複数文字列に対するデカルト木円形パターン照合の索引

Eric Osterkamp* Dominik Köppl†

概要

デカルト木照合は、テキストの部分文字列とパターンが同じデカルト木を共有する場合にマッチとみなす、一般化パターン照合の一形態である。この種の照合は株価の時系列データに応用されており、楽譜間のメロディー照合にも関心が持たれている。索引構築問題における最先端のデータ構造は [14] による、Burrows–Wheeler 変換に基づく解法であり、ほぼ簡潔な領域で動作し、パターン長に線形の時間でデカルト木マッチする部分文字列の個数を数えることができる。著者らは自分たちのデータ構造の構築法として単純な解法を提示しているが、ポインタベースのデータ構造を必要とするため、テキスト長を n としたときに $O(n \lg n)$ ビットの領域を要する [13, 節 A.4]。我々はこのボトルネックを解消するため、$O(n \lg \sigma)$ ビットの領域で構築可能な手法を提案する。このとき構築時間は $O\left(n \frac{\lg \sigma \lg n}{\lg \lg n}\right)$ であり、σ はアルファベットのサイズである。さらに、拡張 Burrows–Wheeler 変換の思想のもと、Kim と Cho の索引を複数の循環テキスト（circular texts）の索引化に拡張できることを示す。我々は、拡張後の索引が同じ計算量を保つことを証明し、動的バリアントも提示する。動的バリアントでは対数的な遅延を支払う代わりに、入力テキスト全体のビット数に線形の追加領域で済み、テキストを逐次的に追加・削除できる。応用例として、循環照合統計量（circular matching statistics）の計算を示す。この拡張的な設定は、オフセットや拡大縮小に依存せず、上述の応用分野に共通する繰り返しモチーフの発見において有用である。

*University of Muenster
†山梨大学 大学院　総合研究部　工学域 電気電子情報工学系

序論

文字列探索（String matching）は計算機科学のあらゆる分野で広く使われており、その変種は多種多様な問題を解決するために特化して設計されている。本稿では、その中でも **部分文字列一貫同値関係**（*substring consistent equivalence relation*, 以下 SCER）と呼ばれる特殊な変種に注目する [18]。2 つの文字列 X と Y は、長さが等しく、かつすべての同じ位置・同じ長さの部分文字列が SCER の意味で一致する場合に SCER-マッチするという。すなわち、すべての $1 \leq i \leq j \leq |X| = |Y|$ に対して、$X[i..j]$ は $Y[i..j]$ と SCER-マッチする。

SCER-matching の具体例としては **順序保持照合**（order-preserving matching）があり、これは数値時系列解析のために研究されてきた [12, 16]。順序保持照合は、2 つの文字列において文字の相対的な大小関係が同じである場合に両者を一致と見なすものである。したがって順序保持照合は、値のオフセットや拡大縮小に依存しない照合を可能にする。

しかし、順序保持照合は文字列中の全体的な順序を厳密に考慮するため、主にピークによって分割される範囲内の局所的な順序を重視する応用においては制約が厳しすぎる場合がある。実際、株価時系列などがその典型例であり、**ヘッド・アンド・ショルダーズ**パターン [5] という代表的なパターンでは、1 つの絶対ピーク（head）と、その両側の局所ピーク（shoulders）を含み、それらの局所ピークが個別に変化しても類似パターンとして扱えることが求められる。このような理由から、Park ら [23] は **デカルト木照合**（Cartesian tree matching）を提案した。デカルト木照合は、順序保持照合を緩和した概念であり、順序保持照合する 2 文字列は必ずデカルト木

マッチするが、その逆は必ずしも成立しない。例えば、数字列 1647253 と 2537164 はヘッド・アンド・ショルダーズパターンに適合し、順序保持照合はしないがデカルト木照合はする。デカルト木照合は、2 つの文字列のデカルト木を比較することで照合を判定する。**デカルト木** [25] は数値の配列を元に構築される二分木であり、すべての数値が異なる場合、それは最小ヒープ（min-heap）であり、その中間順巡回（in-order traversal）によって元の配列が得られる。

関連研究

デカルト木照合とその変種は提案以来、研究者の注目を集めており、複数パターン探索 [23, 24]、近似探索 [1, 15]、部分文字列探索 [2]、部分列探索 [20]、不確定文字探索 [7]、カバー探索 [11]、回文探索 [6] など多様な分野で研究されてきた。

デカルト木を用いたパターン照合は株価時系列以外にも応用があり、楽譜間のメロディー照合にも利用されている。音楽スコアのパターン照合においては差分、方向性、または大きさなどが研究対象となってきた [9]。しかし音楽の中で繰り返し部分を検出することも重要とされてきた [4]。そこで課題となるのは、デカルト木照合するような反復する旋律モチーフ群（つまり入力テキスト）の繰り返し部分文字列を検出できるかどうかである。効率的な探索のためには、これらのモチーフを索引化する必要がある。

