6N-09

DAWG を用いたダブル配列による全文検索手法

市橋 良晃[†1] 泓田 正雄[†2] 三戸 太郎[†3] 森田 和宏[†2] 青江 順一[†2]
[†1] 徳島大学 大学院先端技術科学教育部
[†2†3] 徳島大学 大学院ソシオテクノサイエンス研究部

1. はじめに

現在，蓄積されるテキストデータ量の増大により，高速な検索や高度な検索が求められている．検索に関する研究には検索手法とデータ構造の二面が存在するが，検索手法の 1 つとして高速性・機能性に優れたトライがある．トライは登録キー集合の各キーの共通接頭辞を併合して作られる木構造であり，検索キーを 1 文字単位で比較することにより検索を行うことができる．検索のためのトライを実現するデータ構造としては，検索の高速性と記憶量のコンパクト性を併せ持つダブル配列法[1]がある．ダブル配列は，2 つの 1 次元配列によってトライのノード間遷移を実現するデータ構造である．現在までに，圧縮ダブル配列などの記憶量を圧縮する手法が提案されている．

だが，ゲノムデータなどの長文中のパターン検索をはじめとする，全文検索を目的としたダブル配列の構築手法は提案されていない．そこで，本論文では Suffix CDAWG を用いてダブル配列を構築し, 全文検索する手法を提案する．

2. ダブル配列

青江によって提案されたダブル配列は，$BASE$と$CHECK$という 2 つの 1 次元配列を用いてトライのノード間遷移を実現する．トライのノード番号は，ダブル配列の要素のインデックスと対応する．以降では，ノード番号sはダブル配列のs番目の要素を表す．トライにおいて，ノード番号sからノード番号tへ，遷移文字cによる遷移が定義されている場合，ダブル配列は次式を満足する．

$$t = BASE[s]+CODE[c]$$
$$\boldsymbol{CHECK[t]} = s \quad (1)$$

$BASE$と$CHECK$は遷移番号を表現する．また，トライでは，分岐せずに遷移が続く部分を Separate String と呼ぶ．ダブル配列では Separate String を TAIL に登録することで，記憶容量削減と検索時間の短縮を図っている．

キー集合K={“aac”, “ac”, ”gat”, “gaac” , “gt”}に対するダブル配列を図 1 に示す．ただし，”#”は文字列の終端を示す．さらに，$BASE$値が負の値の場合は TAIL への遷移を表す．TAIL では，共通部分文字列へ集約することで，記憶容量を節約することができる. たとえば図 1 では，ノード番号 4, 7 から遷移する最長の”c#”のみを記憶し，ノード番号 5 や 10 などは途中部分を指して記憶することで, 重複した TAIL を別々に登録する必要はなくなる．

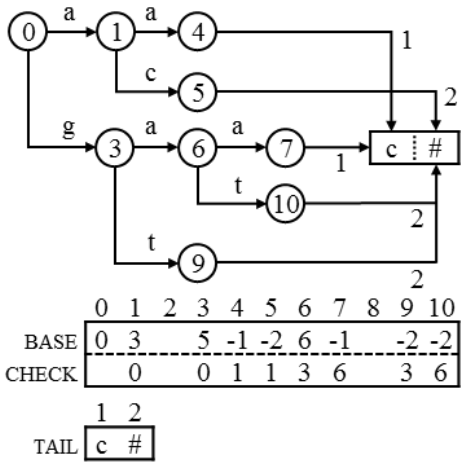

図 1：ダブル配列(キー集合K)

また，ダブル配列で DAWG を実現する手法[2]も提案されている．DAWG とは，共通文字列部分を併合したデータ構造である．従来のトライを用いるダブル配列に比べて，DAWG を用いるダブル配列は検索性能を維持しつつ，メモリ使用量を削減できる．

3. 全文検索

Suffix Tree は，全文検索に用いられるデータ構造であり，ある有限文字列のすべての接尾辞を木構造で表現する．長文中のパターン検索などを高速におこなえるが，記憶容量は大きくなる．また，Suffix DAWG とは，Suffix Tree を DAWG により表現したデータ構造であり，決定性有限オートマトンに属する[3]．有限文字列w=”gtagtaaac”に対する Suffix DAWG を図 2 に示す．

4. 提案手法

Compact Directed Acyclic Word Graphs（CDAWG）は，分岐のない部分文字列が 1 つの遷移に集約されるという特徴を持つデータ構造である[4]．さらに，空間効率にも優れて

A Double Array Method for Full-text Search by DAWG.
†1Yoshiaki Ichihashi. Department of Advanced Technology and Science. University of Tokushima.
†2Masao Fuketa, Kazuhiro Morita and Jun-ichi Aoe. Department of Information Science and Intelligent Systems, University of Tokushima.
†3Taro Mito, Department of Life Systems, Institute of Technology and Science, The University of Tokushima.

