プログラミング-言語・基礎・実践- 16－15
(1994. 3. 10)

並列プログラムを生成するプログラム変換手法について

西田誠幸　辻野嘉宏　都倉信樹

大阪大学基礎工学部情報工学科
560 豊中市待兼山町 1-1

E-mail: s-nisita@ics.es.osaka-u.ac.jp

あらまし　Bird-Meertens formalism (以下 BMF) は関数型の計算モデルであり, 問題の仕様記述から具体的な手続きの記述を導出するのに用いることができる. その導出過程では記述の意味を深く意識する必要がなく, 等式変形による導出が可能である. BMF は逐次の計算記述のために考えられたのだが, 1) アーキテクチャに独立している, 2) 様々なアーキテクチャにおける実行コストを反映できる, という性質を持っているので, 実用的な並列計算モデルとして期待できる. 本稿では並列計算を表現するいくつかの演算子を新たに定義し, それらを用いて, 多項式,FFT の問題の仕様記述から並列処理の手続きの記述への導出を行なった.

キーワード　Bird-Meertens formalism, 並列プログラム, プログラム変換, リスト, 関数型プログラム, 並列計算モデル

On a problem transformation method generating parallel programs

Seikoh Nishita　Yoshihiro Tsujino　Nobuki Tokura

Department of Information and Computer Sciences, Faculty of Engineering Science
Osaka University
Toyonaka, Osaka 560, Japan

E-mail: s-nisita@ics.es.osaka-u.ac.jp

Abstract　Bird-Meertens formalism(BMF) can be used to derive concrete procedures from problem specifications by equational transformations within the framework of sequential computations. BMF has desirable characteristics : 1) independent of architectural models. 2) reflect the cost of the underlying execution. The formalism described in this paper is an extend BMF with additional operations which represent parallel computation. It is shown that parallel computational procedures can be derived from the naive problem specification with respect to polynomial evaluation and FFT.

key word　Bird-Meertens formalism, parallel program, program transformation, list, functional program, parallel model of computation

1 はじめに

並列計算機に対する計算モデルは2つのグループに大別できる. 1つはPRAMなどの理論的なモデルであり, その目的は問題がどの程度並列処理に適しているかを調べるところにある. もう1つはメッシュやハイパーキューブなどの実際の並列計算機を直接抽象化したモデルであり, その目的はモデルの上で作られたソフトウェアを実在する並列計算機上で走らせるところにある. 前者はアーキテクチャからは独立しているが, 実際の計算機の振舞とは直接には関係していない. 一方, 後者は抽象化したもとのアーキテクチャにおける計算を忠実に表現しているが, 例えばメッシュのアルゴリズムをハイパーキューブで用いることができないように, 別の種類のアーキテクチャに対する計算を表現できるほど抽象化されていない. 計算モデルは, 実際の計算機とは独立に問題のクラスや計算量を論じることができるものであるとともに, モデル上での計算が実際の計算機による計算を反映したものであることが望ましいが, 並列計算モデルの前述のような2極化は並列計算機の実用を妨げる大きな要因の1つといえる.

Bird-Meertens formalism [2, 1](以下 BMF) は関数型の計算モデルであり, 逐次計算の記述のために考えられたものであるが, リストを扱うことができること, 並列計算を行なう関数を考えられることなどから, 並列計算モデルとして捉えることができる. 並列計算モデルとしての BMF は 1) アーキテクチャから独立している, 2) 様々なアーキテクチャに対する記述の計算量を求めることができる, などの性質を持つため実用的な並列計算モデルとして期待できる.

本稿では, 並列計算モデルとしての BMF の有用性を示すために, BMF を用いて 1. 並列の計算を表す関数を含めて, 様々な関数を表現するための演算子を定義した, 2. 多項式,FFT(Fast Fourier Transform) について, 問題の仕様記述から, 具体的な逐次, 並列の計算記述を導出した.

2 モデル

2.1 BMF の記述

まず関数に関する記述の定義をする. 関数 f が値 a を与えられて返す値 $f(a)$ を "fa" または "$f \blacktriangleright a$" と記述する. また値に対して関数 f, 関数 g と順に適用する合成関数を "$f \cdot g$" と記述する. 特に関数 f の n 回の合成を "$f^{(n)}$" と記述する. さらに2項演算子 $\oplus$ に対して "$(a\oplus)$", "$(\oplus b)$" を値 c からそれぞれ $(a \oplus c)$, $(c \oplus b)$ を返す関数の記述とする.

