1変数項木パターンに対する多項式時間マッチングアルゴリズム

舛井 里帆[1]　池森 千尋[1]　鈴木 祐介[2,a)]　内田 智之[2,b)]　宮原 哲浩[2,c)]

概要：順序木とは辺ラベルを持ち，順序付けられた子を持つ根付き木である．項木パターンとは，順序木に構造的変数を導入した順序木パターンであり，構造的変数には任意の順序木を代入できる．1変数項木パターンとは，パターン中の全ての変数に同一の順序木を代入しなくてはならない制限を持つ項木パターンである．本研究では，1変数項木パターンに対する多項式時間マッチングアルゴリズムの提案を行う．

キーワード：グラフアルゴリズム，機械学習，データマイニング

A Polynomial Time Matching Algorithm for One-Variable Term Tree Patterns

Riho Masui[1]　Chihiro Ikemori[1]　Yusuke Suzuki[2,a)]　Tomoyuki Uchida[2,b)]
Tetsuhiro Miyahara[2,c)]

Abstract: Ordered trees are rooted trees with ordered children and edge labels. Term tree patterns are rooted ordered trees having internal structured variables. A variable can be replaced with any rooted ordered tree. A one-variable term tree pattern is a term tree pattern with a restriction that all variables in the term tree pattern must be replaced with the same ordered tree. In this paper, we proposed a polynomial time matching algorithm for one-variable term tree patterns.

Keywords: graph algorithm, machine learning, data mining

1. はじめに

HTML ファイルや XML ファイルなどの Web 上に存在するデータは，タグの入れ子を親子関係とみなすと木構造を持つデータとして表現することができる．データマイニングの分野ではこのような木構造を持つデータに共通するパターンを発見することが注目されている．HTML ファイルや XML ファイルのような木構造を持つデータは，辺ラベルを持ち，各内部頂点が順序付けされた子を持つ根付き木として表される．このような根付き木を順序木と呼ぶ．

本研究では，順序木に共通するパターンを表現する方法として，**項木パターン (term tree pattern)** を用いる [3]．項木パターンとは，順序木の内部に変数の存在を認めたもので，変数には任意の順序木を代入することができる．変数は，辺と同様に頂点の組とラベル（変数ラベル）から構成される．代入の際に各変数は変数ラベルによって区別され，同一の変数ラベルを持つ変数には，同一の順序木を代入しなければならない．項木パターン t と順序木 T が与えられたとき，t の各変数に適当な順序木を代入することで順序木 T と同型になるならば，項木パターン t と順序木 T はマッチするという．図 1 に項木パターンと順序木のマッチの例を示す．図 1 の項木パターン f は順序木 T_3 を変数

[1] 広島市立大学情報科学部
Faculty of Information Sciences, Hiroshima City University, Japan
[2] 広島市立大学情報科学研究科
Graduate School of Information Sciences, Hiroshima City University, Japan
[a)] y-suzuki@hiroshima-cu.ac.jp
[b)] uchida@hiroshima-cu.ac.jp
[c)] miyares20@hiroshima-cu.ac.jp

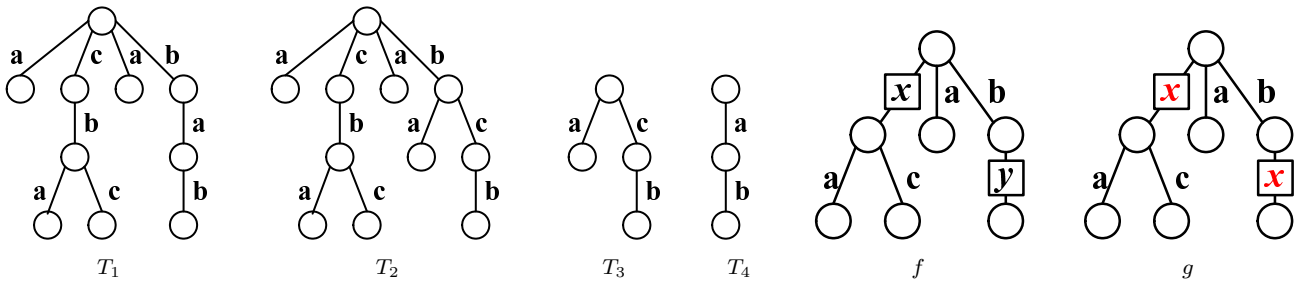

図 1 順序木 T_1, T_2, T_3, T_4，項木パターン f と 1 変数項木パターン g．項木パターンおよび 1 変数項木パターンの変数を，頂点をつなぐ線と線上の四角で表す．四角内のラベルは，その変数の変数ラベルを表す．項木パターン f と順序木 T_1, T_2 はマッチする．1 変数項木パターン g と順序木 T_2 はマッチするが，g と順序木 T_1 はマッチしない．

