**

\[
\tilde{W}XY \implies XY'Y
\]  
(44)

**Reduction to SK:**

\[
\begin{align*}
\tilde{W}_{12} &= \text{CSI} \quad (45) \\
\tilde{W}_{21} &= \tilde{S}(CI) \quad (46) \\
\end{align*}
\]

**Notes:** \( \tilde{W}_{12} \) and \( \tilde{W}_{21} \) are two variants, functionally equivalent to \( \tilde{W} \), but producing differently ordered links to the sharing \( (V) \) complex (see Equivalences below). In the absence of subscripts, we will take \( \tilde{W} \) to be \( \tilde{W}_{12} \), since it is a little more convenient to use. Definition 47 is not very useful, because it needlessly begins replication of the first argument of \( \tilde{W}_{12} \).

Notice that either \( \tilde{W} \) or \( \tilde{S} \) may be taken as a primitive sharing operation, since either can be defined in terms of the other. At this time, it looks as though \( \tilde{S} \) will be the best choice as a primitive, so \( \tilde{W} \) will be defined by Eq. 45 or 46.

**Reaction:**

\[
UA_2\tilde{W}XY + P_2VQ \longrightarrow UA_2XYY + P\tilde{W} + PQ
\]  
(48)

**Reaction Specification:**

\[
\begin{align*}
u a, a_1 a', a'_1 w, a'_2 x, a_2 y, p v_1, p' v_2, v q \\
\implies (\text{SharingWreaction}) \\
u a, a_1 a', a'_1 x, a'_2 v_1, a_2 v_2, v y, p' w, p q.
\end{align*}
\]

**Equivalences:**

\[
\begin{align*}
\tilde{W}_{12}XY & \implies XY'Y \quad (49) \\
\tilde{W}_{21}XY & \implies XYY' \quad (50) \\
\tilde{W}_{12}^mXY & \implies XY^{(n)} \ldots Y^{(n+1)}Y'Y \\
\end{align*}
\]

**Notes:** The primes and superscripts on \( Y \) in Eq. 51 represent successive sharings of \( Y \) (see Sec. 17).

20 \( W_* \) Combinator (Pure Duplicator)

**Definition:**

\[
W_*X \implies XX
\]  
(52)
21 **Y Combinator (Elementary Fixed-point, Replicating)**

**Definition:**

\[ YF \implies X(YX) \]  

(54)

**Reduction to SK:**

\[ Y = SSK(S(K(SS(S(SK))))))K) \]  

(55)

**Size:** \(8S + 4K + 11A = 23\) total.

**Equivalences:**

\[
\begin{align*}
Y &= W(S(BWB)) \\
Y &= SSIB(K(SSI))) \\
Y &= ZZ \text{ where } Z = W(B(II)) \\
Y &= WI \circ W \circ B
\end{align*}
\]

(56) 

(57) 

(58) 

(59)

**Notes:** Definition 55 by John Tromp [LV97] may be the shortest definition in terms of SK (12 combinators). Definitions by Curry and Turing are longer (18 and 20, respectively).

22 **\~Y Combinator (Elementary Fixed-point, Sharing)**

**Definition:**

\[ \~YX \implies y \quad \text{where } y = Fy' \]  

(60)

**Reaction:**

\[ U\~A\~YX + P_2VQ \implies UVA.X + P\~Y + PQ \]  

(61)

**Reaction Specification:**

\[
\begin{array}{l}
\end{array}
\]

\[
\begin{array}{l}
u a, a_1 y, a_2 x, p v_1, p' v_2, v q
\end{array}
\]

\[
\begin{array}{l}
\Rightarrow \text{(SharingYreaction)}
\end{array}
\]

\[
\begin{array}{l}
u v_1, v a, a_1 x, a_2 v_2, p' y, p q.
\end{array}
\]
**Notes:** The following illustrates the self-sharing cycle created by \( \check{Y}F \):

\[
\begin{align*}
\check{Y}F &= y \\
&= Fy' \\
&= F(Fy')' \\
&= F(F'y'') \\
&= F(F'(Fy')''') \\
&= F(F''(F''y''')) \\
&= F(F''(F''(F''(F''(F''(\ldots)))))) \\
&= \ldots \\
&= F(F''(F''(F''(F''(F''(\ldots)))))) 
\end{align*}
\]

Of course, it is the A complex that is shared, not \( F \), as the notation suggests.

