Vol.2021-AL-182 No.11
2021/3/17

1変数項木パターンに対するマッチングアルゴリズムの改良

酒井　笑理[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，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変数項木パターンと順序木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の改良を行う．また，提案したマッチングアルゴリズムの実装を行い，その実験結果を報告する．

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

Improvement of a Matching Algorithm for One-Variable Term Tree Patterns

Emiri Sakai[1]　Yusuke Suzuki[2,a)]　Tomoyuki Uchida[2,b)]　Tetsuhiro Miyahara[2,c)]

Abstract: Ordered trees are rooted trees with ordered children. 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 an improved matching algorithm for one-variable term tree patterns and reported experimental results.

Keywords: graph algorithm, machine learning, data mining

1. はじめに

Web上に存在するデータ量は日々増大しており，それに伴い，HTMLファイルなどの木構造を持つデータもまた増大している．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や，特徴的なパターンがデータ中に出現するかを判定することが注目されている．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の表現方法として，**項木パターン (term tree pattern)** を用いる [4]．項木パターンとは，順序木の内部に構造的変数の存在を認めたもので，構造的変数には任意の順序木を代入することができる．これ以降，構造的変数を単に変数と呼ぶ．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．変数への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

[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) miyares21@hiroshima-cu.ac.jp

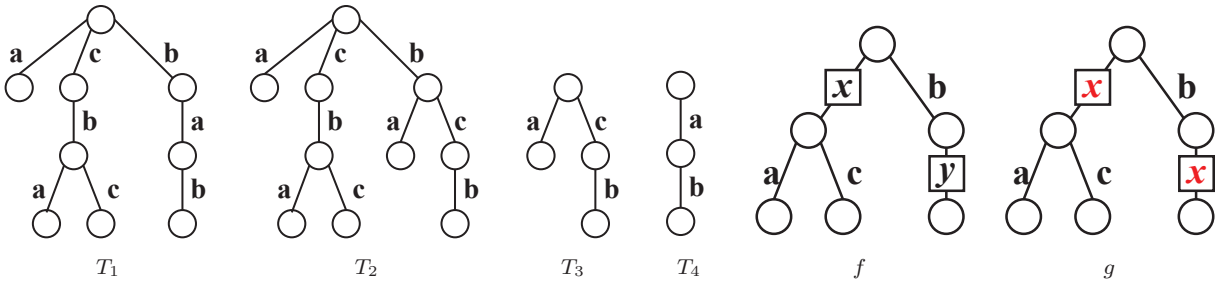

図 1 順序木 T_1, T_2, T_3, T_4，項木パターン f と 1 変数項木パターン g．項木パターンおよび 1 変数項木パターンの変数は，2 つの頂点の組と変数ラベルで構成される．変数を頂点の組をつなぐ線と線上の四角で表す．四角内のラベルは，その変数の変数ラベルを表す．項木パターン f と順序木 T_1, T_2 はマッチする．1 変数項木パターン g と順序木 T_2 はマッチするが，g と順序木 T_1 はマッチしない．

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 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．さらに，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．

項木パターンと順序木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．項木パターンと順序木に対するマッチング問題を解くアルゴリズムを，項木パターンに対する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．一般的な項木パターンに対するマッチング問題は NP 完全であることが示されている．しかし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 [5]．また正則な項木パターンの言語のクラスについては正例から多項式時間学習可能であることが示されている [4]．

Angluin[1] は正例から推論可能な言語族としてパターン言語 (pattern language) を導入した．パターン言語ではパターンに含まれる変数が異なる場合，正則パターンとよばれる．パターン言語に関する研究では，正則パターン，正則でないパターンの両方について研究が行われている．特に，正則でないパターンに関して，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 (one-variable pattern language) について研究されている [2], [3]．舛井ら [6] は，この概念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)** を提案した．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．よって，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 にどのような順序木を代入しても，順序木 T_1 と同型にならないため，1 変数項木パターン g と順序木 T_1 はマッチしない．