同じ型の値の線形の並びをリストと呼ぶ. またリストを成分に持つリストを **2 重リスト**と呼ぶ. 記述 "$[a_m, a_{m+1}, \ldots, a_n]$" は値 a_i を第 $i-m+1$ 成分に持つリストを表す $(m \leq n, i = m, m+1, \ldots, n)$. また同じリストを "$[a_i]_{i=m}^{n}$", あるいは "$[a_i]_m^n$" と記述する.

括弧による冗長な記述を避けるために演算子の結合の強弱を決める. 演算子の結合の強さは, もっとも強いものが 関数, 2 項演算子に対する演算子の適用, 続いて値に対する演算子の適用と関数の合成であり, もっとも弱いものは値に対する 2 項演算子の適用である.

2.2 演算子 [1, 2]

論文 [1, 2] ですでに論じられている演算子を定義し, 各演算子に関する等式を示す.

[定義 1] リストの長さを求める演算子を "#" と表す. ■

[定義 2] リストの連接を行なう演算子を "$+\!\!+$" と表す. ■

[定義 3] (写像) リストのすべての成分に関数 f を適用する演算子を "$f*$" と表す. ■

写像に関しては次の等式が成り立つ.

[等式 4] $(g \cdot f)* = g* \cdot f*$ ■

[等式 5] $f*[a_i]_1^n = [fa_i]_1^n$ ■

[定義 6] (縮約) 結合律を満たす 2 項演算子 $\oplus$ に対して, リストの成分の $\oplus$ に関する和をとる演算子を "$\oplus/$" と表す. ■

縮約に関して次の等式が成り立つ.

[等式 7] $\oplus/[a_1, a_2, \ldots, a_n] = a_1 \oplus a_2 \oplus \cdots \oplus a_n$ ($\oplus$ は結合律を満たす) ■

[等式 8] $\oplus/ \cdot +\!\!+/ = \oplus/ \cdot (\oplus/)*$ ■

[定義 9] 準同形写像 h とはモノイド $(\alpha, \oplus)$ からモノイド $(\beta, \otimes)$ への写像のうち以下の式を満たすものという.

$$\begin{aligned} he_{\oplus} &= e_{\otimes} \\ h(a \oplus b) &= ha \otimes hb \end{aligned}$$

ただし,$e_{\oplus}$ は$\oplus$ の単位元, $e_{\otimes}$ は$\otimes$ の単位元, a, b は D の要素. ■

[等式 10] モノイド $([\alpha], +\!\!+)$ からモノイド $(\beta, \otimes)$ への準同形写像 h は次式を満たす.

$h = (\otimes/) \cdot f*$ (ただし $h[a] = fa$) ■

[等式 11] モノイド $(\alpha, \oplus)$ からモノイド $(\beta, \otimes)$ への準同形写像 h は次式を満たす.

$h \cdot \oplus/ = \otimes/ \cdot h*$ ■

2.3 BMF の演算子

本節では新たに定義した演算子などの性質について簡単に述べる (詳細は [3]).

[定義 12] (多重適用) 共通の値に関数リスト $\mathcal{F}$ の各成分関数を適用する演算子を "$\mathcal{F}\bar{*}$" と記述する. すなわち,

$$\begin{aligned} []\bar{*}a &= [] \\ [f]\bar{*}a &= [fa] \\ (\mathcal{F}+\!\!+\mathcal{G})\bar{*}a &= \mathcal{F}\bar{*}a +\!\!+ \mathcal{G}\bar{*}a \end{aligned}$$

■

多重適用は関数リストの各成分関数を同一の値に適用する関数である.

