様式 C-19

科学研究費補助金研究成果報告書

平成２３年　６月１３日現在

機関番号：６２６１５
研究種目：若手研究（A）
研究期間：２００７～２０１０
課題番号：１９６８０００１
研究課題名（和文）　実際的簡潔データ構造の開発と大量データ活用

研究課題名（英文）　Development of Practical Succinct Data Structures with Application to Huge Data
研究代表者
定兼　邦彦　（SADAKANE KUNIHIKO）
国立情報学研究所・情報学プリンシプル研究系・准教授
研究者番号：２０３２３０９０

研究成果の概要（和文）：これまで理論的な研究だけが行われてきた簡潔データ構造に対し，現実の計算機で用いる際の問題点を解決した，実際的な簡潔データ構造を開発した．順序木に対しては，既存の簡潔データ構造のサイズを 4 割削減し，なおかつこれまで実現できなかった多くの演算を行えるようになった．また，文字列検索の簡潔データ構造である圧縮接尾辞配列，圧縮接尾辞木のライブラリを作成した．これにより，110 ギガバイトの文書データからの検索を行うためのデータ構造のサイズを 680 ギガバイトから 22 ギガバイトに圧縮することができた．

研究成果の概要（英文）：There had been only theoretical researches on succinct data structures.　In this research, we have developed succinct data structures which solve the problems of using them on actual computers.　For ordinal trees, we have reduced the size of succinct data structures by 40%, while supporting various operations which had not been supported in existing data structures.　We have also created a library of compressed suffix arrays and compressed suffix trees, which are succinct data structures for string searches. By using it, we can reduce the size of the data structure, which is used for searching text data of 110 Gigabytes, from 680 Gigabytes to 22 Gigabytes.

交付決定額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	4,600,000	1,380,000	5.980,000
２００８年度	3,600,000	1,080,000	4,680,000
２００９年度	3,200,000	960,000	4,160,000
２０１０年度	3,200,000	960,000	4,160,000
年度			
総　計	14,600,000	4,380,000	18,980,000

研究分野：総合領域
科研費の分科・細目：情報学・情報学基礎
キーワード：データ圧縮，情報検索，文字列検索，簡潔データ構造，接尾辞配列，接尾辞木，圧縮接尾辞配列，順序木

１．研究開始当初の背景

Web ページ，関係データベース，新聞記事，ゲノム配列など，現在大量のデータが存在し，それらを有効活用するための技術の開発が重要となっている．古典的な手法では，データをディスクに格納し，それを逐次的に読み込みながら処理を行うが，これでは速度が非常に遅く，また，限られた処理しか行えない．そこで現在は「主記憶データベース」と呼ばれる，全データを計算機のメモリ(主記憶)に格納して高速処理を行う手法が取られている．しかしこの手法でもデータの検索手法は

古典的なアルゴリズムとデータ構造を用いているため，以下のような問題が存在する：
(1) 検索速度と，検索を行うために必要なメモリ量にはトレードオフが存在する．つまり，高速な検索を行うには大量のメモリが必要となる．
(2) 主記憶量の制限により，高速な検索を行えるデータ量は非常に小さくなる．
(3) データを圧縮することで必要メモリを減らすことができるが，データのランダムアクセスができなくなるため，検索が遅くなる．
これらの問題を解決するために提案されたものが簡潔データ構造である．
簡潔データ構造とは，データおよびそれを高速に処理するためのデータ構造（索引）のサイズを極限まで小さくし，なおかつ従来のデータ構造と同じ処理が同じ計算量で行えるものであり，申請者や海外の研究者らによりここ数年盛んに研究されている新しい概念である．現在は，高速な処理を実現するには大きなデータ構造を利用することが常識となっているが，そのために限られたメモリ内で扱えるデータ量が少なくなってしまう．ディスクを利用するアルゴリズムも存在するが，アクセス速度がメモリよりも数桁遅くなり，複雑な処理を行うことが出来ない．一方，簡潔データ構造を用いれば，大量のデータと索引をメモリに格納でき，高速処理が実現できる．これまでの主な結果は以下のようなものである．
(1) 文字列をそのエントロピーまで圧縮し，かつ，その文字列中に現れる任意のパタンの出現回数および出現位置を高速に求められる．
(2) 任意の配列構造をそのエントロピーまで圧縮し，任意の部分を定数時間で復元できる．つまり，データは圧縮してあるにも関わらず，読み込み速度は圧縮されていない場合と等しい
しかし，簡潔データ構造はまだ理論的な研究が始まったばかりであり，それを実際に活用する段階には至っていない．現在の理論的な結果をそのままプログラムとして実装すると，実行速度，索引サイズの点で満足のいくものにはなっていない．

