数理解析研究所講究録
666巻 1988年 125-134

単項演算に対する局所計算可能な符号化

京大工学部　安浦　寛人（Hiroto Yasuura）

1．はじめに

並列アルゴリズムの設計においては，いかに計算の並列性を抽出し，並列に計算できる部分を独立に並列計算するかが重要な問題である．さらに，各々の独立した計算が局所的な情報のみに依存するようにできれば，個々の独立した計算は単純なものとなり，計算全体における通信量も小さくなる．このように，並列計算を局所的な通信のみを必要とする単純な計算の集まりとすることができれば，通信や計算に要する時間や資源を小さくすることができる．

このようなアイディアに基づき，（１）では，局所計算可能性(Local Computability)という概念を提案し，それを実現するために符号化の工夫が有効であることを示した．集合Ｓとその上での演算ｆが与えられたとき，ある符号化Ｃのもとで，ｆによる演算結果の各桁が，オペランドの高々ℓ桁にのみ依存するような計算規則が作れるとき，演算ｆは符号化Ｃのもとでℓ－局所計算可能であるという．（Ｓ，ｆ）が有限可換群であれば，Ｓの要素数にかかわらず$\ell=28$となる２値の等長符号化が存在する[1]．この符号化はＳの各要素に２つ以上の符号を割り当てる冗長符号化であり，この冗長性が本質的に，計算の局所化に寄与してい

る．

一般的に，与えられた代数系に対し，局所計算可能性を実現する符号化を見つける問題は，並列アルゴリズムの設計の基本問題の一つである．上記の結果は，有限可換群とその上の２項演算だけが対象であったが，より広い枠組みの上での議論が必要である．ここでは，最も単純な場合の一つとして，有限集合とその上で定義される複数の単項演算を対象として，局所計算可能性を実現する符号化について議論する．

２．諸定義

Ｓを有限集合とする．Ｓの要素数をｎとする．$\{OP_1, OP_2, \cdots, OP_m\}$ をＳ上で定義された単項演算とする．すなわち，$OP_i : S \to S$である．Σを有限のアルファベットとし，その要素数をａとする．Ｓに対するΣ上の符号化$\mathcal{C}$は，Σ^{ℓ} の部分集合からＳへの全射（上への写像）として定義する．ℓを符号長と呼ぶ．ここでは，等長符号のみを考える．

$c \in \Sigma^{\ell}$，$s \in S$について$\mathcal{C}(c) = s$であるとき，ｃをｓに対する符号という．Σ^{ℓ} の要素ｃで$\mathcal{C}(c)$が定義されていないｃを非符号語と呼ぶ．$\ell = \lceil \log_a n \rceil$のとき，$\mathcal{C}$を最短符号化と呼ぶ．また，あるｓに対応する符号が２つ以上あるとき，$\mathcal{C}$は冗長であるという．

$f : \Sigma^{\ell} \to \Sigma^{\ell}$ に対し，$\mathcal{C}(f(c)) = OP(\mathcal{C}(c))$が成り立つとき，fは符号化$\mathcal{C}$の下でＳ上の単項演算OPを実現するという．fは

$(z_1, z_2, \cdots, z_k) = f(x_1, x_2, \cdots, x_k)$

$= (g_1(x_1, x_2, \cdots, x_k), g_2(x_1, x_2, \cdots, x_k), , g_k(x_1, x_2, \cdots, x_k))$

と書ける．すなわち，

$z_j = g_j(x_1, x_2, \cdots, x_k)$

とかける．各g_j(j=1,2,…,k)が高々ℓ個の入力変数にしか依存しないとき，fはℓ一局所計算可能(ℓ-locally computable)であるという．S上の単項演算OPが符号化$\mathcal{C}$の下でℓ一局所計算可能であるとは，OPを実現するℓ一局所計算可能な関数fが存在することである．

［定義１］S上の単項演算の集合 $\{OP_1, OP_2, \cdots, OP_m\}$ が符号化$\mathcal{C}$の下でℓ一弱局所計算可能(weakly ℓ-locally computable)であるとは，各演算OP_iがℓ一局所計算可能であることである．

［定義２］S上の単項演算の集合 $\{OP_1, OP_2, \cdots, OP_m\}$ が符号化$\mathcal{C}$の下でℓ一弱局所計算可能であるとき，各演算OP_iを実現する関数f_i(i=1,2,…,m)において，各z_j(j=1,2,…,k)が依存するℓ個の入力変数を共通にすることができるとき，単項演算の集合は符号化$\mathcal{C}$の下でℓ一強局所計算可能(strongly ℓ-locally computable)であるという．

以下の議論は簡単のため２値符号化（$\Sigma = \{0,1\}$）を仮定するが，一般性は失わない．

3．非冗長な最短符号

k=⌈$\log_2 n$⌉かつ非冗長符号で，最も一般的に用いられている符号化について考える．強い制限のため，局所計算可能性は一般には小さくできない．

［定理１］非冗長な最短符号化のもとでは，弱局所計算可能性に関して，$\ell < \ell_c$（=⌈$\log_2 n$⌉）とはできない単項演算の集合が存在する．