本研究で扱う 1 変数項木パターンと順序木に対するマッチング問題とは，1 変数項木パターン 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．舛井らは，1 変数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を $O(N^3)$ 時間で解くマッチングアルゴリズムを提案した [6]．本研究では，舛井らのマッチングアルゴリズムを改良し，1 変数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を $O(N^2)$ 時間で解くマッチングアルゴリズムを提案する．また改良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 [4]，1 変数項木パターン [6] に関する諸定義を行う．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．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 $\Lambda \cup X$ 上の順序木とする．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．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

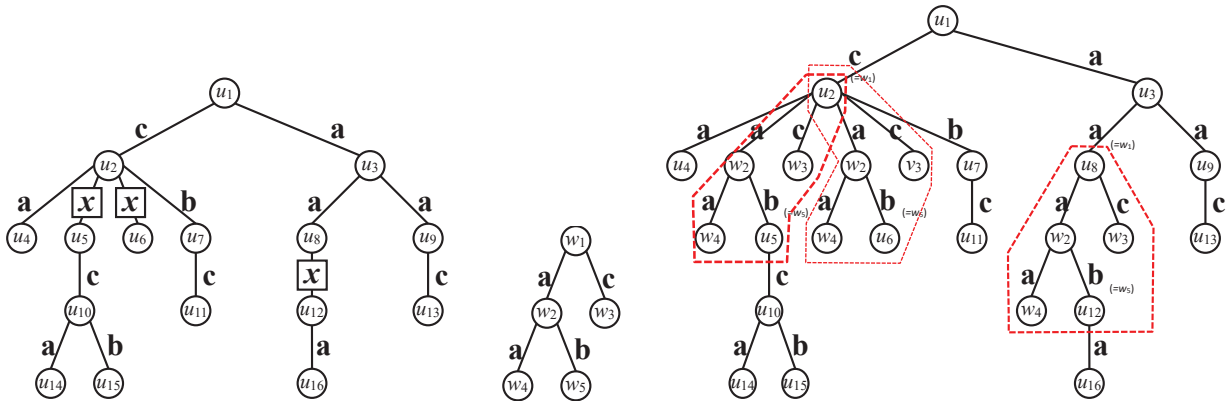

図 2 1 変数項木パターン f への代入の例．変数ラベル x を持つ変数に対し順序木 g を代入し，順序木 $f\{x := [g, [w_1, w_5]]\}$ を得る．順序木 $f\{x := [g, [w_1, w_5]]\}$ 中の順序木 g に該当する部分を破線の枠で示す．

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，つまり変数を持たない項木パターンを Λ **上の順序木** (または単に**順序木**) と呼ぶ．

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．また $g_1, \ldots, g_n$ 中の変数は，変数ラベルとして $x_1, \ldots, x_n$ を持たない．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．

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 1 変数項木パターン，または同一の順序木が代入される．f, g を 1 変数項木パターン，f 中の全ての変数は変数ラベル x を持ち，g 中の全ての変数は変数ラベル y を持つとする（x と y は互いに異なる変数ラベル）．σ を g の根と 1 つの葉からなる頂点のリストとする．代入 $\theta = \{x := [g, \sigma]\}$ を f に適用して得られる項木パターン $f\theta$ は，$f\theta$ 中の全ての変数が変数ラベル y を持つような 1 変数項木パターンである．1 変数項木パターンに対する代入の例を図 2 に示す．

T 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．T の部分木 S と，S の頂点の組 (u, v) が次の 3 つの条件を満たすとき，S を (u, v) **に関する** T **の 2-port 対応部分木**（または単に T **の 2-port 対応部分木**）とよび，$S(u, v)$ と表す．(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

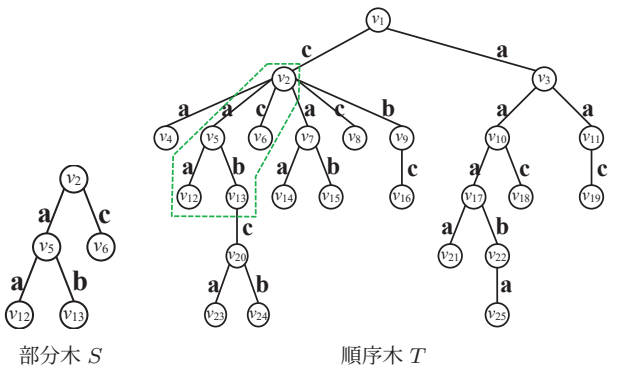

図 3 2-port 対応部分木の例．部分木 S は (v_2, v_{13}) に関する T の 2-port 対応部分木である．

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，(u, v) に関する 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 に対し，(u, v) に関する 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．2-port 対応部分木の例を図 3 に示す．