[定理 13] $[f_i]_1^n \bar{*} a = [f_i a]_1^n$ ■

[定義 14] (ポイントワイズ演算) 2 つのリストの第 i 成分どうしに 2 項演算子$\oplus$ を適用する演算子を$\curlyvee_{\oplus}$ と表す.

$$\begin{aligned} []\curlyvee_{\oplus}[] &= [] \\ [a]\curlyvee_{\oplus}[b] &= [a \oplus b] \\ (x_1 +\!\!+ x_2)\curlyvee_{\oplus}(y_1 +\!\!+ y_2) & \\ &= (x_1 \curlyvee_{\oplus} y_1) +\!\!+ (x_2 \curlyvee_{\oplus} y_2) \end{aligned}$$

(ただし$\#x_1 = \#y_1, \#x_2 = \#y_2$) ■

[定理 15] $[a_i]_1^n \curlyvee_{\oplus} [b_i]_1^n = [a_i \oplus b_i]_1^n$($n$ は非負整数) ■

[定義 16] 演算子 "$\propto$" は転置を行なう関数を表す. すなわち値 a_{ij} $(i = 1, 2, \ldots, m, j = 1, 2, \ldots, n)$ に対して

$$\propto \left[[a_{ij}]_{i=1}^m\right]_{j=1}^n = \left[[a_{ij}]_{j=1}^n\right]_{i=1}^m$$

■

[定理 17] $\propto \cdot \propto = id$ ■

[定理 18] $\curlyvee_{\oplus}/ = \oplus/* \cdot \propto$ ($\oplus$ は結合律を満たす) ■

[証明] 2 重リスト $[[a_{ij}]_{i=1}^m]_{j=1}^n$ に対して $\curlyvee_{\oplus}/$ を適用することを考える. もともと a_{ij} は "第 j 成分リスト $[a_{ij}]_{i=1}^m$ の第 i 番目の成分" である. 2 重リストに転置を適用すると a_{ij} は "第 i 成分リスト $[a_{ij}]_{j=1}^n$ の第 j 成分" となる. この i 番目の成分リスト に対して $\oplus$ に関する和をとることは, もともとの 2 重リストの成分リスト $[a_{ij}]_{i=1}^m$ $(j = 1, 2, \ldots, n)$ の第 i 成分どうしの和をとること, すなわち $\curlyvee_{\oplus}/$ を適用することに等しい. ■

[定義 19] リストの各第 i 成分にそれぞれ $f^{(i-1)}$ を適用する演算子を "$f\Leftarrow$"(多重合成のリスト適用) と表す. すなわち,

$$\begin{aligned} F[] &= [] \\ F([a] +\!\!+ x) &= [a] +\!\!+ (f*) \cdot Fx \end{aligned}$$

■

[定理 20] $f\Leftarrow x = [f^{(i)}]_0^{(\#x-1)} \curlyvee_{\blacktriangleright} x$ ■

[系 21] $f\Leftarrow[a_i]_1^n = [f^{(i-1)} a_i,]_1^n$ ■

[定理 22] 関数 f, g に対して, 関数 h が存在し, 次式を満たすとする.

$$g \cdot f = h \cdot g \qquad (1)$$

このとき

$g* \cdot (f\Leftarrow n) = (h\Leftarrow n) \cdot g*$ (n は非負整数)

■

[証明] リストの第 i 成分に着目する. リストに関数 $g* \cdot (f\Leftarrow n)$ を適用することによって, 第

i 成分には関数 $g \cdot f^{(i-1)}$ が施される. 仮定の式より, $g \cdot f^{(i-1)} = h^{(i-1)} \cdot g$ が成り立つ. 右辺を第 i 成分に施すことは, リストに対して定理の右辺の関数を適用することに等しい. ■

[定理 23] $(f*) \Leftarrow \cdot \propto = \propto \cdot (f \Leftarrow)*$ ■

[証明] 2 重リスト $[[a_{ij}]_{i=1}^{m}]_{j=1}^{n}$ に定理の式の右辺を適用すると,$[[f^{(}i-1)a_{ij}]_{j=1}^{n}]_{i=1}^{m}$ となる. これは $[(f^{(n-1)})*[a_{ij}]_{j=1}^{n}]_{i=1}^{m}$ に等しい. よって系 21, 定義 16 を順に用いて, この 2 重リストが, 定理の右辺に 2 重リスト $[[a_{ij}]_{i=1}^{m}]_{j=1}^{n}$ を適用して返されるものに等しいと示すことができる. ■

[定義 24] リストの成分の種類, 内訳を変えず, 成分の並び順を変える演算を置換と呼ぶ. ■

交換律を満たす 2 項演算子の縮約はリストの並び順に左右されない.