x に，順序木 T_4 を変数 y に代入することにより，順序木 T_1 と同型になる．よって，項木パターン f と順序木 T_1 はマッチする．また，項木パターン f は順序木 T_3 を変数 x, y に代入することにより，順序木 T_2 と同型になるため，項木パターン f と順序木 T_2 はマッチする．

項木パターンに対するマッチング問題とは，項木パターン t と順序木 T が入力として与えられたときに，t と T がマッチするか否かを判定する問題である．項木パターンに対するマッチング問題を解くアルゴリズムをマッチングアルゴリズムという．項木パターン中の変数が互いに異なる変数ラベルを持つとき，その項木パターンを正則な項木パターンと呼ぶ．正則な項木パターンに対するマッチング問題は多項式時間計算可能であることが示されている [4]．しかし正則でない項木パターンに対するマッチング問題は NP 完全であることが分かっている [4]．

パターン言語 (pattern language) に関する研究では，パターン中に 1 種類の変数しか現れない 1 変数パターンの学習可能性が研究されている [1], [2]．本研究では，これを項木パターンに拡張し，**1 変数項木パターン (one-variable term tree pattern)** を提案する．1 変数項木パターンとは，項木パターン中の全ての変数が同一の変数ラベルを持つものである．代入の定義より，1 変数項木パターンは，項木パターン中の全ての変数に同一の順序木を代入するという制限を持つ．図 1 に 1 変数項木パターンの例を示す．図 1 の 1 変数項木パターン g は，順序木 T_3 を変数 x に代入することにより，順序木 T_2 と同型になるため，1 変数項木パターン g と順序木 T_2 はマッチする．しかし，1 変数項木パターン g と順序木 T_1 はマッチしない．本研究では，1 変数項木パターンに対するマッチング問題を解く多項式時間マッチングアルゴリズムを提案する．また提案したマッチングアルゴリズムを計算機上に実装し，その評価実験を行った結果を報告する．

2. 準備

本節では，項木パターン，項木パターン言語に関する諸定義を行う．Λ を辺ラベルの集合，X を $\Lambda \cap X = \emptyset$ を満たす変数ラベルの集合とする．$T = (V, E)$ を $\Lambda \cup X$ 上の木とする．ここで，V を頂点集合，$E \subseteq V \times (\Lambda \cup X) \times V$ を辺集合とする．Λ の要素 a でラベル付けされている頂点 u, v 間の辺を $e = (u, a, v)$ で表し，辺 e の辺ラベルを $\Lambda(e)$ で表す．X の要素 x でラベル付けされている頂点 u, v 間の辺を変数 (variable) と呼び，$h = [u, x, v]$ で表し，変数 h の変数ラベルを $X(h)$ で表す．2 つの頂点 $u, v \in V$ に対して，辺 $e = (u, a, v) \in E$ または変数 $h = [u, x, v] \in E$ であるとき，u は v の親であるといい，v は u の子であるという．特に変数 $h = [u, x, v] \in E$ であるとき，u は v の親ポート，v は u の子ポートであるという．頂点 $v \in V$ に対して，$indeg(v)$ を頂点 v の入次数 (頂点 v に入ってくる辺の数) とし，$outdeg(v)$ を v の出次数（頂点 v から出ていく辺の数）とする．$indeg(u) = 0$ であるような頂点 u を根と呼び，$outdeg(v) = 0$ であるような頂点 v を葉と呼ぶ．順序木とは，根を持ち，任意の葉以外の頂点に対して，その頂点の子の集合に順序が定義されている木である．順序木 T の頂点 u と，u の 2 つの子 u', u'' に対して，$u' <_u^T u''$ は u の子の順序において u' が U'' より小さいことを表す．

定義 1 $\Lambda \cup X$ 上の順序木 $T = (V, E)$ に対し，$V' = V$，E 中の変数全体の集合を H'，$E' = E - H'$ とするとき，3 つ組 $t = (V', E', H')$ を $\Lambda \cup X$ **上の項木パターン** (または単に**項木パターン**) という．

