社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－MPS－54 (4)
2005／5／10

DNAを用いた乗算の方法の提案とその計算手順の設計

星野哲男† 涌井智寛†
下田明宏† 畠山正行††

要約：本論文では，DNAを用いた乗算の方法を構築／提案する．複数桁の乗算は1桁×1桁の乗算部分とそれらの結果の加算部分に分かれ，加算部分は既に実現されている．そこで本論文では，従来ほとんど扱われていなかった2より大きい基数の乗算を対象とし，1桁×1桁の計算を2桁に分けて計算するという方法で実現する．また，制限酵素によって，解と次桁への繰上りを分離できるように設計した．この方法によって$O(n^3)$の塩基配列の種類数と$O(n)$の実験操作で計算できるという結論を得た．また，DNAの長大化を防ぎ，配列設計の容易さから実現可能性が高いことも判明した．

A Proposal of Multiplication Method based on DNA Technology and A Design of its Computing Procedures

Tetsuo Hoshino† ,Tomohiro Wakui† ,Akihiro Shimoda†
and Masayuki Hatakeyama††

Abstract : In this paper, we have proposed and designed a multiplication method based on the DNA technology. In general, a multiplication with two (plural-digit) numbers result in (1) a multiplication with two (one-digit) number operation, and (2) additions of these multiplication results. The latter part (2) has already solved by several researchers. In thie paper, therefore, we aim at solving the former part (1) of the multiplication, and we have solved this problem (1) by separating two digits computing from the one-digit×one-digit computing. By developing this method, we have realized a new method to perform a multiplication by using $O(n^3)$ different kinds of DNA strands and $O(n)$ bio-steps. It can be estimated that our method realizes the lower errors since DNA strands can be prevented from becoming longer, and our method has higher realizability because of the simplicity of sequence design.

1. はじめに

1994年にAdlemanがDNAを用いた分子生物学的実験によって有向ハミルトンパス問題を解いて[1]以来，世界で様々なDNA計算の研究が行われている．それらはNP完全問題など特定の問題を解くための研究から，DNA計算以外の分野への応用を目指した多くの開発や研究が行われているが，四則演算や論理演算などをDNAを用いて解くという研究は前述の研究と比べると少ないのが現状である．

四則演算の研究について言えば，全加算器など加算の方法は提案されている[2)~5)]．乗算の方法もいくつかの研究例がある[3),6),7)]．DNAコンピュータを実現する時には算術演算などの基礎演算は必須なものである．その際，全加算器を使用して乗算を行うことは可能である[7)]が，DNAの実験操作に要する時間を考えると，乗算に特化したより効率的な方法を確立するのは有用であると思われる．

本論文では，DNAを用いた乗算の方法を構築・提案する．それは，次章で述べる従来の乗算方法の問題点を考慮した，任意基数が扱えて計算効率と実現可能性の高い方法である．

2. 従来のDNA乗算の方法の問題点

DNA全加算器を用いた乗算方法[5),7)]では乗算の加算への変形を外部の計算者自身が行わなければならず，これにはカウンタと連続的な加算の機能が必要だが，これはDNA全加算器では実現していない．

Barua らの方法[3)]とPelletier らの方法[6)]は2進数の乗算を実現する方法である．大きな数(大桁数)の乗

† 茨城大学大学院理工学研究科
Graduate School of Science and Engineering, Ibaraki University
†† 茨城大学工学部情報工学科
Department of Computer and Information Sciences, Ibaraki University

算の場合，Barua らの方法では DNA の長さが長大化し，Pelletier らの方法では用意する DNA タイルの種類数が膨大となり，ミスハイブリダイズなどのエラーが起こりやすくなる．

Barua らの方法は塩基配列が定まっており基数を上げるのは不可能である．Pelletier らの方法は基数を B とした場合，配列が B^4 種類以上必要となり現実的には難しい．しかし，配列の種類数を減らすことは，エラー率を下げ計算効率を上げることにつながり，DNA 乗算の実現可能性を上げることになるので重要である．

以上，従来の方法を考慮した上で本論文の目標をより高い実現可能性 (計算効率) を持つ DNA 乗算の方法を構築することに置く．そのために有効な方法は，数の基数を上げて乗算の桁数を減らすことである．

3. 新しい乗算方法の提案

3.1 新しい乗算方法構築の方針

一般に複数桁 × 複数桁の乗算は

(1) 1 桁 ×1 桁の乗算部分

(2) 全加算器を用いた加算部分

に帰着する．(2) については既に多くの方法が提案されている．(1) については基数が 2 の場合は 1 桁 ×1 桁の結果は 1 桁にしかならず，論理積を実現するだけで済むので，対処は容易である．しかし基数が 3 以上の場合は結果が 2 桁となってしまう．