### 23 Z Combinators (Iterators or Church Numerals)

**Definition:**

\[
\begin{align*}
Z_0 &= KL \quad (62) \\
Z_{n+1} &= SBJ_n \quad (63)
\end{align*}
\]

**Size:** \((3n + 1)S + (2n + 3)K + (5n + 3)A = 10n + 7\) total for \( Z_n \).

**Equivalences:**

\[
\begin{align*}
Z_nX &= X^n \\
Z_{m+n} &= \Phi BZ_mZ_n \quad (64) \\
Z_{m+n} &= Z_m \circ Z_n \quad (65) \\
Z_{n+m} &= Z_mZ_n \quad (66)
\end{align*}
\]

### 24 Φ Combinator (Dyadic Compositor)

**Definition:**

\[
\begin{align*}
\Phi X Y Z U \quad \Rightarrow \quad X(YU)(ZU) 
\end{align*}
\]

**Equivalences:**

\[
\begin{align*}
\Phi &= B(BS)B \\
\Phi^n F G H X_1 \cdots X_n \quad \Rightarrow \quad F(GX_1 \cdots X_n)(HX_1 \cdots X_n) 
\end{align*}
\]

*Size:* \(7S + 6K + 12A = 25\) total (Def. 69).
25 \( \Phi_n \) Combinator (Polyadic Compositor)

**Definition:**
\[
\Phi_nX_{Y_1}\cdots Y_nZ \rightarrow X(Y_1Z)\cdots(Y_nZ)
\]  \( (71) \)

**Reduction to SK:**
\[
\Phi_n = S_n \circ K
\]  \( (72) \)

**Size:** \((5n - 2)S + (4n - 1)K + (9n - 4)A = 18n - 7\) total for \( \Phi_n \).

**Notes:** \( \Phi_n \) can be replicating or sharing (\( \bar{\Phi}_n \)), depending on whether \( S_n \) or \( \bar{S}_n \) is used in definition 72. If it is sharing, then the following structure is generated:
\[
\bar{\Phi}_nX_{Y_1}\cdots Y_nZ \rightarrow X(Y_1Z^{(n-1)})\cdots(Y_{n-1}Z')(Y_nZ)
\]  \( (73) \)

**Equivalences:**
\[
\Phi_{n+1} = B_0 \circ B
\]  \( (74) \)
\[
\Phi_n^mX_{Y_1}\cdots Y_nZ_1\cdots Z_m \rightarrow X(Y_1Z_1\cdots Z_m)\cdots(Y_nZ_1\cdots Z_m)
\]  \( (75) \)
\[
\Phi_{n+1}^mL_{X_1}\cdots Y_1Z_1\cdots Z_m \rightarrow XZ_1\cdots Z_m(Y_1Z_1\cdots Z_m)\cdots(Y_nZ_1\cdots Z_m)
\]  \( (76) \)

26 \( \Psi \) Combinator (\( \Psi \) Formalizer)

**Definition:**
\[
\PsiXYUV \rightarrow X(YU)(YV)
\]  \( (77) \)

**Reduction to SK:**
\[
\Psi = B(BW(BC))(BB(BB))
\]  \( (78) \)

**Size:** \( 26S + 24K + 49A = 99 \) total.
Other Notation

27 Composition

Definition:
\[ X \circ Y = BXY \]

Size: \(2S + 2K + 5A = 9\) total, plus \(|X| + |Y|\).

Equivalences:
\[
\begin{align*}
X \circ 1 &= 1 \circ X = X \\
X \circ (Y \circ Z) &= (X \circ Y) \circ Z \\
B(X \circ Y) &= BX \circ BY
\end{align*}
\]

28 Powers

Definition:
\[
\begin{align*}
X^0 &= I \\
X^1 &= X \\
X^{n+1} &= X \circ X^n
\end{align*}
\]

Size: \(2(n - 1)S + 2(n - 1)K + 5(n - 1)A = 9(n - 1)\) total, plus \(n|X|\), for \(X^n, n \geq 1\).