項木パターン $g = (V_g, E_g, H_g)$ に対し，$H_g = \emptyset$ のとき，g を基礎項木パターンという．これ以降，基礎項木パターンを Λ **上の順序木** (または単に**順序木**) と呼ぶ．Λ 上の順序木の集合を OT_Λ と記述する．

定義 2 $\Lambda \cup X$ 上の項木パターン $f = (V_f, E_f, H_f), g = (V_g, E_g, H_g)$ に対し，f と g が**同型**であるとは，次の 3 つの条件を満たす全単射 $\varphi : V_f \to V_g$ が存在するときという．このとき $f \cong g$ と書く．

(1) $(u, a, v) \in E_f$ のとき，その時に限り $(\varphi(u), a, \varphi(v)) \in E_g$ である．

(2) ある $x \in X$ に対して $[u, x, v] \in H_f$ のとき，その時に限り，ある $y \in X$ に対して $[\varphi(u), y, \varphi(v)] \in H_g$ である．さらに 2 つの変数 $[u, x, v], [u', x', v'] \in H_f$

に対し，$x \neq x'$ $(x, x' \in X)$ ならばその時に限り $[\varphi(u), y, \varphi(v)], [\varphi(u'), y', \varphi(v')] \in H_g$ に対し，$y \neq y'$ $(y, y' \in X)$ である．

(3) f の頂点 u とその 2 つの子 u', u'' において，$u' <_u^f u''$ のとき，そのときに限り，$\varphi(u') <_{\varphi(u)}^g \varphi(u'')$ である．

項木パターンに対する代入について定義する．f, g を 2 つ以上の頂点を持つ項木パターンとする．$h = [v_0, x, v_1]$ を f の変数，$\sigma = [u_0, u_1]$ を g の異なる頂点のリストとする，ここで u_0 は g の根であり，u_1 は g の葉である．このとき $x := g, [u_0, u_1]$ を変数ラベル x に対する束縛という．束縛 $x := g, [u_0, u_1]$ を f に次のように適用して新しい項木パターン f' を得る．変数ラベル x を持つ変数 h に対して，f の変数の集合 H_f から変数 h を削除し，頂点 v_0 と g の頂点 u_0 を，頂点 v_1 と g の頂点 u_1 をそれぞれ同一視する．

新しい項木パターン f' の 2 つ以上の子を持つ頂点 v の子の順序を次のように定める．v', v'' を v の 2 つの子とする．(1) $v, v', v'' \in V_g$ であり，$v' <_v^g v''$ ならば，$v' <_v^{f'} v''$ である．(2) $v, v', v'' \in V_f$ であり，$v' <_v^f v''$ ならば，$v' <_v^{f'} v''$ である．(3) $v = v_0$，$v' \in V_f$，$v'' \in V_g$ であり，$v' <_v^f v_1$ ならば，$v' <_v^{f'} v''$ である．(4) $v = v_0$，$v' \in V_f$，$v'' \in V_g$ であり，$v_1 <_v^f v'$ ならば，$v'' <_v^{f'} v'$ である．

束縛の有限集合 $\theta = \{x_1 := [g_1, \sigma_1], \ldots, x_n := [g_n, \sigma_n]\}$ を代入と呼ぶ．ここで $x_1, \ldots, x_n$ は X に含まれる互いに異なる変数ラベルである．代入 θ に含まれる束縛を項木パターン f にすべて適用して得られる新しい項木パターンを $f\theta$ と書く．順序木 T と項木パターン t に対し T と $t\theta$ が同型となるような代入 θ が存在するとき，t と T がマッチするという．

同一の変数ラベルを持つ変数が項木パターン中に複数回現れるとき，その変数を繰り返し変数 (repeated variable) と呼ぶ．項木パターン中の変数ラベルの数が 1 のとき，つまり，全ての変数が同一の変数ラベルを持つとき，その項木パターンを **1 変数項木パターン**と呼ぶ．代入の定義より，1 変数項木パターンへの代入は，ただ 1 つの変数ラベルに対する束縛だけからなる，つまり，項木パターンの全ての変数に同一の順序木が代入される．簡単のため，1 変数項木パターン中の変数ラベルを x と書く．1 変数項木パターンが複数の変数を持つとき，その 1 変数項木パターンを 1 つの繰り返し変数を持つ項木パターンと呼ぶ．