テキスト文字列 T に対するデカルト木照合のための索引とは、T 上に構築され、与えられたパターンとデカルト木照合する T の部分文字列の個数を報告できるデータ構造である。T がサイズ σ の整数アルファベット上の長さ n の文字列であるとき、Park ら [23] および Nishimoto ら [19] は、それぞれ $O(n \lg n)$ ビットの領域を使用するデカルト木照合用の索引を提案している。構築にはそれぞれ $O(n \lg n)$ および $O(n\sigma \lg n)$ の追加ビット数が必要である。Park ら [23, Section 5.1] はカウントクエリに対して $O(m \lg \sigma)$ 時間を、Nishimoto ら [19, Section 5.1] は $O(m(\sigma + \lg m) + occ)$ 時間を達成しており、ここで occ は長さ m のパターンの出現回数である。

Ferragina と Mantaci による厳密文字列探索用の FM-index [3] を応用して、Kim と Cho [13] は $3n + o(n)$ ビットの領域で動作し、長さ m のパターンに対するカウントクエリを $O(m)$ 時間で答える索引を提案した。構築には、Park ら [23] のデカルト接尾辞木（Cartesian suffix tree）を入力として用いる単純な解法を提示しているが、$O(n \lg n)$ の追加ビット領域を必要とする。

結果とまとめ

本稿 [22] では、次の 2 つの目標を掲げる。第 1 は、Kim と Cho の索引の構築アルゴリズムであり、作業領域を $O(n \lg \sigma)$ ビットに抑えるものである。これはアルファベットサイズが定数のときコンパクトである。長さ n のテキスト T の 1 文字へのアクセスを対応する索引を $O(n \lg \sigma)$ ビットの領域で構築できればコンパクトと考えるが、ここではデカルト木パターン照合のクエリに限定している。デカルト木は n ノードを持つとき、$2n$ ビットで表現可能であることが知られている [23, Section 3.5]。従って、デカルト木パターン照合に特化したコンパクト索引は $O(n)$ ビットで表現できる。

第 2 は、上述のいずれの索引も複数のテキストの索引化は部分的には対応できるものの、パターンが入力テキスト群のいずれかの繰り返しであるかを検出するのが難しいという課題である。これは繰り返し可能な旋律モチーフの索引化を行う際に重要となる問題である。本稿の目標は、同一の繰り返し内でも異なるオフセットから始まるマッチを検出可能な索引を構築することである。具体的には、デカルトパターン照合のために複数のテキストを索引化することを目指す。パターンの探索領域は、各テキストが無限に自己連結した文字列として考えられる。

本稿では、Kim と Cho のデカルト木照合用索引 [13] を拡張し、拡張 Burrows–Wheeler 変換（extended BWT） [17] の技術を用いることを提案する。我々はこの新たなデータ構造を *cBWT* **索引** と

呼ぶ。cBWT 索引 は、複数の入力テキストをデカルト木照合のために循環的に索引化できるという意味で拡張である。本稿では、すべてのテキストの総長を n とするとき、cBWT 索引 を $O(n\frac{\lg\sigma\lg n}{\lg\lg n})$ 時間、かつ $O(n\lg\sigma)$ ビットの領域で構築可能であることを示す。我々の構築法は cBWT 索引 のインクリメンタル構築を可能とし、新たな文字列を逐次的に索引に追加できる。また、Kim と Cho の元の索引も同様の計算量で構築できる。我々の着想は、パラメータ化照合用索引の構築アルゴリズム [8, 10] から得たものであり、最近これを複数の循環テキストに対して拡張した [21]。cBWT 索引 は構築中に後向き探索（backward search）およびパターン文字列に対するカウントクエリを $O(m\frac{\lg\sigma\lg n}{\lg\lg n})$ 時間で対応する。ここで m はパターンの長さである。

参考文献

[1] Bastien Auvray, Julien David, Richard Groult, and Thierry Lecroq. Approximate Cartesian tree matching: An approach using swaps. In *Proc. SPIRE*, volume 14240 of *LNCS*, pages 49–61, 2023.

[2] Simone Faro, Thierry Lecroq, Kunsoo Park, and Stefano Scafiti. On the longest common Cartesian substring problem. *The Computer Journal*, 66(4):907–923, 2022.

[3] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Proc. FOCS*, pages 390–398, 2000.

[4] Peter Foster, Anssi Klapuri, and Simon Dixon. A method for identifying repetition structure in musical audio based on time series prediction. In *Proc. EUSIPCO*, pages 1299–1303. IEEE, 2012. ISBN 978-1-4673-1068-0.

[5] Tak-Chung Fu, Korris Fu-Lai Chung, Robert Wing Pong Luk, and Chak-man Ng. Stock time series pattern matching: Template-based vs. rule-based approaches. *Eng. Appl. Artif. Intell.*, 20(3):347–364, 2007.

