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

2005－MPS－53 (10)
2004／3／10

ビットパラレル手法によるアライメントアルゴリズム

馬場謙介, 于雲青, 村上和彰
九州大学大学院システム情報科学研究院
〒816-8580 福岡県春日市春日公園6-1

概要 近似文字列照合問題は，二つの文字列と閾値が与えられて，片方の文字列の任意の部分文字列に対してもう一方の文字列との編集距離を計算し，与えられた閾値以下となるものを見つける問題である．Myersは，ビットパラレル手法により，近似文字列照合問題を高速に解くアルゴリズムを提案した．しかし，出力として編集距離ではなくアライメントを求める場合，単純な方法を適用できず，高速化が効果的でない場合がある．本論文では，近似文字列照合問題に対するアライメントを，ビットパラレル手法によって求めるアルゴリズムを提案する．

An Alignment Algorithm by Bit-Parallelism

Kensuke Baba, Yunqing Yu, and Kazuaki Murakami
Department of Informatics, Kyushu University
Kasugakoen 6-1, Kasuga-City, Fukuoka 816-8580, Japan

Abstract Approximate matching problem is, given two strings and a parameter, to find all substring of a string whose edit distance with the other string is at most the parameter. Myers introduced an efficient algorithm for approximate matching problem based on bit-parallelism. However, if we need alignment as the answer of the problem, the straightforward method can't be applied. In this paper, an alignment algorithm based on bit-parallelism for approximate matching problem is presented.

1 序論

ヒトゲノムの解読をはじめとして，分子生物学が対象とする膨大な生命情報の理解を，文字列の解析というアプローチによって深めようとする研究が世界中で行われている．その中でも重要な概念は，二つの文字列が似ているかどうかということである．一致する文字の多い部分等の文字列間の類似性を見つけることは，相同性検索（ホモロジーサーチ）と呼ばれ，例えば，遺伝子の構造を予測したり，進化による関係を解析するために用いられる．また，分子生物学の例を挙げずとも，コンピュータによる大規模なデータの処理が社会基盤となった現代では，文字列の高度な解析手法を開発することが必要不可欠である．

二つの文字列が似ているかどうかを考えるための指標として，文字列間の類似度，あるいは距離を定義することが求められる場合がある．もっとも単純なものは，「片方の文字列からもう一方の文字列を得るために，文字の置換や挿入，削除を最低何回行えばよいか」を表す編集距離 [3] と呼ばれる値である．この距離は，編集の種類や，文字の種類について重み付けを行うことによって，より一般的なものになる．この値を考える際，「片方の文字列中のある文字を，もう一方の文字列中のどの文字と比較するか」を表すのがアライメント [1] である．編集距離を求める計算は，考え得る全てのアライメントに対応する編集操作の数のうち，最適なものを見つけるものであり，対象となる文字列が非常に長い場合，単純なアルゴリズムでは現実的な時間で計算することができない．一般に，この計算には動的計画法が用いられる．対象となる二つの文字列のそれぞれの接頭辞についての編集距離を各要素とする行列（DP 行列）を再帰的に求めることで，計算の重複を避けており，長さ m と n の文字列間の編集距離を $O(mn)$ 時間で計算することができる [5]．

近似文字列照合問題は，二つの文字列間の編集距離を求める問題をより一般的にしたもので，片方の文字列の任意の部分文字列に対してもう一方の文字列との編集距離を計算し，与えられた閾値以下となるものを見つける問題である．Myers [4] は，近似文字列照合問題を解くために，ビットパラレル手法により DP 行列を計算するアルゴリズムを提案した．このアルゴリズムの計算時間は，w をコンピュータのワード長として，$O(\lceil m/w \rceil n)$ 時間で抑えられる．ここで用いられているビットパラレル手法は，文字列間の類似度を二値の論理式の演算によって表すことで，DP 行列の成

分のうち最大 w 個の値を並行して計算するものである．出力として編集距離ではなくアライメントを求める場合，単純には，DP 行列から $O(m+n)$ 時間で計算が可能である．しかし，ビットパラレル手法においては，DP 行列の要素うち値を求めないものがあるため，単純な方法を適用できない．アライメントを求めるための計算を加えた場合，高速化が効果的でない場合がある．