Equivalences:
\[
\begin{align*}
X^m \circ X^n &= X^{m+n} \\
(X^m)^n &= X^{mn} \\
(BX)^m &= B(X^m)
\end{align*}
\]

29 Deferred Combinators

Definition:
\[
\begin{align*}
X_{(0)} &= X \\
X_{(n+1)} &= BX_n
\end{align*}
\]

Size: \(2nS + 2nK + 4nA = 8n\) total, plus \(|X|\), for \(X_{(n)}\).
Equivalences:

\[ F_{(n)}X_0X_1 \cdots X_n \implies F(X_0X_1 \cdots X_n) \quad (93) \]
\[ X_{(m+n)} = B^m X_{(n)} \quad (94) \]

Notes: If \( F \) is regular (p. 2), \( FXY_1 \cdots Y_n \implies XZ_1 \cdots Z_m \), then

\[ F_{(k)}GX_1 \cdots X_kY_1 \cdots Y_n \implies GX_1 \cdots X_kZ_1 \cdots Z_m. \]

That is, \( F_{(k)} \) defers the action of \( F \) by \( k \) steps. Since \( B, C, I, K, \) and \( W \) are regular:

\[ B_{(n)}FX_1 \cdots X_nGY \implies FX_1 \cdots X_n(GY) \quad (95) \]
\[ C_{(n)}FX_1 \cdots X_nYZ \implies FX_1 \cdots X_nZY \quad (96) \]
\[ I_{(n)}X_0 \cdots X_n \implies X_0 \cdots X_n \quad (97) \]
\[ K_{(n)}X_0 \cdots X_nY \implies X_0 \cdots X_n \quad (98) \]
\[ W_{(n)}FX_1 \cdots X_nY \implies FX_1 \cdots X_nYY \quad (99) \]

30 Left Reduction

Definition:

\[ X_{[0]} = I \quad (100) \]
\[ X_{[1]} = X \quad (101) \]
\[ X_{[n+1]} = B_{[n]} \circ X \quad (102) \]

Size: \( 4(n - 1)S + 4(n - 1)K + 9(n - 1)A = 17(n - 1) \) total, plus \( n[J] \), for \( X_{[n]} \).

Equivalences:

\[ F_{[n]}X_0X_1 \cdots X_n \implies F(F \cdots (F(FX_0X_1)X_2) \cdots X_{n-1})X_n \quad (103) \]
\[ F_{[n+1]}X_0X_1 \cdots X_n \implies F_{[n]}(FX_0X_1)X_2 \cdots X_n \quad (104) \]
\[ X_{[n+1]} = B^nX \circ B^{n-1}X \circ \cdots \circ B^2X \circ BX \circ X \quad (105) \]
\[ X_{[n+1]} = X_{(n)} \circ X_{(n-1)} \circ \cdots \circ X_{(2)} \circ X_{(1)} \circ X_{(0)} \quad (106) \]
\[ X_{[n+1]} = (CB^2X)^nX \quad (107) \]
\[ X_{[m+n]} = B^mX_{[n]} \circ X_{[m]} \quad (108) \]
\[ C_{[n]}FX_1 \cdots X_nX_{n+1} \implies FX_{n+1}X_1 \cdots X_n \quad (109) \]
\[ S_{[n]} = S_n \quad (110) \]

Notes: \( F_{[n]} \) can be called a *left reduction* [Mac90]. To see this, write \( F \) in infix form, \( Fxy = x \circ y \) and assume \( \circ \) associates to the left (so \( x \circ y \circ z = (x \circ y) \circ z \)). Then:

\[ F_{[n]}x_0x_1 \cdots x_n = x_0 \circ x_1 \circ \cdots \circ x_n. \]

For \( F \) regular,

\[ F_{[n]} = (CB^2F)I \quad (111) \]
31 Polyadic Extension

Definition:

\[ X^0 = 1 \]  
\[ X^1 = X \]  
\[ X^{n+1} = X \circ BX^n \]  
\[ (X^{[n+1]} = (B^2XB)^nX \]  
\[ C^nF X_1X_2 \cdots X_{n+1} \implies FX_2 \cdots X_{n+1}X_1 \]

Size: \( 4(n-1)S + 4(n-1)K + 9(n-1)A = 17(n-1) \) total, plus \( n \| X \| \), for \( X^{[n]} \).

Equivalences:

Notes: If \( F \) is regular,

\[ F^n = (B^2XB)^nI \]  
\[ F^{[n+1]} = F \circ BF \circ \cdots \circ B^nF \]  
\[ F^{[n+1]} = F_0 \circ F_1 \circ \cdots \circ F_n \]  
\[ F^{[m+n]} = F^m \circ B^mF^n \]

References