２．研究の目的

簡潔データ構造に関するこれまでの理論的な結果をさらに洗練し，現実のデータに対して効率の良いデータ構造を開発し，それを用いて大量のデータからの知識発見を行う．具体的には，(1) 関係データベースマイニング（頻出要素発見），(2) 汎用連想計算エンジン GETA などで用いられている，文書や単語の類似度計算，(3) Web 構造マイニング（ハブ，コミュニティ発見），(4) 自然言語処理における，単語の共起関係計算や訳語の推定，(5) ゲノム情報処理，などで用いられる基本処理を少ないメモリで高速に実行するための基本データ構造を開発する．これらの問題に対する現在のデータ構造は古典的なものが使われており，最適とはほど遠い．例えば，(1) に対するデータ構造ではデータは圧縮されて保存されることが多いが，その結果，ランダムアクセスが出来なくなり，データが必要な場合には全て復元することになる．また，(2) では行列の行および列ベクトルの内積を計算するが，両方を効率よく計算するためにデータ構造のサイズが 2 倍になってしまっている．また，ランダムアクセスについても (1) と同様の問題がある．(3) については，Web のリンク構造を小さいサイズで表現し，かつ隣接点問い合わせなどを高速に実現する必要がある．(4) は翻訳の精度を上げるために動詞と名詞の共起関係を調べる必要があり，(5) では DNA，たんぱく質配列の検索を高速に行うための索引サイズが問題となる．

本研究では，上述の問題にあらわれる基本的な処理を少ないメモリで高速に実行するための実際的な簡潔データ構造を開発する．そして，それを用いて超大規模データを計算機の主記憶に圧縮して格納し，高速処理を行う．

開発したソフトウェアライブラリは申請者ホームページ等で公開する．

３．研究の方法

これまでの研究では，簡潔データ構造の漸近的な性能を理論的に示したものがほとんどである．例えば，集合を表現する $n + o(n)$ ビットの簡潔データ構造が構築できることを示す場合，$o(n)$ ビットの項は漸近的には無視できるとするが，その収束の速度が遅いため，極端に大きな n に対してのみ成り立つ議論となっていることが多い．しかしこれでは机上の空論であるため，実際に計算機のプログラムとして簡潔データ構造を作成する場合は，漸近的な議論ではなく，実データに対してどの程度の大きさのデータ構造になるのか，また，問い合わせ速度がどの程度なのかという点に注意する必要がある．そこで，本研究では，これまでに提案されてきた理論的な結果を洗練し，実際に有益な（サイズが小さく，問い合わせが高速な）データ構造を開発する．圧縮されたデータ構造についてのこのような立場の研究は過去にはほとんど存在しない．ごく最近になって，実データに対する問い合わせ速度を考慮したデータ構造が提案されているが，これらはサイズが大きく，また問い合わせ速度もまだ不十分である．本研究では，データ構造のサイズが小さく，問い合わせ速度も高速な簡潔データ構造を開発する．