本論文では，近似文字列照合問題に対するアライメントをビットパラレル手法によって求める高速なアルゴリズムを提案する．同様のアルゴリズムとして，Hyyrö [2] は，編集操作によって異なる三種類の編集距離について，ビットパラレル手法によるアライメントアルゴリズムを提案している．しかし，近似文字列照合問題の場合，このアルゴリズムを直接適用することはできない．さらに，一つの編集距離に対応するアライメントが複数存在するのに対し，このアルゴリズムによって求められるアライメントがその中のどのようなものかについての詳しい考察や，アルゴリズムの正しさについての証明はなされていない．我々は，近似出現に対するアライメントについて正規化を行うことで，アルゴリズムによって求められるアライメントを厳密に表現し，アルゴリズムの正統性を証明する．

2 準備

Σ を文字の有限集合とし，これを**アルファベット**と呼ぶ．Σ 上の文字列全体からなる集合を Σ^* で表し，空文字列を ε で表す．また，$\Sigma^+ = \Sigma^* - \{\varepsilon\}$ の表記を用いる．

ある文字列 s について，s の長さを $|s|$ で表し，s の i 番目の要素を s_i で表す．ただし，$1 \leq i \leq |s|$ である．文字列 $s_i s_{i+1} \cdots s_j$ を s の**部分文字列**と呼び，$s_{i,j}$ で表す．特に，$i = 1$ のとき**接頭辞**，$j = |s|$ のとき**接尾辞**と呼ぶ．便宜上，$j < i$ のとき $s_{i,j} = \varepsilon$ とする．また，$s^{-1} = s_{|s|} s_{|s|-1} \cdots s_1$ とする．

二つの文字列間の**編集距離**とは，片方の文字列から他方の文字列を得るために必要な編集操作のうち最小の回数である．ただし，編集操作としては文字の「挿入」,「削除」及び「置換」を許すものとする．二つの文字列 $p, t \in \Sigma^+$ の間の編集距離を $d(p,t)$ で表す．

二つの文字列 $p, t \in \Sigma^+$ 及び，ある閾値 $\delta < |p|$ が与えられて，t の部分文字列 t' が p の**近似出現**であるとは，$d(p,t') \leq \delta$ であることである．**近似文字列照合問題**とは，テキスト t 中のパターン p の近似出現を全て見つける問題である．

編集転写とはアルファベット $\{I, D, R, M\}$ 上の文字列であり，各文字は，それぞれ，三つの編集操作と「一致」を表している．つまり，ある文字列に対する編集転写は，対応する編集操作を順番に行うことで，その文字列を他の文字列に転写するものである．

3 近似出現に対するアライメント

本節では，アライメントを編集転写によって表し，近似文字列照合問題に対するアライメントを定義する．

近似文字列照合問題に対するアライメントは自明ではない．その理由のひとつは，ある近似出現に対応する編集転写が複数存在し得ることである．編集転写の正規形を以下のように定義する．

定義 1 文字列 $p \in \Sigma^+$ から $t \in \Sigma^+$ への正規編集転写とは，p と t の間の編集距離に対応する編集転写のうち，$I < R < D < M$ の順による辞書式順序で最大のものである．

近似文字列照合問題に対するアライメントが自明でないもう一つの理由は，近似文字列照合問題の解となる近似出現が複数あり得ることである．

定義 2 ある位置についての代表近似出現とは，その位置から始まる近似出現のうち，対応する編集距離が最も小さく，長さが最も短いものである．

補題 1 ある位置についての代表近似出現に対する正規編集転写は，その位置から始まり，対応する編集距離が最小となる近似出現に対する正規編集転写のうち，$I < R < D < M$ の順による辞書式順序で最大のものである．

本論文では，近似文字列照合問題に対するアライメントを次のように定義する．

定義 3 近似文字列照合問題に対するアライメントアルゴリズムとは，文字列 $p, t \in \Sigma^+$ と閾値 δ が与えられて，t 中の p の全ての代表近似出現の正規編集転写を返すものである．