[定理 25] $\oplus/ = \oplus/ \cdot \rho$ (ただしρは任意の置換 ,$\oplus$は結合, 交換律を満たす 2 項演算子.) ■

[証明] $\oplus$ に関するリストの成分の和 $\oplus/$ の結果は $\oplus$ に交換律より, 適用するリストの並び順に左右されない. ゆえに定理が成り立つ. ■

[定理 26] $\exists \rho (+\!\!+/ \cdot \propto xs = \rho \cdot +\!\!+/xs)$ (xs は成分リスト長が一定の 2 重リスト, ρは置換) ■

[証明] 2 つのリスト $(+\!\!+/ \cdot \propto xs)$ $(+\!\!+/xs)$ は互いに成分の種類内訳が等しい. よって 2 つのリストの間には置換が存在する. ■

[定義 27] リスト x の第 $n \times i$ 成分から第 $n \times i+n-1$ 成分 $(i = 0, 1, \ldots, m-1)$ を長さ n の 1 つのリストとして分割し, それらのリストからなる 2 重リストを返す演算子 を $div\ n$(分割) とする. ■

多重合成のリスト適用の分割統治法を表す計算記述を示す.

[定理 28]

$f \Leftarrow = \left(+\!\!+/ \cdot (f \Leftarrow)* \cdot \propto\right) \cdot (f^{(n)} \Leftarrow)* \cdot \propto \cdot (div\ n)$

■

定理の右辺は, $(f^{(n)} \Leftarrow)* \cdot \propto \cdot (div\ n)$ が分割を, $\left(+\!\!+/ \cdot (f \Leftarrow)* \cdot \propto\right)$ が統治を表しており, 全体として分割統治法の計算を表している.

3 単位環に対する多項式

本章では, 単位環に対する多項式を評価する問題の仕様記述を求め, 仕様記述から逐次, 並列計算の記述を導出する.

3.1 単位環

単位環 $(D, \oplus, \otimes)$ は乗法に関する単位元をもつ環である. 記述を簡単にするために, 乗法 $\otimes$ の演算は, 加法 $\oplus$ に関する演算に優先するものとする.

単位環の性質のうち以下で用いるものについて述べる.

- 加法演算子 $\oplus$ についての交換律:

$$(a\oplus) = (\oplus a)(\text{ただし } a \in D) \quad (2)$$

- 加法の単位元 $e_\oplus$ は乗法の零元:

$$a \otimes e_\oplus = e_\oplus \otimes a = e_\oplus \quad (3)$$

- 乗法, 加法の分配律:

$$(\otimes a)(b \oplus c) = (\otimes a)b \ \oplus \ (\otimes a)c \quad (4)$$

(4),(3) 等式 11 より

$$(\otimes a) \cdot \oplus/ = \oplus/ \cdot (\otimes a)* \quad (5)$$

式 (5) と定理 22 より

$$\oplus/* \cdot \big((\otimes a)*\big) \Leftarrow = (\otimes a) \Leftarrow \cdot \oplus/* \quad (6)$$

3.2 単位環における多項式

[定義 29] 単位環 $(D, \oplus, \otimes)$ の多項式は次の式で定義される.

$$a_0 \otimes b^0 \ \oplus \ a_1 \otimes b^1 \ \oplus \cdots \oplus \ a_{n-1} \otimes b^{n-1}$$
$$(a_i, b \in D) \quad ■$$

ここではこの多項式の定義を "多項式を評価する問題の仕様記述" とする.

[定義 30] 多項式を作る関数を$poly$ とする. すなわち,

$$(poly\ b)[a_0, a_1, \ldots, a_{n-1}]$$
$$= a_0 \otimes b^0 \ \oplus \ a_1 \otimes b^1 \ \oplus \cdots \oplus \ a_{n-1} \otimes b^{n-1}$$

■

3.3 多項式の並列計算

等式 7, 系 21 より多項式について次の等式が成り立つ.

$$(poly\ b)[a_i]_0^{n-1} = \oplus/ \ \cdot \ (\otimes b) \Leftarrow [a_i]_0^{n-1} \quad (7)$$

(ただし $a_i, b \in D$)

前式と分割統治法の定理を使って多項式の並列計算の記述を導出する.