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 変数項木パターンと順序木に対するマッチング問題を次のように定義する．

1 変数項木パターンと順序木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
問題： t と T がマッチするか否かを判断する

舛井らは，1 変数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を $O(N^3)$ 時間で解くマッチングアルゴリズム OneMatching を提案した [6]．本節では，舛井らのマッチングアルゴリズムを改良し，時間計算量を $O(N^2)$ に削減した新しいマッチングアルゴリズム OneMatchingS の提案を行う．

提案したマッチングアルゴリズム OneMatchingS 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 と頂点数 N の順序木 T が与えられたときのアルゴリズム OneMatchingS の動作を説明する．OneMatchingS は順序木 T の 2-port 対応部分木を列挙する．この時に 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 中の変数の位置や個数に基づき高々 N 個の 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，その時点で列挙を終了し OneMatchingS は “yes” を返す．同型になる 2-port 対応部分木が存在しなければ，OneMatchingS は “no” を返す．同型になるか否かを確かめるのに $O(N)$ 時間かかるため，OneMatchingS は，1 変数項木パターンに対するマッチング問題を $O(N^2)$ 時間で計算する．

1 変数項木パターン t（または順序木 T）の頂点 v に対し，v の子の数を $ch(t, v)$（または $ch(T, v)$）で表す．1 変数項木パターン t の頂点 v に対し，v を親ポートとする変数の数を $va(t, v)$ で表す．また，本節では 1 変数項木パターン中の変数ラベルを全て x とする．

補題 1 [6] 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．

補題 2 t を 1 変数項木パターン，T を順序木とする．幅優先探索で最初に見つかる t の変数を $[u', v']$ とする．t と T がマッチするならば，$t\{x := [S(u, v), [u, v]]\} \cong T$ となる T の 2-port 対応部分木 $S(u, v)$ の中に次の 2 つの条件を満たすものが存在する．
(1) $S(u, v)$ の根 u の T 中での幅優先探索順と，t の u' の幅優先探索順が等しい．
(2) $S(u, v)$ の根 u の T における全ての子を $c_1, \ldots, c_k$ とする．このとき $S(u, v)$ における u の子 $c_i, \ldots, c_j$ $(1 \leq i \leq j \leq k)$ とすると，i, j は次の条件を満たす．(2-a) v' は u' の i 番目の子である．(2-b) $S(u, v)$ の根の子の数 $ch(S(u, v), u) = j - i + 1$ に対して，$ch(S(u, v), u) = (ch(T, u) - ch(t, u'))/va(t, u') + 1$ が成り立つ．

証明： 補題 1 より，t と T がマッチするので，T のある頂点の組 (u, v) に対し，$t\{x := [S(u, v), [u, v]]\} \cong T$ を満たす T の 2-port 対応部分木 $S(u, v)$ が存在する．幅優先探索で最初に見つかる t の変数を $[u', v']$ とする．このとき，$[u', v']$ に代入される 2-port 対応部分木 $S(u, v)$ について考える．代入の定義より，$S(u, v)$ の根 u は u' と同一視され，v は v' と同一視される．t の u' の幅優先探索順を b とする．T の頂点 p を，幅優先探索順で b 番目の頂点とする．根から u' までの間に代入が行われないため，$S(u, v)$ の根 u は p と同一視される．つまり，$S(u, v)$ の根の T 中での幅優先探索順と，t の u' の幅優先探索順が等しい．u の T における全ての子を $c_1, \ldots, c_k$ とする．u' の子における順番で v' の順番を i $(1 \leq i)$ 番目とする．代入の定義より，$c_1, \ldots, c_{i-1}$ までの子は代入前の t と代入後の T で変化しない．よって，$S(u, v)$ の u の子は $c_i, \ldots, c_j$ $(1 \leq i \leq j \leq k)$ となる．また，$t\{x := [S(u, v), [u, v]]\} \cong T$ なので $ch(t\{x := [S(u, v), [u, v]]\}, u') = ch(T, u)$ が成り立つ．t の u' の子の数は $ch(t, u')$ であり，u' を親ポートとする変数の数は $va(t, u')$ なので，$ch(t\{x := [S(u, v), [u, v]]\}, u') = ch(t, u') - va(t, u') + va(t, u') \times ch(S(u, v), u)) = ch(T, u)$ が成り立つ．よっ