4 アルゴリズム

4.1 動的計画法に基づく単純なアルゴリズム

長さ m と n の二つの文字列についての編集距離は，動的計画法により $O(mn)$ 時間で求めることができる．このアルゴリズムでは，**DP 行列**と呼ばれる行列が用いられる．これは，p, t を二つの文字列とするとき，以下の (i,j) 成分を持つ $(m+1) \times (n+1)$ 行列として定義される．

$$C[i,j] = \min\{C[i-1,j-1] + W(p_i, t_j), C[i-1,j] + 1, C[i,j-1] + 1\}$$

ただし，$i = 0$ のとき $C[0,j] = jd$，$j = 0$ のとき $C[i,0] = id$ とする．また，$W(a,b)$ は $a = b$ のとき 0，$a \neq b$ のとき 1 をとるものとする．このとき，τ を文字列 $p_{1,i}$ から $t_{1,j}$ への編集距離に対応する編集転写とすると，

- ρ_1 を $p_{1,i-1}$ から $t_{1,j-1}$ への編集距離に対応する編集転写として，$\tau = \rho_1 M$ または $\tau = \rho_1 R$，

- ρ_2 を $p_{1,i-1}$ から $t_{1,j}$ への編集距離に対応する編集転写として，$\tau = \rho_2 D$，または，
- ρ_3 を $p_{1,i}$ から $t_{1,j-1}$ への編集距離に対応する編集転写として，$\tau = \rho_3 I$

のいずれかが成り立つので，$1 \leq i \leq m$ と $1 \leq j \leq n$ について，$C[i,j]$ は $p_{1,i}$ と $t_{1,j}$ の間の編集距離であることがわかる．

このアルゴリズムは，DP 行列の初期条件を，$i = 0$ のとき $C[0,j] = 0$，$j = 0$ のとき $C[i,0] = id$ とすることで，近似文字列照合問題へ拡張することができる．つまり，$C[i,j]$ は $p_{1,i}$ と，位置 j で終わる t の部分文字列との間の編集距離のうち最も小さいものであるから，以下が明らかである．

補題 2 $C[i,j]$ は，t^{-1} 中の位置 $n-j+1$ での p^{-1} の代表近似出現に対応する編集距離である．

アライメントを求める場合，DP 行列を計算した後，各成分が最大値としてどの成分からの値をとったかを辿ることで，対応する編集転写を求めることができる．一般に，DP 行列からアライメントを求める作業は**トレースバック**と呼ばれる．このアルゴリズムにおいて，DP 行列を求めるには，$(m+1) \times (n+1)$ 個の成分について最大値を求める計算を行うので，$O(mn)$ 時間が必要である．また，トレースバックに要する時間は単純な方法でも $O(m+n)$ 時間であるから，$O(mn)$ 時間で解くことができる．

4.2 ビットパラレル手法によるアルゴリズム

Myers [4] は，近似文字列問題を解くために，ビットパラレル手法により DP 行列を計算するアルゴリズムを提案した．アルゴリズムの概要を図 1 に示す．詳しい説明は省略するが，このアルゴリズムで用いられている表記を列挙する．

$0 \leq i \leq m$ と $1 \leq j \leq n$ に対して，

$$\Delta h[i,j] = C[i,j] - C[i,j-1]$$

と表す．また，$1 \leq i \leq m$ と $0 \leq j \leq n$ に対して，

$$\Delta v[i,j] = C[i,j] - C[i-1,j]$$

と表す．また，

$$\begin{aligned} Eq[i,j] &= \iota(p_i = t_j) \\ Pv_j[i] &= \iota(\Delta v[i,j] = 1) \\ Mv_j[i] &= \iota(\Delta v[i,j] = -1) \end{aligned}$$

である．ただし，ι は引数の式が真のときに 1，偽のときに 0 を返す関数である．

DP 行列の隣り合う 4 つの成分 $C[i-1,j-1], C[i,j-1], C[i-1,j], C[i,j]$ を一つのブロックとして考えると，二つの Δv が存在するが，それらを $\Delta v_{out} = \Delta v[i,j], \Delta v_{in} = \Delta v[i,j-1]$ とする．また，DP 行列の行についても列と同様の概念を用いて，$\Delta h_{out} = \Delta h[i,j], \Delta h_{in} = \Delta h[i-1,j]$ とする．このとき，

$$\begin{aligned} \Delta v_{out} &= \min\{-Eq, \Delta v_{in}, \Delta h_{in}\} + (1 - \Delta h_{in}) \\ \Delta h_{out} &= \min\{-Eq, \Delta v_{in}, \Delta h_{in}\} + (1 - \Delta v_{in}) \end{aligned}$$