[定理 31] 長さ $2n$ のリスト $x = [a_i]_0^{2n-1}$ に対して, $even_x = [a_{2i}]_0^{n-1}$, $odd_x = [a_{2i+1}]_0^{n-1}$ とする. このとき

$$(poly\ b)x = \oplus/ \cdot \left([id, \otimes b]\curlyvee_{\blacktriangleright}\right)\left[(poly\ b^2)even_x, (poly\ b^2)odd_x\right]$$

■

[証明] 準備として補題を示す.

[補題 32] $\propto \cdot (div\ 2)x = [even_x, odd_x]$ ■

補題の証明は省略する. 関数 F を $(\propto \cdot (div\ 2))$ とし, 関数 $(poly\ b)$ について調べる.

$$\begin{aligned}
&(poly\ b) = \oplus/ \cdot (\otimes b) \Leftarrow x \\
&\qquad\qquad \text{(式 (7) より)} \\
&= \oplus/ \cdot +\!\!+/ \cdot (\otimes b) \Leftarrow * \cdot \propto \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(定理 28 より)} \\
&= \oplus/ \cdot +\!\!+/ \cdot (\otimes b) \Leftarrow * \cdot \propto \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(式 (32) より)} \\
&= \oplus/ \cdot +\!\!+/ \cdot \propto \cdot \left((\otimes b) *\right) \Leftarrow \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(定理 23 より)} \\
&= \oplus/ \cdot \rho \cdot +\!\!+/ \cdot \left((\otimes b) *\right) \Leftarrow \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(定理 26 より)} \\
&= \oplus/ \cdot +\!\!+/ \cdot \left((\otimes b) *\right) \Leftarrow \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(定理 (25) より)} \\
&= \oplus/ \cdot \oplus/ * \cdot \left((\otimes b) *\right) \Leftarrow \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(等式 8 より)} \\
&= \oplus/ \cdot (\otimes b) \Leftarrow \cdot \oplus/ * \cdot \left((\otimes b)^{(2)} \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(式 (6) より)} \\
&= \oplus/ \cdot (\otimes b) \Leftarrow \cdot \left(\oplus/ \cdot (\otimes b^2) \Leftarrow\right) * \cdot F \\
&\qquad\qquad \text{(等式 4 より)} \\
&= \oplus/ \cdot (\otimes b) \Leftarrow \cdot (poly\ b^2) * \cdot F \\
&\qquad\qquad \text{(式 (7) より)} \\
&\qquad\qquad\qquad\qquad (8)
\end{aligned}$$

定理の式の左辺から右辺を導出する.

$$\begin{aligned}
&\text{(左辺)} = (poly\ b)x \\
&= \oplus/ \cdot (\otimes b) \Leftarrow \cdot (poly\ b^2) * \cdot Fx \\
&\qquad\qquad \text{(式 (8) より)} \\
&= \oplus/ \cdot (\otimes b) \Leftarrow \cdot (poly\ b^2) * \cdot [even_x, odd_x] \\
&\qquad\qquad \text{(補題 32 より)} \\
&= \oplus/ \cdot (\otimes b) \cdot [(poly\ b^2)even_x, (poly\ b^2)odd_x] \\
&\qquad\qquad \text{(等式 5 より)} \\
&= \oplus/ \cdot [id, \oplus b]\curlyvee_{\blacktriangleright} [(poly\ b^2)even_x, (poly\ b^2)odd_x] \\
&\qquad\qquad \text{(定理 20 より)} \\
&= \text{(右辺)}
\end{aligned}$$

■

定理 31 にしたがって, 再帰的に多項式を計算するときの, 並列の計算ステップ数およびプロセッサ数を考える. 長さ n のリストに対する $poly$ の実行ステップ数を $T(n)$, 使用プロセッサ数を $S(n)$ とおくと, 定理の記述より次の式が得られる.

$$T(2n) = T(n) + 4c, \quad T(1) = O(1)$$
$$S(2n) = 2 \times S(n), \quad S(1) = 1$$

これらの式より

$$T(n) = O(\log n), \quad S(n) = n$$

すなわち定理 31 で導かれる並列計算法では, $n-1$ 次の多項式を計算するのに n 個のプロセッサで $O(\log n)$ ステップかかる.