$T \in OT_\Lambda$ を順序木とする．T の部分グラフ S に対し，S が順序木であり，S の各頂点の親子関係と兄弟関係の順序関係が T と等しいならば，S を T の部分木という．次の 3 つの条件を満たすとき，T の部分木 S と，T の頂点の組 (u, v) を T の **2-port 対応部分木**と呼び，$S(u, v)$ と書く．(1) u は S の根，v は S の葉である．(2) S の任意の 2 つの頂点 w_1, w_2 に対し，S 上で w_1 のすぐ隣の兄弟が w_2 であるなら，T 上でも w_1 のすぐ隣の兄弟が w_2 である．(3) u, v を除く S の全ての頂点の次数は，T における各頂点の次数と等しい．条件 (3) より，T の 2-port 対応部分木 $S(u, v)$ に対して，v を除く $S(u, v)$ の全ての葉は T の葉である．順序木 T の 2-port 対応部分木 $S(u, v)$ は，$S(u, v)$ の根 u，$S(u, v)$ において u の子の先頭となる頂点，$S(u, v)$ において u の子の末尾となる頂点，$S(u, v)$ の葉 v の 4 つの頂点を T から選ぶことで定めることができる．よって T の全ての 2-port 対応部分木の数は高々 $O(N^4)$ である．ここで N は T の頂点数である．

1 変数項木パターンに対する代入の例と 2-port 対応部分木の例を図 2 に示す．

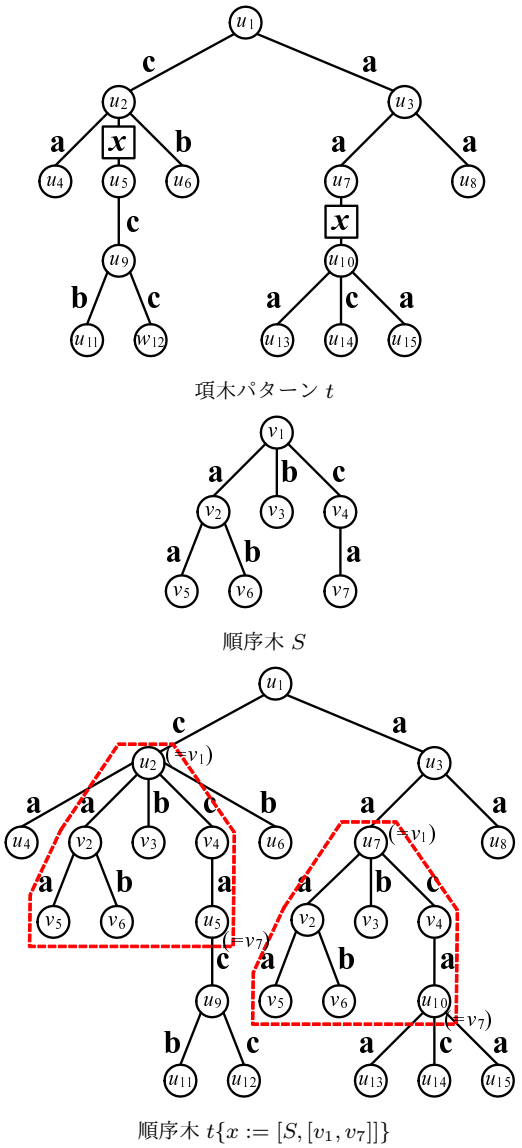

図 2 項木パターン t への代入の例．変数ラベル x を持つ変数に対し，順序木 S を代入し，順序木 $t\{x := [S, [v_1, v_7]]\}$ を得る．順序木 $t\{x := [S, [v_1, v_7]]\}$ の 2-port 対応部分木 $S(v_1, v_7)$ を破線の枠で示す．