である．ここで，

$$Xv = Eq \vee \iota(\Delta v_{in} = -1), \quad Xh = Eq \vee \iota(\Delta h_{in} = -1)$$

とすると，次が成り立つ．

$$Pv_{out} = Mh_{in} \vee \neg(Xv \vee Ph_{in}) \qquad Mv_{out} = Ph_{in} \wedge Xv$$
$$Ph_{out} = Mv_{in} \vee \neg(Xh \vee Pv_{in}) \qquad Mh_{out} = Pv_{in} \wedge Xh$$

また，(Pv_{j-1}, Mv_{j-1}) と t_j から $(Pv_j[i], Mv_j[i])$ を計算するために，p_i についての表

$$Peq(s)[i] = \iota(s = p_i)$$

を作成しておく．ただし，$s \in \Sigma$ である．

このアルゴリズムでは，w をワード長とすると，各列についての二つの段階からなる計算は $O(\lceil m/w \rceil)$ 時間で行われるので，$1 \leq j \leq n$ について $C[m,j]$ を求めるには $O(\lceil m/w \rceil n)$ 時間が必要である．また，前処理として行列 Eq の値を求めなければならないが，$Peq(s)$ を導入することで $O(|\Sigma| m)$ 時間で計算できる．

Procedure Bit-Parallel Algorithm
inputs: $p = p_1 \cdots p_m \in \Sigma^+$, $t = t_1 \cdots t_n \in \Sigma^+$, Σ ;
outputs: $C[m,0], \ldots, C[m,n]$;
for $i = 1, \ldots, m$ **do** {
  for $s \in \Sigma$ **do** {
    if $(p_i = s)$ $Peq(s)[i] = 1$ **else** $Peq(s)[i] = 0$;
  }
}
$P_v = 1^m$; $M_v = 0$; $C[m,0] = m$;
for $j = 1, \ldots, n$ **do** {
  $Eq = Peq[t_j]$;
  $Xv = Eq \mid Mv$;
  $Xh = (((Eq \;\&\; Pv) + Pv) \hat{} Pv) \mid Eq$;
  $Ph = Mv \mid \sim (Xh \mid Pv)$;
  $Mh = Pv \;\&\; Xh$;
  if $(Ph \;\&\; 10^{m-1})$ $C[m,j] = C[m-1,j] + 1$;
  else if $(Mh \;\&\; 10^{m-1})$ $C[m,j] = C[m-1,j] - 1$;
  $Ph \ll 1$;
  $Pv = (Mh \ll 1) \mid \sim (Xv \mid Ph)$;
  $Mv = Ph \;\&\; Xv$;
}

図 1: ビットパラレルによる近似文字列照合アルゴリズム

4.3 アライメントアルゴリズムへの拡張

前節での議論からもわかるとおり，アライメントとしての編集転写は DP 行列から自明に求められるわけではない．

4.1 小節で用いた単純な方法では，トレースバックに要する計算時間は $O(m+n)$ であり，ビットパラレル手法による高速化の効果がなくなってしまう場合がある．ここでは，前小節の Myers のアルゴリズムをアライメントアルゴリズムへ拡張する．

我々は，編集転写の候補が，前小節のアルゴリズムにおいて用いられた行列 Eq, Pv, Mv によって制限されることに注目した．これによって，前小節のアルゴリズムから，新たな計算を行うことなく編集転写を求めることができる．

補題 3 $\tau, \rho_1, \rho_2, \rho_3$ を，それぞれ，文字列 $p_{1,i}$ から $t_{1,j}$，$p_{1,i-1}$ から $t_{1,j-1}$，$p_{1,i-1}$ から $t_{1,j}$，$p_{1,i}$ から $t_{1,j-1}$ への編集距離に対応する編集転写とする．このとき，以下が成り立つ．