（証明）$n=2^p$（pは正整数）とする．図１のようにある演算OPでs_0はs_0，その他の状態はs_1であるとする．$\ell < \ell_c$と仮定する．s_0に割り当てられる符号を(0,0,...,0)としても一般性を失わない．s_1に割り当てられる符号を$(a_1,a_2,...,a_k)$とする．このベクトル中でi番目の要素が0でなかったとする．ここで，仮定よりx_iが依存しない変数が少なくとも１つ存在するから，それをx_jとする．符号の中でj番目の要素だけが1であり，他のすべての要素が0であるものを考える．この符号を割り当てられたSの要素をs_jとすると，状態変数x_iはx_jに依存しないから，演算OPの下でs_jとs_0の次状態のx_iの値は一致しなければならない．しかし，$a_i=1$と選んだので矛盾である．□

4．非冗長符号

最短符号の制限をはずすと符号化の自由度は大きく増大する．実用的にも，１ホット符号等は広く用いられている．

要素	単項演算	符号語
	.. OP...	X_0 X_1 X_i X_j X_{k-1}
S_0	S_0	0 0 ...0 ...0 ...0
S_1	S_1	a_1 a_2...1 ...a_j...a_{k-1}
S_2	S_1	
S_j	S_1	0 0 ...0 ...1 ...0
S_{n-1}	S_1	

図１．定理１の証明

［定理２］複数の単項演算が定義された任意の集合Ｓに対し，１－弱局所計算可能となる符号化が構成できる．

（証明）非冗長符号化のもとでは，符号の各桁x_iは，Ｓの上の１つの2ブロック分割に対応している．任意の2ブロック分割$\{S_1,S_2\}$について，各演算における遷移に対応して，S_3の各要素はS_1中の要素に遷移し，S_4中の要素はS_2へ遷移するような2ブロック分割$\{S_3,S_4\}$が存在する．すべての２ブロック分割に対応するように各桁を選んだ符号長2^nの非冗長符号は，任意の単項演算の集合を１－弱局所計算可能とする．実用的には，与えられた単項演算の集合に対して2^nより小さな必要

最小限の符号長を求めることができる．□

［定理３］ｍ個の単項演算が定義された任意の集合Ｓに対し，min（ｍ,「$\log_2 n$┐）一強局所計算可能となる符号化が構成できる．但しｎはＳの要素数である．

（証明）定理２の証明と同様の議論で，一般に，

$$z_i = g_j(x_1, x_2, \cdots, x_k)$$

$$= OP_1 x_{i1}^* + OP_2 x_{i2}^* + \cdots + OP_m x_{im}^* \qquad (1)$$

（x^*はxの肯定または否定を表す）となるように符号化を決めることができる[2]．ｍ<「$\log_2 n$┐の時は(1)式を満たす符号化，それ以外は任意の最短符号化でよい．□

定理３で与えられるℓの最小値の上界が上限となるような集合が在る．

［定理４］非冗長符号化のもとでは，強局所計算可能性ℓをmin（ｍ,「$\log_2 n$┐）より小さくできない集合Ｓが存在する．

（証明）「$\log_2 n$┐<ｍの場合は，次のような演算が定義された集合を考える．但し，$n=2^p$(pは正整数)とする．Ｓの要素s_j（j=0,1,...,n-1）の演算OP_i（i=0,1,...,「$\log_2 n$┐-1）における遷移先は，jの「$\log_2 n$┐ビットの２進数表現の2^iの桁が0のときs_0，1のときs_1である．状態s_0とs_1を分離する符号語の桁zは少なくとも1個存在する．このとき，$s \neq s'$なる任意のＳの要素について，演算OP_i（i=0,1,...,「$\log_2 n$┐-1）のいずれかで，これらの遷移先は異なるので，zはすべての要素を

分離するのに必要な少なくとも「$\log_2 n$┐個の入力変数に依存する（図２参照）．次に「$\log_2 n$┐≧mの場合を考える．同様にn要素の集合Ｓ’（$n=2^p$　pは正整数）を考える．Ｓ’の状態s_j（$j=0,1,\ldots,n-1$）の入力OP_i（$i=0,1,\ldots,m-1$）における遷移先は，jの「$\log_2 n$┐ビットの２進数表現の2^iの桁が0のときs_0，1のときs_1である．要素s_0とs_1を分離する桁zは少なくとも1個存在する．Ｓ’の要素は2^m個の部分集合に分かれ，異なる集合に属する状態においては，ある適当な演算によって遷移先が異なる．すなわち，zは2^m個の部分集合を分離するのに必要なm個以上の変数に依存する．□