3. 1 変数項木パターンに対するマッチングアルゴリズム

1 変数項木パターンに対するマッチング問題を次のように定義する．

1 変数項木パターンに対するマッチング問題

入力： 1 変数項木パターン t，順序木 $T \in OT_\Lambda$

問題： t と T がマッチするか否かを判断する

本節では，1 変数項木パターンに対するマッチング問題を解く多項式時間マッチングアルゴリズム OneMatching を提案する．

マッチングアルゴリズム OneMatching をアルゴリズム 1 に示す．入力として 1 変数項木パターン t と順序木 T が与えられたときのアルゴリズム OneMatching の動作を説明する．OneMatching は順序木 T の 2-port 対応部分木を列挙する．そして列挙した各 2-port 対応部分木を t の変数に代入し，T と同型になるか否かを確かめる．もし T と同型になる 2-port 対応部分木が存在すれば OneMatching は “yes” を返し，同型になる 2-port 対応部分木が存在しなければ，OneMatching は “no” を返す．N を順序木 T の頂点数とする．OneMatching は順序木 T の 2-port 対応部分木を列挙する際に T の全ての 2-port 対応部分木を列挙するわけではなく，項木パターン t の変数の位置に基づき高々 N^2 個の 2-port 対応部分木を列挙する．列挙した各 2-port 対応部分木を t の変数に代入し，T と同型になるか否かを確かめるのに $O(N)$ 時間かかるため，OneMatching は，1 変数項木パターンに対するマッチング問題を $O(N^3)$ 時間で計算する．

アルゴリズム 1 OneMatching(t, T)

Input: 頂点数 n と変数の数 m を持つ 1 変数項木パターン t, 頂点数 N の順序木 T
Output: “yes” or “no”

```
1: if (N − n) が m で割り切れない then
2:     return “no”
3: end if
4: u′ を幅優先探索で最初に到達する t の変数の親ポート，f をその順番とする
5: u を幅優先探索で f 番目の T の頂点とする
6: c_1, ..., c_k を u の全ての子とする
7: i を u′ を親とする変数の子ポートで，最も順番が小さいものの順番とする
8: for j := i to k do
9:     subT を u と，c_i ..., c_j とその子孫全体からなる T の部分木とする
10:    for each subT の頂点 v do
11:        S を subT から v の子孫全体を除いた T の部分木とする（v は S に含まれる）
12:        s を S の頂点数とする
13:        if N = (n + (s − 2) × m) then
14:            if t{x := [S, [u, v]]} ≅ T then
15:                return “yes”
16:            end if
17:        end if
18:    end for
19: end for
20: return “no”
```

補題 1 1 変数項木パターン t と順序木 T がマッチするならばその時に限り，$t\{x := [S(u,v),[u,v]]\} \cong T$ を満たす T の 2-port 対応部分木 $S(u,v)$ が存在する．

証明 ($\Longrightarrow$) t と T がマッチするならば，$t\{x := [S(u,v),[u,v]]\} \cong T$ を満たす T の 2-port 対応部分木 $S(u,v)$ が存在することを示す．

t と T がマッチするので，マッチの定義より，$t\{x := [Q,[w_1,w_2]]\} \cong T$ を満たす順序木 Q が存在する．ここで w_1 は Q の根，w_2 は Q のある葉である．代入の定義より，Q の親子関係と兄弟の順序関係は代入後の T においても保たれる．よって Q と同型な T の部分木 S が存在する．V_Q を Q の頂点集合，V_S を S の頂点集合とし，ξ を V_Q から V_S への同型写像とする．w_1, w_2 以外の Q の頂点 w は，代入の前後で次数が変化しないため，S の頂点 $\xi(w) \in V_S \setminus \{\xi(w_1), \xi(w_2)\}$ の S における次数と T における次数は等しい．また代入の前後で，Q の連続する 2 つの頂点 c_1, c_2 の間に頂点が挿入されることはないため，S 上で $\xi(c_1)$ のすぐ隣の兄弟が $\xi(c_2)$ であるなら，T 上でも $\xi(c_1)$ のすぐ隣の兄弟が $\xi(c_2)$ である．よって部分木 S と頂点の組 $(\xi(w_1), \xi(w_2))$ は T の 2-port 対応部分木である．このとき，$t\{x := [Q,[w_1,w_2]]\} \cong T$ なので，$t\{x := [S(\xi(w_1),\xi(w_2)),[\xi(w_1),\xi(w_2)]]\} \cong T$ が成り立つ．

($\Longleftarrow$) マッチの定義より，$t\{x := [S(u,v),[u,v]]\} \cong T$ を満たす順序木 $S(u,v)$ が存在するならば，t と T がマッチする． □

1 変数項木パターン t に対し，T を t とマッチする順序木とする．k を幅優先探索で最初に到達する t の変数の子ポートの順番とする．代入の定義より，幅優先探索で t の根から $k-1$ 番目の頂点までで構成される順序木と T の根から $k-1$ 番目の頂点までで構成される順序木は同型となる．よって，$t\{x := [S(u,v),[u,v]]\} = T$ を満たす T の 2-port 対応部分木 $S(u,v)$ の 1 つに対し，$S(u,v)$ の根 u と， $S(u,v)$ において u の子の先頭となる頂点の 2 つを一意に特定することができる．これを用いて，アルゴリズム OneMatching は入力順序木 T の 2-port 対応部分木を高々 $O(N^2)$ 個列挙する．