アルゴリズム 1 OneMatchingS

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”

```
if (N − n) が m で割り切れない then
    return “no”
end if
u′ を幅優先探索で最初に到達する t の変数の親ポート，b をその順番とする
u を幅優先探索で b 番目の T の頂点とする
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
w := (ch(T, u) − ch(t, u′))/va(t, u′) + 1
if w が非負整数でない then
    return “no”
end if
subT を u と，u の i 番目から (i + w − 1) 番目までの子とその子孫全体からなる T の部分木とする
for each subT の頂点 v do
    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）
    s を S の頂点数とする
    if N = (n + (s − 2) × m) then
        if t{x := [S, [u, v]]} ≅ T then
            return “yes”
        end if
    end if
end for
return “no”
```

て $ch(S(u,v),u) = (ch(T,u) - ch(t,u'))/va(t,u') + 1$ である．以上より，$t\{x := [S(u,v),[u,v]]\} \cong T$ となる T の 2-port 対応部分木 $S(u,v)$ の中に条件 (1),(2) を満たすものが存在する．□

また補題 2 より，$(ch(T,u) - ch(t,u'))/va(t,u') + 1$ が非負整数でないとき，補題 2 の条件 (2) を満たす T の 2-port 対応部分木が存在しないため，t と T はマッチしない．

以上より，次の定理が示せる．

定理 1 アルゴリズム OneMatchingS は，1 変数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を $O(N^2)$ 時間で正しく解く．

証明： 補題 2 の条件 (1),(2) を満たすような T の 2-port 部分木 $S(u,v)$ 中で $t\{x := [S(u,v),[u,v]]\} \cong T$ を満たす 2-port 部分木が存在しないならば，t と T はマッチしない．アルゴリズム OneMatchingS は補題 2 の条件 (1),(2) を満たす T の 2-port 部分木 $S(u,v)$ をすべて列挙し，変数に代入し，入力順序木と同型になるかを確かめる．よって OneMathingS はマッチング問題を正しく解く．

T の頂点数を N とする．補題 2 の条件 (1),(2) を満たす 2-port 部分木は高々 N 個しかなく，列挙にかかる時間は $O(N)$ 時間である．変数に代入後の木が同型になるかどうか確かめるのにかかる時間は高々 $O(N)$ 時間である．よって，アルゴリズム OneMatchingS は，1 変数項木パターンに対するマッチング問題を $O(N^2)$ 時間で正しく解く．□

4. 評価実験

3 節で提案した, 1 変数項木パターンと順序木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 OneMatchingS 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:16.0GB，CPU:Intel(R) Core i7 2.90GHz，OS:MacOS Mojave の計算機上に開発環境:eclipse，開発言語:Java を用いてアルゴリズム OneMatchingS 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，OneMatchingS の入力として与え，その実行時間を計測した．実験データの作成方法は以下のとおりである．(1) ランダムに頂点数 n の 1 変数項木パターン t を作成する．(2) t の変数にランダムに作成した順序木を代入することで，t とマッチする頂点数 N の順序木 T を作成する．(3) 変数ラベルを除いて t と同型な項木パターン t' を作成し（t' は 1 変数項木パターンでない），t' の変数にランダムに作成した順序木を代入し，t とマッチしない頂点数 N の順序木 T' を作成する．(1),(2) の方法で作成した 1 変数項木パターンと順序木の組 (t,T) をマッチする場合の入力，(1),(3) の方法で作成した 1 変数項木パターンと順序木の組 (t,T') をマッチしない場合の入力として実験を行った．