4 FFT

本節では FFT(Fast Fourier Transform) の仕様記述を求め, 仕様記述から並列計算の記述を導出する.

4.1 複素数と 1 の n 乗根

FFT の計算は複素数の上で行なわれる.

[定義 33] 複素数 ω_n を次式で定義する.

$\omega_n = e^{2\pi i/n}$ (ただし $i^2 = -1$, e は自然対数)

このとき, $\omega_n^0, \omega_n^1, \ldots, \omega_n^{n-1}$ それぞれは 1 の n 乗根である. ■

1 の n 乗根に関して以下の性質が成り立つ.

[補題 34] $\omega_{dn}^{dk} = \omega_n^k$(ただし n, k, d は正の整数) ■

[系 35] $\omega_n^{n/2} = \omega_2 = -1$(ただし n は正の偶数) ■

4.2 DFT

本節では FFT の仕様記述に相当する DFT(Discrete Fourier Transform) について述べる. 議論はすべて複素数上で行なう. 複素数 $(\mathbb{C}, \times, +)$ は単位環なので, 関数 $poly$ を利用できる.

[定義 36] 実数のリスト $x = [a_0, a_1, \ldots, a_{n-1}]$ に対する DFT y とは x の成分を係数とする多項式に n 種類の 1 の n 乗根を代入して返される値からなるリストである. すなわち,

$$\begin{aligned}
y &= [y_0, y_1, \ldots y_{n-1}] \\
y_k &= (poly\ \omega_i^k)x
\end{aligned}$$

また長さ n のリスト x を入力とし, DFT y を出力する関数を$\mathcal{DFT}$ と記述する.

$$\mathcal{DFT}_n x = y = [(poly\ \omega_n^j)x]_0^{n-1}$$ ■

FFT とは 1 の複素根の性質を用いて DFT を高速に計算する手段である. この $\mathcal{DFT}$ を FFT の仕様記述として, 次節では $\mathcal{DFT}$ から FFT の並列計算の記述を導出する.

4.3 FFT の計算記述

本節では FFT の並列計算の記述を導出する. まず準備として, 補題を 3 つ示す.

[補題 37] $\left([\times a^j]_0^{n-1}\Upsilon_{\blacktriangleright}\right) = (\times a)\Leftarrow$
(ただし $a \in \mathbb{C}$) ■

[補題 38]
$\left([\times(-a^j)]_0^{n-1}\Upsilon_{\blacktriangleright}\right) = \left(\times(-1)\right)* \cdot (\times a)\Leftarrow$
(ただし $\in \mathbb{C}$) ■

[補題 39] $x\ \Upsilon_+\ \left(\times(-1)\right)*y = x\ \Upsilon_-\ y$
(ただし x, y は実数 のリスト) ■

FFT の計算記述を導出する.

[定理 40] 長さ $2n$ の実数のリスト $x = [a_i]_0^{2n-1}$ に対して, $even_x = [a_{2i}]_0^{n-1}$, $odd_x = [a_{2i+1}]_0^{n-1}$ とする. このとき

$$\mathcal{DFT}_{2n}x = +\!\!+/ \cdot [\Upsilon_+/, \Upsilon_-/]\bar{*} \cdot \left([id, (\times\omega_{2n})\Leftarrow]\Upsilon_{\blacktriangleright}\right)[\mathcal{DFT}_n even_x, \mathcal{DFT}_n odd_x]$$

■

[証明] 付録参照のこと. ■

5 あとがき

本論文では, Bird-Meertens formalism に新しい演算子を定義し, 演算子の性質を表す等式を導いた. また, 定義した演算子とその等式を用いて, 多項式を計算する問題,FFT の問題の仕様記述から, 具体的な逐次, 並列の計算記述を導出した.

今後の課題としては, 分割統治法のように導出に大きな方針を与えるような等式の探索, コストに関しての考察, 問題から仕様記述を得るための手法の開拓, 具体的なアーキテクチャに対して,BMF の手法を適用することなどがある.

参考文献

[1] Richard S.Bird, *Logic of Programming and Calculi of Discrete Design: An Introduction to the Theory of Lists*, volume F36 of *NATO ASI Series*, pp. 5–42, Springer-Verlag Berlin Heidelberg, 1987.