定理 1 アルゴリズム OneMatching は 1 変数項木パターンと N 頂点を持つ順序木に対するマッチング問題を $O(N^3)$ 時間で正しく解く．

4. 評価実験

3 節で提案した，1 変数項木パターンに対するマッチング問題を解くマッチングアルゴリズム OneMatching を計算機上に実装し，その評価実験を行った．主記憶メモリ:32.0GB，CPU:Intel(R) Core i5 3.70GHz，OS:MacOS

Mojave の計算機上に開発環境:eclipse，開発言語:Java を用いて提案したアルゴリズム OneMatching を実装し，評価実験を行った．

評価実験として，頂点数 n の項木パターンと，その項木パターンにマッチする（またはマッチしない）頂点数 N の順序木の組を，OneMatching の入力として与え，その実行時間を計測した．項木パターンと順序木の頂点数に関して，以下の二つの実験を行った．

(1) 項木パターンの頂点数 $n \in \{10, 20\}$，順序木の頂点数 $N \in \{50, 100, \ldots, 450, 500\}$ とし，各パラメータに対し項木パターンと順序木の組を 100 組ランダムに作成し，OneMatching の平均実行時間を計測した．項木パターンと順序木がマッチする場合とマッチしない場合の 2 通りの実験を行った．

(2) 項木パターンの頂点数 $n \in \{10, 20, 30, 40, 50\}$，順序木の頂点数 $N = 100$ とし，各パラメータに対し項木パターンと順序木の組を 100 組ランダムに作成し，OneMatching の平均実行時間を計測した．項木パターンと順序木がマッチする場合とマッチしない場合の 2 通りの実験を行った．

実験 (1) の結果を図 3,4 に示す．実験 (1) の結果より，順序木の頂点数が増加すると OneMatching の実行時間は増加するが，線形ではないことが確認できる．アルゴリズム OneMatching の計算量は $O(N^3)$ であり，実装したアルゴリズムの実行時間が計算量の見積もりに沿っていることが確認できた．

実験 (2) の結果を図 5,6 に示す．実験 (2) の結果より，項木パターンの頂点数が増加すると OneMatching の実行時間が減少することが確認できる．項木パターンの頂点数が増加すると，項木パターン中で幅優先探索で最初に発見される変数の位置が根から遠ざかるため，順序木中の 2-port 対応部分木を探索する範囲が狭まり，列挙する 2-port 対応部分木の数が減るためだと考えられる．

さらに，実験 (1)(2) の結果より，入力順序木と入力項木パターンがマッチする場合とマッチしない場合では，マッチしない場合のほうが実行時間がかかることが分かる．この理由として，OneMatching では，マッチする場合は 2-port 対応部分木の列挙が途中で終了するが，マッチしない場合は候補となりうる全ての 2-port 対応部分木を列挙しなくてはならないため実行時間が増加すると考えられる．

5. おわりに

本研究では，項木パターン中の全ての変数が同一の変数ラベルを持つ 1 変数項木パターンを提案した．また，1 変数項木パターンと順序木がマッチするか否かを判定するマッチング問題を解く多項式時間マッチングアルゴリズムを提案した．さらに，提案したマッチングアルゴリズムを計算機上に実装し，その評価実験を行った．今後の課題と