以上から，(1) の 1 桁 ×1 桁の乗算部分をどのように高い実現可能性を持たせて構築するかが本論文で扱う解決すべき問題の焦点である．

算数の教科書では 10 進数の乗算は九九の表を用いて計算を行っている．そこで本論文では，1 桁 ×1 桁の乗算部分はこの九九の表に替わるものを DNA を用いて表現することにより乗算を実現する方法を採用する．つまり入力される全ての値に対して解候補を生成しておいて，実際の入力値に対応する DNA を投入して，必要な結果のみを取り出すという方法を考える．

本論文では，まず複数桁 ×1 桁の乗算を実現する方法を提案する．任意基数における 1 桁 ×1 桁の乗算が多くとも 2 桁にしかならないという特徴から，1 桁 ×1 桁の乗算結果 2 桁の生成を 1 桁目と 2 桁目に分けて考える．ここでは (第 1 項)×(第 2 項) の計算において第 1 項，第 2 項それぞれ 1 桁の値を計算するとする．

乗算結果の 1 桁目の計算は

i. “第 1 項の値” と “第 2 項の値” から乗算結果の 1 桁目の解候補の選択

ii. i. の結果と “前桁からの繰り上がり” から乗算結果の 1 桁目と，2 桁目への繰り上がりの決定

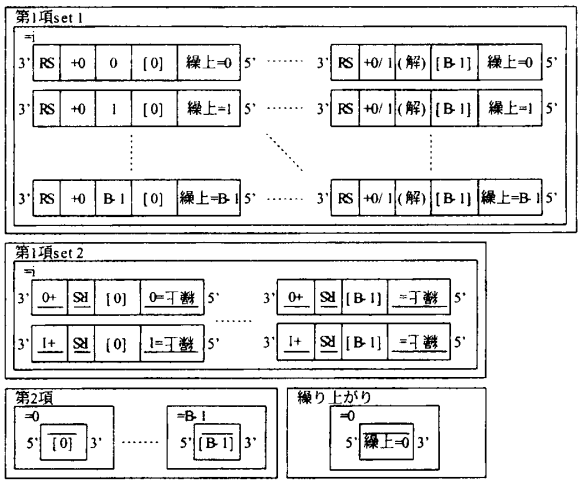

図 1 任意基数 (B) の乗算を実現する DNA の論理塩基配列 (ただし，第 1 項 = i のときのみ掲載)

Fig. 1 DNA logical sequences for multiplication of arbitary digit (B) (only for the first term = i)

という順に，2 桁目では

iii. “第 1 項の値” と “第 2 項の値” から乗算結果の 2 桁目の値の候補の選択

iv. ii. と iii. の結果から次桁への繰り上がりの決定

という順に計算することとする．

3.2 DNA 乗算の論理配列の設計

3.1 節の解の生成手順に合わせて，乗算を実現する DNA の論理塩基配列を設計する．設計した論理塩基配列を図 **1** に示す．図中において “RS” は Recognition Site の略で制限酵素の認識部位を表し，“AGTC” の相補関係にある論理配列を “$\overline{\text{AGTC}}$” と表記し，“繰り上がり” を “繰上” と省略する．“(解)” と “+0/1” はそれぞれ ‘(第 1 項)×(第 2 項) の 1 桁目+前桁からの繰り上がり’ の 1 桁目と 2 桁目の値，“繰上=” は ‘(第 1 項)×(第 2 項) の 2 桁目+1 桁目からの繰り上がり’ の値となる．

論理配列の設計の方針としては

1) 第 1 項に九九の部分と繰り上がりの部分を計算させ，第 2 項に九九の部分の決定をさせる，

2) 第 1 項を 1 桁目と 2 桁目を分け，第 2 項は各桁共通に使用する，

3) 繰り上がりの部分を切断可能にする，

という 3 点を考慮して行った．そのため第 1 項の 1 桁目と 2 桁目の DNA 群を “第 1 項 set1” と “1 桁目 set2” と名付けた．“第 1 項 set1” は “第 2 項の値” と “前桁からの繰り上がり” から “その桁の乗算結果” と “2 桁目への繰り上がり” を計算するように，“第 1 項 set2” は “1 桁目からの繰り上がり” と “第 2 項の値” から “次桁への繰り上がり” を計算するように論理配

列を決定した．“第 2 項” と “繰り上がり” はそれ自身の値を表すものとした．また，3) を実現させるために図中の位置に “認識部位 (RS)” を挿入した．

第 1 項 set2 で 3 つの配列が上下逆となっているのは，第 1 項 set2 の “1 桁目からの繰り上がり” と “次桁への繰り上がり” がそれぞれ第 1 項 set1 の “2 桁目への繰り上がり” と “前桁からの繰り上がり” とハイブリダイズさせるためである．