[6] Mitsuru Funakoshi, Takuya Mieno, Yuto Nakashima, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Computing maximal palindromes in non-standard matching models. In *Proc. IWOCA*, volume 14764 of *LNCS*, pages 165–179, 2024.

[7] Paweł Gawrychowski, Samah Ghazawi, and Gad M. Landau. On indeterminate strings matching. In *Proc. CPM*, volume 161 of *LIPIcs*, pages 14:1–14:14, 2020.

[8] Daiki Hashimoto, Diptarama Hendrian, Dominik Köppl, Ryo Yoshinaka, and Ayumi Shinohara. Computing the parameterized Burrows–Wheeler transform online. In *Proc. SPIRE*, volume 13617 of *LNCS*, pages 70–85, 2022.

[9] Yuzuru Hiraga. Structural recognition of music by pattern matching. In *Proc. ICMC*. Michigan Publishing, 1997.

[10] Kento Iseri, Tomohiro I, Diptarama Hendrian, Dominik Köppl, Ryo Yoshinaka, and Ayumi Shinohara. Breaking a barrier in constructing compact indexes for parameterized pattern matching. In *Proc. ICALP*, volume 297 of *LIPIcs*, pages 89:1–89:19, 2024.

[11] Natsumi Kikuchi, Diptarama Hendrian, Ryo Yoshinaka, and Ayumi Shinohara. Computing covers under substring consistent equivalence relations. In *Proc. SPIRE*, volume 12303 of *LNCS*, pages 131–146, 2020.

[12] Jinil Kim, Peter Eades, Rudolf Fleischer, Seok-Hee Hong, Costas S. Iliopoulos, Kunsoo Park, Simon J. Puglisi, and Takeshi

Tokuyama. Order-preserving matching. *Theor. Comput. Sci.*, 525:68–79, 2014.

[13] Sung-Hwan Kim and Hwan-Gue Cho. A compact index for Cartesian tree matching. In *Proc. CPM*, volume 191 of *LIPIcs*, pages 18:1–18:19, 2021.

[14] Sung-Hwan Kim and Hwan-Gue Cho. Simpler FM-index for parameterized string matching. *Inf. Process. Lett.*, 165:106026, 2021.

[15] Sungmin Kim and Yo-Sub Han. Approximate Cartesian tree pattern matching. In *Proc. DLT*, volume 14791 of *LNCS*, pages 189–202, 2024.

[16] Marcin Kubica, Tomasz Kulczyński, Jakub Radoszewski, Wojciech Rytter, and Tomasz Waleń. A linear time algorithm for consecutive permutation pattern matching. *Inf. Process. Lett.*, 113(12):430–433, 2013.

[17] Sabrina Mantaci, Antonio Restivo, Giovanna Rosone, and Marinella Sciortino. An extension of the Burrows–Wheeler transform. *Theor. Comput. Sci.*, 387(3):298–312, 2007.

[18] Yoshiaki Matsuoka, Takahiro Aoki, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Generalized pattern matching and periodicity under substring consistent equivalence relations. *Theor. Comput. Sci.*, 656:225–233, 2016.

[19] Akio Nishimoto, Noriki Fujisato, Yuto Nakashima, and Shunsuke Inenaga. Position heaps for Cartesian-tree matching on strings and tries. In *Proc. SPIRE*, volume 12944 of *LNCS*, pages 241–254, 2021.

[20] Tsubasa Oizumi, Takeshi Kai, Takuya Mieno, Shunsuke Inenaga, and Hiroki Arimura. Cartesian tree subsequence matching. In *Proc. CPM*, volume 223 of *LIPIcs*, pages 14:1–14:18, 2022.

[21] Eric M. Osterkamp and Dominik Köppl. Extending the parameterized Burrows–Wheeler transform. In *Proc. DCC*, pages 143–152, 2024.

[22] Eric M. Osterkamp and Dominik Köppl. Extending the Burrows–Wheeler transform for Cartesian tree matching and constructing it. In *Proc. CPM*, volume 331 of *LIPIcs*, pages 26:1–26:17, 7 2025.

[23] Sung Gwan Park, Magsarjav Bataa, Amihood Amir, Gad M. Landau, and Kunsoo Park. Finding patterns and periods in Cartesian tree matching. *Theor. Comput. Sci.*, 845:181–197, 2020.

[24] Siwoo Song, Geonmo Gu, Cheol Ryu, Simone Faro, Thierry Lecroq, and Kunsoo Park. Fast algorithms for single and multiple pattern Cartesian tree matching. *Theor. Comput. Sci.*, 849:47–63, 2021.

[25] Jean Vuillemin. A unifying look at data structures. *Commun. ACM*, 23(4):229–239, 1980.