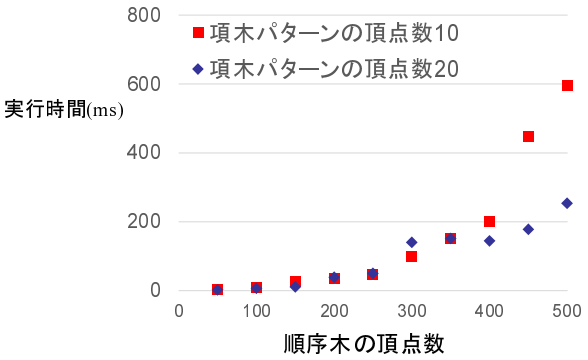

図 3 実験 (1) 項木パターンと順序木がマッチする場合の順序木の頂点数に対する平均実行時間の変化

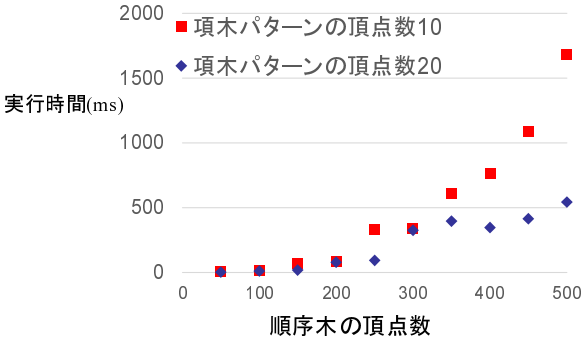

図 4 実験 (1) 項木パターンと順序木がマッチしない場合の順序木の頂点数に対する平均実行時間の変化

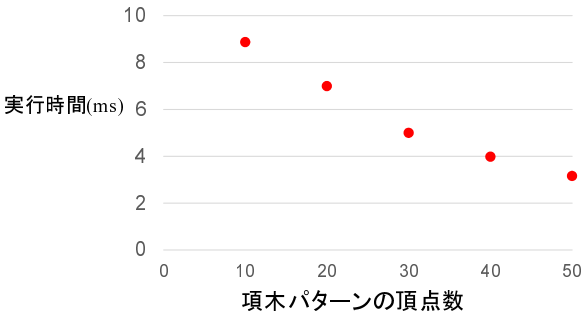

図 5 実験 (2) 項木パターンと順序木がマッチする場合の項木パターンの頂点数に対する平均実行時間の変化

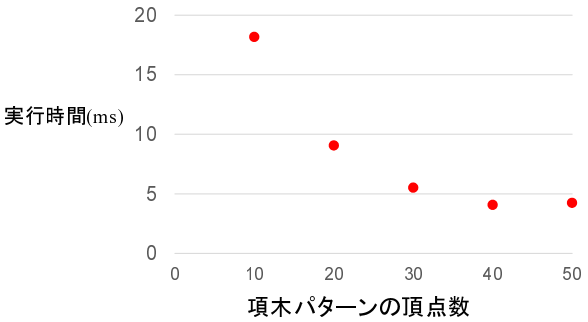

図 6 実験 (2) 項木パターンと順序木がマッチしない場合の項木パターンの頂点数に対する平均実行時間の変化

してはマッチングアルゴリズムの高速化が考えられる．項木パターン中の全ての変数が互いに異なるラベルを持つとき，正則であるという．頂点数 n の正則な項木パターンと頂点数 N の順序木に対するマッチング問題は $O(nN)$ 時間で解けることが示されている [4]．これに対し，本研究で提案したマッチングアルゴリズムの計算量は $O(N^3)$ 時間である．また，項木パターン中の変数ラベルの数が高々 k 個である項木パターン (k 変数項木パターン) に対するマッチングアルゴリズムの開発が考えられる．

本研究の発展としては，1 変数項木パターン言語のクラスの正例からの多項式時間帰納推論可能性の考察が考えられる．正則な項木パターンに対しては，項木パターン言語のクラスが正例から多項式時間帰納推論可能であることが示されている [3]．1 変数項木パターン言語のクラスが正例からの多項式時間帰納推論可能性であることを示すために，1 変数項木パターンに対する極小言語問題を解くアルゴリズムの提案を行う必要がある．

謝辞

本研究は JSPS 科研費 JP19K12102 の助成を受けたものです．

参考文献

[1] D. Angluin, Finding pattern common to a set of string, Journal of Computer and Systems Science, 21: 46–62, 1980.

[2] K. Baba, S. Tsuruta, A. Shinohara, M. Takeda, On the Length of the Minimum Solution of Word Equations in One Variable, Proc. 28th International Symposium on Mathematical Foundations of Computer Science (MFCS2003), LNCS 2747, pp. 189-197, 2003.

[3] Y. Suzuki, T. Shoudai, T. Uchida,T. Miyahara, Ordered Term Tree Languages Which Are Polynomial Time Inductively Inferable from Positive Data, Theoretical Computer Science, 350(1): 63–90, 2006.

[4] Y. Suzuki, T. Shoudai, T. Uchida,T. Miyahara, An Efficient Pattern Matching Algorithm for Ordered Term Tree Patterns, IEICE Trans. Inf. Syst., Vol. E98-A(6): 1197–1211, 2015.