演算は各桁毎に独立しているので必要に応じて任意の時点で切断できる．ただし切断は各桁毎の必須作業ではなく，必要に応じて実行する

この論理塩基配列は，論理塩基配列内に桁情報を組み入れずに，各桁の乗算は上下の桁とは独立に計算させ，他桁の計算のために投入した塩基配列と混じり合うことのない計算方法であるため，多桁の計算を行う際にもそのまま使用する．ただし，複数桁 (2 桁以上) を図 1 の論理塩基配列で計算を行うと，認識部位 (RS) が桁毎に現れ，切断を行う際に解を含む DNA が桁毎にバラバラになってしまう．そのため，任意に切断を行えるようにするためには図 1 と同様であるが認識部位を持たない配列群が必要となる．

3.3 DNA 乗算の計算手順の設計

設計した計算手順を説明する．

段階 **1**：(第 1 項)×(第 2 項) の解候補の生成．
第 1 項 set1 と第 2 項を表す DNA 群を試験管へ投入してハイブリダイズさせる．その後，ハイブリダイズした分子を抽出する．

段階 **2**：解と 2 桁目への繰り上がりの生成．
繰り上がりを表す DNA 群を試験管へ投入し，段階 1 で得られた DNA とハイブリダイズさせる．その後，ハイブリダイズした分子を抽出する．

段階 **3**：(第 1 項)×(第 2 項) の次桁への繰り上がりの候補の生成．

第 1 項 set2 と第 2 項を表す DNA 群を試験管へ投入してハイブリダイズさせる．その後，ハイブリダイズした分子を抽出する．

段階 **4**：次桁への繰り上がりの生成．
段階 2 で得られた DNA と段階 3 で得られた DNA を同一試験管に投入してハイブリダイズさせる．その後，ハイブリダイズした分子を抽出する．

段階 **5**：解と次桁への繰り上がりの切断・抽出．
必要に応じて次桁への繰り上がりを表す部分とそれまでの桁の乗算結果を表す部分とを認識部位の位置で切断し，抽出する．

段階 5 で得られる DNA には次桁への繰り上がりが符号化されており，特別な加工なしで次桁の計算に “繰り上がり” として使用することができる．したがって，段階 1 から段階 5 の手順を繰り返すことで任意桁の乗算が可能となる．ただし最下位桁の最初の計算を行う段階 1 では，前桁からの繰り上がりがないため “繰り上がり=0” を与える．

被乗数の桁数回この計算手順を繰り返した後，段階 1 に戻り第 1 項=0 として段階 2 まで計算を行って計算終了となる．

3.4 複数桁 × 複数桁の計算

複数桁 × 複数桁の乗算は，3.1 節でも述べたように，各桁の乗算結果を全加算器を用いて加算することで実現する．乗数が n 桁目 ($n \geq 2$) の場合，$n-1$ 回だけ “第 1 項=0” または “第 2 項=0” の乗算を行ってから被乗数との乗算を行うものとする．

4. DNA の配列種類数を減らす方法

3.2 節で設計した論理配列は，2 進数で 19 種類，B 進数で $B(B+1)^2+1$ 種類も必要となる．これは 3 種類の入力値におけるすべての解候補を生成する方法で実現しているからであり，これを減らすことは DNA 乗算の実現可能性を大きく向上させることにつながる．

4.1 $O(n^2)$ への改良方針の提案

3.1 節で述べた複数桁 ×1 桁の乗算方法の構成を以下の 2 つに分解する．

[1] 1 桁 ×1 桁の乗算部分，つまり九九の部分
[2] 繰り上がりとの加算部分

[1] の九九の部分は，前述の乗算方法と同様に ‘1 桁目’ と ‘2 桁目’ に分けて計算を行う．必要とする配列種類数は “第 1 項” と “第 2 項” の値にのみ依存し $2B^2$ である．この DNA から乗算結果の ‘1 桁目’ と ‘2 桁目’ をそれぞれ生成する．[2] の加算部分も [1] 同様に ‘1 桁目’ と ‘2 桁目’ に分ける．‘1 桁目’ では “前桁からの繰り上がり” と，[1] から受け取った “1 桁目の解” を加算し，“その桁の乗算結果” と “2 桁目への繰り上がり” を生成するように設計する．‘2 桁目’ では “1 桁目からの繰り上がり” と，[1] から受け取った “2 桁目の解” を加算し，“次桁への繰り上がり” を生成する．よって配列種類数はそれぞれ 2B 種類となる．

この構成を本節までに提案した方法への適用を考えたとき，“第 1 項 set1” を改良すれば $O(n^2)$ の配列種類数で乗算が実現可能となる．そこで “第 1 項 set1” を “set1.1” と “set1.2” に分け，“set1.1” は [1]，“set1.2” は [2] の ‘1 桁目’ を実現するように再構成する．これを実施することで $O(n^2)$ が実現する．ただしこの方法では，実験操作や論理配列がより複雑化するという難点を抱えている．

4.2 容易に実施可能な方法

容易に実施できる2つの方法を考案した.