(1) $Eq[i,j]=1$ ならば，$\tau=\rho_1 M$ である．

(2) $Eq[i,j]=0$ かつ $Pv_j[i]=1$ ならば，$\tau=\rho_2 D$ かつ $\tau \neq \rho_1 M$ である．

(3) $Eq[i,j]=0$ かつ $Pv_j[i] \neq 1$ かつ $Mv_{j-1}[i] \neq 1$ ならば，$\tau=\rho_1 R$ かつ $\tau \neq \rho_1 M$ かつ $\tau \neq \rho_2 D$ である．

(4) $Eq[i,j]=0$ かつ $Pv_j[i] \neq 1$ かつ $Mv_{j-1}[i]=1$ ならば，$\tau=\rho_3 I$ かつ $\tau \neq \rho_1 M$ かつ $\tau \neq \rho_2 D$ かつ $\tau \neq \rho_1 R$ である．

本論文で提案するアルゴリズムは以下の通りである．まず，前小節のアルゴリズムによって DP 行列の最後の行 $C[m,j]$ $(0 \leq j \leq n)$ を求める．この際，途中の計算で用いられる Pv と Mv を保持しておく．また，$i=m$ のとき $C[i,j] \leq \delta$ となる全ての j について，以下の手順を再帰的に繰り返してトレースバックを行う．

1 $Eq[i,j]=1$ のとき，$i=i-1, j=j-1$ とし，$\tau=\tau M$ とする

2 それ以外のとき，

2-1 $Pv_j[i]=1$ のとき，$i=i-1$ とし，$\tau=\tau D$ とする

2-2 それ以外のとき，

2-2-1 $Mv_{j-1}[i] \neq 1$ のとき，$i=i-1, j=j-1$ とし，$\tau=\tau R$ とする

2-2-2 それ以外のとき，$j=j-1$ とし，$\tau=\tau I$ とする

これによりアライメントへ拡張されたアルゴリズムの概要を図 2 に示す．

定理 1 *Bit-Parallel Algnment Algorithm* は近似文字列照合問題に対するアライメントアルゴリズムである．

証明 補題 3 から，トレースバックのための手順によって選ばれる編集転写が正規形であることがわかる．また，補題 2 および補題 1 から，このアルゴリズムによって得られる出力は，代表近似出現に対する編集転写であることがわかる．□

```
Procedure Bit-Parallel Alignment Algorithm
inputs: p = p_1 ··· p_m ∈ Σ^+, t = t_1 ··· t_n ∈ Σ^+, δ, Σ
;
outputs: τ_1, ..., τ_k ∈ {I, D, R, M}^+ ;
Compute C[m,0], ..., C[m,n] by Bit-Parallel Algorithm ;
i = m ; k = 1 ;
for j s.t. C[i,j] ≤ δ do {
  τ = ε ;
  while (i, j ≥ 1) {
    if (Eq[i,j] = 1) {
      i = i − 1 ; j = j − 1 ; τ = τM
    } else if (Pv_j[i] = 1) {
      i = i − 1 ; τ = τD ;
    } else if (Mv_{j−1}[i] ≠ 1) {
      i = i − 1 ; j = j − 1 ; τ = τR ;
    } else {
      j = j − 1 ; τ = τI ;
    }
  }
  τ_k = τ^{−1} ; k = k + 1 ;
}
```

図 2: ビットパラレルによるアライメントアルゴリズム

5 結論

近似文字列照合問題に対するアライメントをビットパラレル手法によって求める高速なアルゴリズムを提案した．近似出現とアライメントについて正規化を行い，アルゴリズムによってその正規形が求められることを証明した．

参考文献

[1] Gusfield, D., "Algorithms on Strings, Trees, and Sequences", Cambridge University Press, 1997.

[2] Hyyrö, H., "A Note on Bit-Parallel Alignment Computation", Proc. the Prague Stringology Conference '04 (PSC 2004), 2004.

[3] Masek, W. J. and Paterson, M. S., "A Faster Algorithm for Computing String Edit Distance", J. Comput. Syst. Sci., vol.20, pp.18–31, 1980.

[4] Myers, G., "A Fast Bit-Vector Algorithm for Approximate String Matching Based on Dynamic Programming", J. ACM, vol.46, No.3, pp.395–415, 1999.

[5] Needleman, S. B. and Wunsch, C. D., "A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins", J. Mol. Biol., vol.48, pp.443–453, 1970.