４．研究成果

(1) 圧縮接尾辞木

接尾辞は文字列検索のための代表的データ構造である．長さ n，アルファベットサイズ σ の文字列 S に対し，従来のデータ構造では接尾辞木は O(n log n) ビット，具体的な実装では n < 2^{30} のときに 10n ～ 13n バイトの領域を必要としていた．これは文字列自身のサイズ(n log σ ビット，通常の文字列では n バイト，DNA では n/4 バイト）と比較すると非常に大きい．本研究で提案した圧縮接尾辞木は，接尾辞木の機能を保ったままサイズを圧縮する．圧縮接尾辞木のデータ構造は，文字列に対する圧縮接尾辞配列，接尾辞木の木構造を表現する簡潔データ構造，木の枝長を表現するデータ構造から構成される．各構成要素のサイズはそれぞれ O(n log σ) ビット，4n+o(n)ビット，2n+o(n)ビットである．このデータ構造は接尾辞木を線形サイズ（O(n log σ ビット）で表現する初めてのデータ構造である．接尾辞木の巡回などの演算は多くは圧縮前と同じ時間計算量で行える．一部の演算は圧縮前よりも遅くなるが，その計算量は圧縮接尾辞配列の 1 要素を復元する時間と等しく，速度低下はわずかである．

(2)最長反復部分文字列

文字列中の最長反復部分文字列を見つけるオンラインアルゴリズムを開発した．このアルゴリズムはデータ圧縮，パタン解析，データマイニングに利用できる．長さ n，アルファベットサイズ σ の文字列に対し，このアルゴリズムは O(n log σ) ビットの領域を用いて O(n log^3 n) 時間で動作する．このアルゴリズムを実装し，他手法と比較したところ，半分の作業領域で動作し，実行時間は他手法に引けを取らないことがわかった．

(3)圧縮接尾辞配列構築アルゴリズム

文字列検索のための簡潔データ構造である圧縮接尾辞配列を作成するための省スペースアルゴリズムを開発した．圧縮接尾辞配列を作成するためにはまず文字列の Burrows-Wheeler 変換を行う必要がある．これは文字列の長さ n の線形時間で構築できるが，単純なアルゴリズムでは O(n log n) ビットの作業領域を必要するが，本研究のアルゴリズムではこれを O(n log s log log_s n) ビット（ s はアルファベットサイズ）に削減した．これは線形時間で動作するアルゴリズムの中では必要メモリが最小になっている．

(4) 順序木の簡潔データ構造

順序木の新しい簡潔データ構造を開発した．これは既存のものよりも単純であり，データ構造のサイズも小さい．そして木に対する様々な操作，例えば子，親，祖先の節点を求める，節点の深さ，部分木の節点数，2 節点の共通祖先，節点の次数，部分木中の深さ最大の節点を求める，節点の行きがけ順，帰りがけ順，通りがけ順を求めるなどの操作を定数時間で実現できる．木が動的に変化する場合にも全ての操作を O(log n) 時間（n は木の節点数）で行える．順序木のデータ構造はこれまでは非常に複雑であり実用的ではなかったが，このデータ構造は単純であるため容易に実装でき，かつデータ構造のサイズも小さい．既存手法では子と親を求める操作のみを実現するもので 3.73n ビットを必要としていたが，提案手法では様々な演算を実現し，サイズは 2.32n ビットしか必要としない．この提案により現実的な解法が得られたといえる．

(5) 圧縮接尾辞配列ライブラリ csalib

これまでに開発してきた簡潔データ構造をライブラリとして公開した．このライブラリは，ビット列，文字列を格納するデータ構造，文字列の検索を行うための圧縮接尾辞配列，圧縮接尾辞木の簡潔データ構造を含む．また，メモリに収まらない大きさの接尾辞配列を構築する 2 つのプログラムを公開した．1 つは文字列の BW 変換がメモリに収まるが接尾辞配列はメモリに収まらない場合に，BW 変換を高速に実行するもので，もう 1 つは文字列の BW 変換もメモリに収まらない場合にディスクを使って変換を行うものである．後者のプログラムを用いて，日本の特許 5 年分の全文書約 110 ギガバイトに対する圧縮接尾辞配列を構築することに成功した．接尾辞配列のサイズは 680 ギガバイトだが，これを圧縮した圧縮接尾辞配列のサイズは約 22 ギガバイトとなり，大幅な圧縮を達成した．