(1) 常に (第1項)≥(第2項) となるような乗算を行う方法を採る.

(2) 実際計算では使用されない論理塩基配列を乗算を実現する配列から削除する.

この2つの方法を適用すると配列種類数は約半数近く削除することができる.

(2) の方法は単純に計算で使用されない不要なDNA配列を取り除くことで実現できる. しかし (1) の方法は, 外部で計算者が "第1項" と "第2項" の値の比較を行わなければならないという問題がある.

5. 考　　察

[乗算を実現する方法の比較] Barua らの方法は, 乗算を加算とシフト演算で実現しているという点から見れば涌井の方法と近い. Pelletier らの方法は1桁 ×1 桁の乗算の解候補を全て生成し, 複数桁 × 複数桁の乗算を一度に計算を行うという方法で実現している.

我々の方法は複数桁 ×1 桁の乗算を2段階に分けて計算を行う方法で実現し, 全加算器を用いて乗算結果の加算を行う. そのため解候補は "第1項", "第2項", "前桁からの繰り上がり" の3要素から生成されるが, Pelletier らの方法は "他桁の乗算結果" という要素が加わる. また, この解候補の DNA は我々は線形の1次元配列で実現しているが, Pelletier らは DNA タイルを用いて2次元的に実現しているため論理塩基配列が複雑な構造をしている.

[配列種類数及び実験工程数の比較] 本論文で提案する乗算方法は, 複数桁 × 複数桁の乗算は前述の論理配列とは異なる配列の全加算器を利用する必要がある. この点では同一の塩基配列で加算も乗算も行える Barua らの方法は, 我々の方法よりも優れている.

しかし大桁数の乗算を行う場合, Barua らの方法では DNA 長大化してしまう. 我々の方法は必要に応じて DNA を切断できるため DNA の長大化を防ぐことができ, Barua らの方法よりミスハイブリダイズなどエラーが起こりにくいと考えられる.

計算効率を上げるには基数を上げて桁数を減らす方法が考えられる. Barua らの方法は基数を上げることは難しく, Pelletier らの方法では配列種類数が $O(n^4)$ となる. 我々の方法は $O(n^3)$ で実現可能であり, この点においては2つの方法より優れているといえる.

さらに, Pelletier らの方法はすべての DNA を一度に同一試験管に投入し反応させるが, 我々の方法では同一試験管に投入する配列種類数が $O(n^2)$ に限定されるため, ミスハイブリダイズなどのエラーが Pelletier らの方法より起こりにくいと考えられる. また, 我々の方法は複雑な二次元タイルではなく一次元配列で実現した. そのため配列設計が容易であり, Pelletier らの方法より実現可能性が高いといえる.

[比較考察のまとめ] 以上より, 本論文で提案する方法は従来の乗算方法と異なる乗算方法で乗算を実現し, さらに計算効率がよく実現可能性が高い方法であると言えよう.

6. 結論と今後の課題

本論文では,

- 乗算を加算の形にせず,
- 桁数の増加に伴う DNA の長大化を防止可能で,
- 任意の基数に拡張でき,
- 複数桁 ×1 桁の乗算を実現する

DNA を用いた乗算の方法を提案した. また, この方法は $O(n^3)$ の配列種類数で $O(n)$ の実験工程で実現可能であることがわかった.

今後の課題としては, 配列種類数を $O(n^2)$ に減らす方法にできるように改良/再構築することと, 本論文で提案する乗算方法の妥当性および実現可能性の検証をすることである.

参　考　文　献

1) Adleman, L. M.: Molecular Computation of Solutions to Combinational Problems, *Science*, Vol. 266, pp. 1021–1024 (1994).
2) Guarnieri, F., Fliss, M. and Bancroft, C.: Making DNA Add, *Science*, Vol. 273, No. 5272, pp. 200–223 (1996).
3) Barua, R. and Misra, J.: Binary Arithmetic for DNA Computers, *DNA Computing: 8th International Workshop on DNA Based Computers*, pp. 124–132 (2002).
4) Fujiwara, A., Matsumoto, K. and Chen, W.: Addressable procedures for logic and arithmetic operations with DNA strands, *Proceedings of Workshop on Advances in Parallel and Distributed Computational Models (IPDPS 2003)* (2003).
5) 涌井智寛: DNA を用いた全加算器の提案とその設計/検証, 修士論文, 茨城大学 (2005).
6) Pelletier, O. and Weimerskirch, A.: Algorithmic Self-assembly Of DNA Tiles And Its Application To Cryptanalysis, *GECCO 2002*, pp. 139–146 (2002).
7) 涌井智寛: DNA 四則演算 (2003). 茨城大学工学部情報工学科 平成14年度卒業論文.