[2] Richard S.Bird, *Constructive Methods in Computing Science: Lectures on Constructive Functional Programing*, volume F55 of *NATO ASI Series*, pp. 151–216, Springer-Verlag Berlin Heidelberg, 1989.

[3] 西田 誠幸 : 並列プログラムを生成するプログラム変換手法, 平成 5 年度大阪大学大学院基礎工学研究科 (情報工学分野) 修士学位論文 ,(1994.2).

付録

(定理 40 の証明)

$\mathcal{DFT}_{2n}x$ について調べる.

$$\begin{aligned} DFT_{2n}x &= [(poly\ \omega_{2n}^{j})x]_{0}^{2n-1} && \text{(定義 36 より)} \\ &= [(poly\ \omega_{2n}^{j})x]_{0}^{n-1} +\!\!+ [(poly\ \omega_{2n}^{j})x]_{n}^{2n-1} \end{aligned} \tag{9}$$

ここで式 (9) の連接の前半を Fmr, 後半を Ltr とおき, 式を 2 つに分けてそれぞれを変形する.

i) Fmr について

$$\begin{aligned} Fmr &= [(poly\ \omega_{2n}^{j})x]_{0}^{n-1} \\ &= \left[+/\ \cdot\ ([id, \times\omega_{2n}^{j}]\Upsilon_{\blacktriangleright})\ [(poly\ \omega_{2n}^{2j})even_x\ ,\ (poly\ \omega_{2n}^{2j})odd_x]\right]_{0}^{n-1} && \text{(定理 31 より)} \\ &= \left[+/\ \cdot\ ([id, \times\omega_{2n}^{j}]\Upsilon_{\blacktriangleright})\ [(poly\ \omega_{n}^{j})even_x\ ,\ (poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(補題 34 より)} \\ &= +/* \left[[id, \times\omega_{2n}^{j}]\Upsilon_{\blacktriangleright}\ [(poly\ \omega_{n}^{j})even_x\ ,\ (poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(等式 5 より)} \\ &= +/* \left[[(poly\ \omega_{n}^{j})even_x\ ,\ (\times\omega_{2n}^{j})\cdot(poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(定理 15 より)} \\ &= +/* \cdot\ \propto\ \cdot\ \propto\ \left[[(poly\ \omega_{n}^{j})even_x\ ,\ (\times\omega_{2n}^{j})\cdot(poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(定理 17 より)} \\ &= +/* \cdot\ \propto \left[[(poly\ \omega_{n}^{j})even_x]_{0}^{n-1}\ ,\ [(\times\omega_{2n}^{j})\cdot(poly\ \omega_{n}^{j})odd_x]_{0}^{n-1}\right] && \text{(定義 16 より)} \\ &= +/* \cdot\ \propto \left[[(poly\ \omega_{n}^{j})even_x]_{0}^{n-1}\ ,\ [(\times\omega_{2n}^{j})]_{0}^{n-1}\Upsilon_{\blacktriangleright}\ [poly\ \omega_{n}^{j})odd_x]_{0}^{n-1}\right] && \text{(定理 15 より)} \\ &= +/* \cdot\ \propto \left[\mathcal{DFT}_n even_x\ ,\ [(\times\omega_{2n}^{j})]_{0}^{n-1}\Upsilon_{\blacktriangleright}\ \mathcal{DFT}_n odd_x\right] && \text{(定義 36 より)} \\ &= +/* \cdot\ \propto [\mathcal{DFT}_n even_x\ ,\ (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x] && \text{(補題 37 より)} \\ &= \Upsilon_{+}/\ [\mathcal{DFT}_n even_x\ ,\ (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x] && \text{(定理 18 より)} \end{aligned}$$