(6) 文法圧縮

文法圧縮（grammar compression）は文字列をそれを生成する文脈自由文法に置き換える圧縮法であり，Lempel-Ziv，Byte-pair encoding，Sequitor，Re-Pair などの圧縮法を含む広い枠組みである． 本論文では，文法圧縮された文字列のランダムアクセスおよび部分復元アルゴリズムを与える． 長さ N の文字列がサイズ n の文法で表現されているとき，位置 i の長さ m の部分文字列の復元は O(m + log N) 時間で行える． データ構造の構築時間はポインタマシンモデルでは O(n αk(n))時間であり，RAM モデルでは O(n) 時間である（αk(n)は k 番目のアッカーマン逆関数）．また，文法圧縮された文字列上でのあいまい検索アルゴリズムや，文法圧縮された木構造の巡回アルゴリズムも提案した．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 11 件）
[1] P. Bille, G.M. Landau, R. Raman, K. Sadakane, S.S. Rao, O. Weimann. Random Access to grammar-Compressed Strings, Proceedings of ACM-SIAM SODA, 査読有, 373—389, 2011.
[2] T. Asano, J. Jansson, K. Sadakane, R. Uehara, G. Valiente. Faster Computation of the Robinson-Foulds Distance between Phylogenetic Networks, Proceedings of CPM, 査読有, LNCS 6129, 190—201, 2010.
[3] 田中洋輔, 小野廣隆, 定兼邦彦, 山下雅史. 高速復元可能な接尾辞配列圧縮法, 電子情報通信学会論文誌 D, 査読有, J93-D(8), 1567—1575, 2010.
[4] K. Sadakane, G. Navarro. Fully-Functional Succinct Trees, Proceedings of ACM-SIAM SODA, 査読有, 134—149, 2010.
[5] D. Arroyuelo, R. Canovas, G. Navarro, K. Sadakane. Succinct Trees in Practice, Proceedings of Workshop on Algorithm Engineering and Experiments (ALENEX), 査読有, 84—97, 2010.
[6] D. Okanohara, K. Sadakane. A Linear-Time Burrows-Wheeler Transform Using Induced Sorting, Proceedings of SPIRE, 査読有, LNCS 5721, 90—101. 2009.
[7] W. K. Hon, K. Sadakane, W. K. Sung. Breaking a Time-and-Space Barrier in Constructing Full-Text Indices, SIAM Journal on Computing, 査読有, 38(6)::2162—2178, 2009.
[8] D. Okanohara, K. Sadakane. An Online Algorithm for Finding the Longest Previous Factors, Proc. 16th Annual European Symposium (ESA), 査読有, LNCS 5193, 696—707, 2008.
[9] J. Larsson, K. Sadakane. Faster Suffix Sorting, Theoretical Computer Science, 査読有, 387(3):258—272, 2007.
[10] K. Sadakane. Compressed Suffix Trees with Full Functionality. Theory of Computing Systems. 査読有, 41(4):589—607, 2007.
[11] K. Sadakane. Succinct Data Structures for Flexible Text Retrieval Systems, Journal of Discrete Algorithms, 査読有, 5(1):12—22, 2007.

〔学会発表〕（計 5 件）
[1] 定兼　邦彦. 文法圧縮された文字列のランダムアクセス, 日本オペレーションズ・リサーチ学会研究部会 画期における最適化, 2010 年 12 月 7 日, 京都大学.
[2] 定兼　邦彦. 順序木の簡単・簡潔な表現法, 日本オペレーションズ・リサーチ学会研究部会 画期における最適化, 2009 年 11 月 26 日, 京都大学.
[3] 定兼　邦彦. 動的簡潔順序木, 電子情報通信学会コンピュテーション研究会, 2009 年 4 月 17 日, 京都大学.
[4] 定兼　邦彦. 括弧列の簡単・簡潔な表現法. 電子情報通信学会コンピュテーション研究会, 2008 年 10 月 10 日, 東北大学.
[5] 定兼　邦彦. DNA配列に適した圧縮全文索引, 電子情報通信学会コンピュテーション研究会, 2008 年 3 月 10 日, 日本IBM東京基礎研究所.

〔その他〕
ホームページ等

http://researchmap.jp/sada/csalib/

６．研究組織
(1)研究代表者
定兼　邦彦（SADAKANE KUNIHIKO）
国立情報学研究所・情報学プリンシプル研究系・准教授
研究者番号：20323090