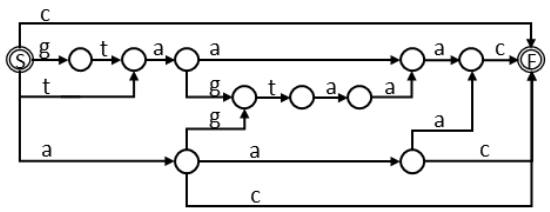

図 2：Suffix DAWG(gtagtaaac)

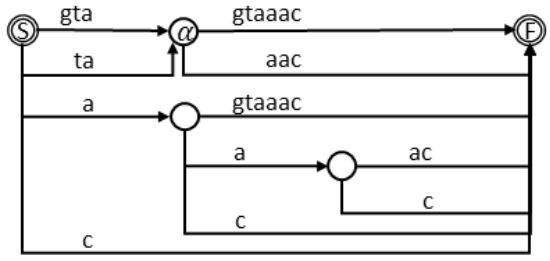

図 3：Suffix CDAWG(gtagtaaac)

いる．ここで，Suffix Tree を CDAWG で表現することが可能と考え，このデータ構造を Suffix CDAWG と定義する．有限文字列w=”gtagtaaac”に対する Suffix CDAWG を図 3 に示す．ノードαから遷移し始める”gtaaac”のような，分岐なしに遷移が終端まで続く箇所が，CDAWG では 1 つの遷移にまとめられていることがわかる．この箇所は Separate String と見なすことができる．よって，ルートノードからノードαまでは従来通りダブル配列に登録し，α以降の Separate String を TAIL として登録する．TAIL には接尾辞が格納されるので，集約することが可能となり，効率的である．本提案手法では，グラフ構築に際しての省空間化，計算時間の短縮を図る．

5. 評価

全文検索において提案手法の有効性を実証するために検証実験をおこなった．Intel Xeon 2.53GHz Quad-Core×2，L2 キャッシュ 8M を実験環境として用いた．また，検索対象となる全文は Pizza&Chili Corpus[1]において公開されている DNA データ 50MB を使用し，そのうち先頭から文字列長分を抽出した．DNA データであるため，文字種は 4 種類のみである．今回は出現箇所を考慮せず，文書中にキーが含まれるか否かを検索する．全文のランダムな場所から抽出した 10000 個のキーに対する検索時間の合計を，検索時間として計測し，10 回の平均を取った．検索キー長は 81 文字とした．比較対象は簡潔データ構造によって実現されている圧縮接尾辞配列ライブラリ csalib100810[2]を用いた．圧縮する際のオプションは”-P3:512 –I:128:256”とした．検索

表 1：キー長 81 文字での検索時間[ms]

文字列長	100,000	500,000	900,000
csalib	110.4	108.6	114.5
従来のダブル配列	25.9	—	—
提案手法	10.6	12.4	12.4

表 2：辞書容量

文字列長	100,000	500,000	900,000
csalib	30KiB	147KiB	262KiB
従来のダブル配列	1.0MiB	—	—
提案手法	0.8MiB	4.2MiB	7.6MiB

時間と辞書容量の比較結果を表 1，2 に示す．また，Suffix DAWG をダブル配列で構築した場合を，従来のダブル配列として比較する．

結果として，csalib は文字列長が増加するにつれて，検索時間も増加するのに対し，提案手法は一定の検索速度を保つことが確認された．また，従来のダブル配列では長い文字列長において辞書構築できなかった．短い文字列から構築した辞書においては，従来のダブル配列と比べて，2 倍以上の検索速度を達成した．

6. おわりに

本論文では，キー検索法の 1 つであるダブル配列について，CDAWG を用いてグラフ構造を表現することにより，検索速度が高速な全文検索手法を提案した．今後は，より長い文字列長における実験と複数の文書を用いた全文検索実験を課題として設定する．複数文書における検索でも，部分共通文字列を統合できる本提案手法が活かせると考えている．

参考文献

[1] Aoe, J.: An efficient digital search algorithm by using a double-array structure, Software Engineering, IEEE Transactions on, Vol. 15, No. 9, pp. 1066--1077 (1989).

[2] Yata, S, Morita, K, Fuketa, M, and Aoe, J, "Fast string matching with space-efficient word graphs", In Proc. Innovations in Information Technology, pp. 79--83(2008).

[3] Crochemore, M. and Rytter, W.: Jewels of Stringology, World Scientific (2002).

[4] Crochemore, Maxime, and Renaud Vérin. "On compact directed acyclic word graphs." Structures in Logic and Computer Science. Springer Berlin Heidelberg, pp. 192--211(1997).

1 Pizza&Chili Corpus, http://pizzachili.dcc.uchile.cl/

2 圧縮接尾辞配列 csalib, http://researchmap.jp/sada/csalib/