要素	単項演算 $OP_0 OP_1 OP_2 .. OP_{p-1} .. OP_m$	符号語 ... z ...
S_0	$S_0\ S_0\ S_0 \ldots S_0$	0
S_1	$S_1\ S_0\ S_0 \ldots S_0$	1
S_2	$S_0\ S_1\ S_0 \ldots S_0$	
S_3	$S_1\ S_1\ S_0 \ldots S_0$	
S_n	$S_1\ S_1\ S_1 \ldots S_1$	

図２．定理４の証明

5．冗長符号

冗長符号化は，１つの要素に複数の符号を割り当てることを許すもので，符号化の自由度は飛躍的に増大する．この自由度を利用して局所計算可能性を小さくできる．複数の単項演算が定義された任意の集合Ｓに対し，２－強局所計算可能とするような符号化が構成できる．この結果から，順序機械のパイプライン化に関する一般的な結果が導ける．

［定理５］複数の単項演算が定義された任意の有限集合Ｓに対し，その演算の集合を２－強局所計算可能とするような冗長符号化が構成できる．

（証明）符号化には，並列プレフィックス計算（ＰＰＣ）を利用する[3][4]．以下に，単項演算を実現する順序回路の構成法を示す．

1)集合Ｓに対し，ＳからＳの中への自己写像全体の集合Ｍを考える．Ｓの単項演算は，Ｍの中の写像の１つに対応する．自己写像の合成を２つの写像に対する２項演算と見ると，Ｍはこの演算によって，モノイドをなす．Ｍの要素はn^n個であるから，その符号化には$n\log_2 n$ビットあればよい．モノイドの演算を計算する２入力論理素子のみで構成した組合せ回路（以下Ｉ回路とよぶ）の段数をDとすると，$D=O(n\log_2 n)$とできる．

2)次のような組合せ回路Ｃを考える．まず，単項演算OP_iから対応するＭの要素に変換する回路（Ｉｐ回路）をD個用意する．これらを葉として，高さ「$\log_2 D$」のＩ回路からなる２分木を作る．根の出力をさらに別のＩ回路（最終Ｉ回路という）の入力とし，その出力は自分自身のもうひとつの入力とする．Ｃは適当にダミー

の素子を入れて，入力から，最終Ｉ回路の出力までのすべての経路の段数を一致させておく（同期化）．

3)回路Ｃを構成するすべての論理素子の出力に記憶素子を挿入する（パイプライン化）．Ｃ中のすべての記憶素子は高々２つの他の記憶素子の出力にしか依存していない．

4)入力された演算OP_iを記憶する長さDのシフトレジスタを用意する．また，長さDの１ビットの巡回シフトレジスタを用意し，その中の一つのビットを基準ビットとする．巡回シフトレジスタには，１ヶ所のみ１をセットしておく．基準ビットが１のとき，入力シフトレジスタの内容が一斉にＩｐ回路の入力に送られるように回路を組む（パイプライン入力の同期）．

このようにして組んだ順序回路のすべての記憶素子（Ｃ中のものとシフトレジスタ中のもの）を符号の各桁に対応させる．各記憶素子（桁）は高々２個の記憶素子（桁）にしか依存しないので，２一局所計算可能となっている．また，この回路が集合Ｓ上の任意の単項演算の集合を計算することは容易に示される．□

定理５の証明の構成法では，最悪の場合nの指数オーダー以上の符号長となる．現実的には，与えられた単項演算の集合から必要となる最小限の自己写像だけを対象として，より短い符号長で実現できる．この定理から，次の系が導かれる．

［系］任意の順序機械は，入次数制限のある論理素子モデルのもとで，入力周期を素子１段分と記憶素子の遅延時間程度になるように１相のクロックのもとでパ

イプライン化できる[2].

6．応用と課題

今回は符号長についてはあまり議論しなかったが，明らかに符号長と局所計算可能性の間にはある種のトレードオフが存在すると考えられる．今後，このトレードオフについても考察を行ないたい．また，強局所計算可能性は，２項演算の弱局所計算可能性と密接な関係がある．この点についても考察していく．

謝辞　日頃御指導頂く本学田丸啓吉教授に感謝いたします．また，御討論頂いた京都産業大学岩間一雄助教授ならびに本学の矢島脩三教授はじめ諸氏に深謝いたします．本研究は一部文部省科学研究補助金62750321「局所計算可能な符号を用いた並列アルゴリズムの研究」による．

参考文献

(1)安浦，高木，矢島，”冗長符号化を利用した高速並列アルゴリズムについて”，信学論 vol.J70-D, no.3, March 1987.

(2)安浦寛人，”順序回路の局所計算可能な状態符号化について”，電子情報通信学会コンピュテーション研究会資料，COMP87-51, Nov. 1987.

(3)Unger.S.H.,"Tree Realization of Iterative Circuits," IEEE Trans. on Comput., vol.C-26, no.4, pp.365-383, April 1977.

(4)Ladner, R.E. and Fischer, M.J., "Parallel Prefix Computation," J. of ACM. vol 27, no.4, pp.831-838, Oct. 1980.