ii) Ltr について

$$\begin{aligned} Ltr &= [(poly\ \omega_{2n}^{j})x]_{n}^{2n-1} \\ &= [(poly\ \omega_{2n}^{n+j})x]_{0}^{n-1} \\ &= [(poly\ -\omega_{2n}^{j})x]_{0}^{n-1} && \text{(系 35より)} \\ &= \left[+/\ \cdot\ ([id, \times -\omega_{2n}^{j}]\Upsilon_{\blacktriangleright})\ [(poly\ \omega_{2n}^{2j})even_x\ ,\ (poly\ \omega_{2n}^{2j})odd_x]\right]_{0}^{n-1} && \text{(定理 31 より)} \\ &= \left[+/\ \cdot\ ([id, \times -\omega_{2n}^{j}]\Upsilon_{\blacktriangleright})\ [(poly\ \omega_{n}^{j})even_x\ ,\ (poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(補題 34 より)} \\ &= +/* \left[[id, \times -\omega_{2n}^{j}]\ \Upsilon_{\blacktriangleright}\ [(poly\ \omega_{n}^{j})even_x\ ,\ (poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(等式 5 より)} \\ &= +/* \left[[(poly\omega_{n}^{j})even_x\ ,\ (\times -\omega_{2n}^{j})\cdot(poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(定理 15 より)} \\ &= +/* \cdot\ \propto\ \cdot\ \propto\ \left[[(poly\ \omega_{n}^{j})even_x\ ,\ (\times -\omega_{2n}^{j})\cdot(poly\ \omega_{n}^{j})odd_x]\right]_{0}^{n-1} && \text{(定理 17 より)} \\ &= +/* \cdot\ \propto \left[[(poly\ \omega_{n}^{j})even_x]_{0}^{n-1}\ ,\ [(\times -\omega_{2n}^{j})\cdot(poly\ \omega_{n}^{j})odd_x]_{0}^{n-1}\right] && \text{(定義 16 より)} \\ &= +/* \cdot\ \propto \left[[(poly\ \omega_{n}^{j})even_x]_{0}^{n-1}, [(\times -\omega_{2n}^{j})]_{0}^{n-1}\Upsilon_{\blacktriangleright}\ [(poly\ \omega_{n}^{j})odd_x]_{0}^{n-1}\right] && \text{(定理 15 より)} \\ &= +/* \cdot\ \propto \left[\mathcal{DFT}_n even_x, [(\times -\omega_{2n}^{j})]_{0}^{n-1}\Upsilon_{\blacktriangleright}\ \mathcal{DFT}_n odd_x\right] && \text{(定義 36 より)} \\ &= +/* \cdot\ \propto [\mathcal{DFT}_n even_x, (\times -1)* \cdot\ (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x] && \text{(補題 38 より)} \\ &= \Upsilon_{+}/\ [\mathcal{DFT}_n even_x, (\times -1)* \cdot\ (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x] && \text{(定理 18 より)} \\ &= \mathcal{DFT}_n even_x\ \Upsilon_{+}\ (\times -1)* \cdot\ (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x && \text{(等式 7 より)} \\ &= \mathcal{DFT}_n even_x\ \Upsilon_{-}\ (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x && \text{(補題 39 より)} \\ &= \Upsilon_{-}/\ [\mathcal{DFT}_n even_x, (\times\omega_{2n})\Leftarrow \mathcal{DFT}_n odd_x] && \text{(等式 7 より)} \end{aligned}$$

i),ii) の結果を使って, 式 (9) を変形する.

$$
\begin{aligned}
&\mathcal{DFT}_{2n}x = Fmr +\!\!+ Ltr && \text{(式 (9) より)}\\
&= +\!\!+/\,[Fmr, Ltr] && \text{(等式 7 より)}\\
&= +\!\!+/\,\Big[\curlyvee_+/\,\big[\mathcal{DFT}_n even_x, (\times\omega_{2n}) \Leftarrow \mathcal{DFT}_n odd_x\big],\\
&\qquad\qquad \curlyvee_-/\,\big[\mathcal{DFT}_n even_x, (\times\omega_{2n}) \Leftarrow \mathcal{DFT}_n odd_x\big]\Big] && \text{(i),ii) より)}\\
&= +\!\!+/\,\cdot\,[\curlyvee_+/, \curlyvee_-/]\bar{*}\,\big[\mathcal{DFT}_n even_x, (\times\omega_{2n}) \Leftarrow \mathcal{DFT}_n odd_x\big] && \text{(定理 13 より)}\\
&= +\!\!+/\,\cdot\,[\curlyvee_+/, \curlyvee_-/]\bar{*}\,\cdot\,\big([id, (\times\omega_{2n})]\curlyvee_{\blacktriangleright}\big)\,[\mathcal{DFT}_n even_x, \mathcal{DFT}_n odd_x] && \text{(定理 15 より)}
\end{aligned}
$$

(証明終り)