項木パターンの頂点数 n と順序木の頂点数 N に関して，以下の 2 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 1000 組作成し，OneMatching[6] と本論文で提案した OneMatchingS の平均実行時間を計測し，実行時間の比較を行った．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 1000 組作成し，OneMatchingS 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 4,5,6,7 に示す．実験 (1) の結果より，順序木の頂点数が増加すると OneMatchingS 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．また舛井らの OneMatching よりも高速であることが確認できる．OneMatching に対して，今回のアルゴリズムは入力順序木の最大次数分の一程度の時間に高速化できる．頂点数 N の順序木であれば，次数は最大で $N-1$ であるが，今回の実験データはランダムに作成しているため，それほどの削減が行われなかったと考えられる．

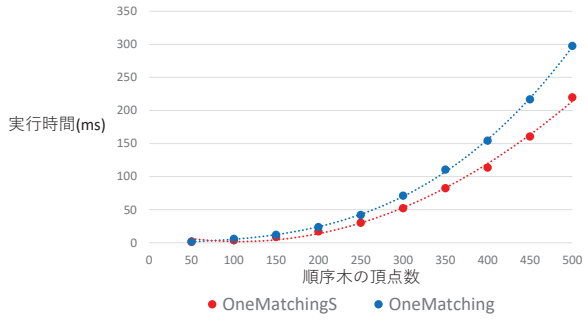

図 4 実験 (1) 項木パターンの頂点数が 10，項木パターンと順序木がマッチする場合の順序木の頂点数に対する OneMatchingS と OneMatching の平均実行時間の比較

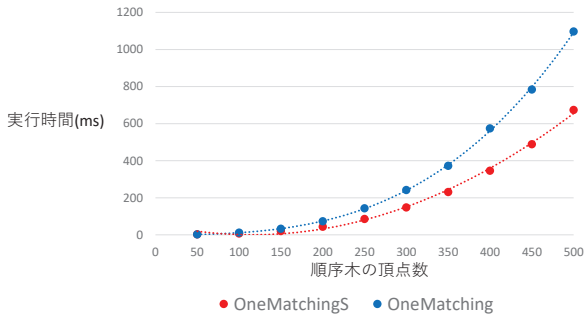

図 5 実験 (1) 項木パターンの頂点数が 10，項木パターンと順序木がマッチしない場合の順序木の頂点数に対する OneMatchingS と OneMatching の平均実行時間の比較

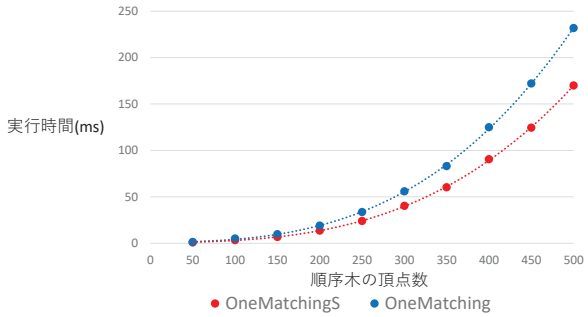

図 6 実験 (1) 項木パターンの頂点数が 20，項木パターンと順序木がマッチする場合の順序木の頂点数に対する OneMatchingS と OneMatching の平均実行時間の比較

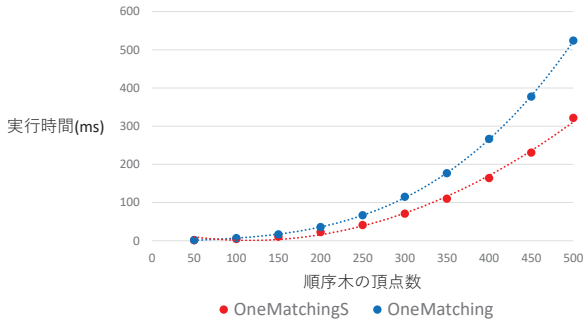

図 7 実験 (1) 項木パターンの頂点数が 20，項木パターンと順序木がマッチしない場合の順序木の頂点数に対する OneMatchingS と OneMatching の平均実行時間の比較

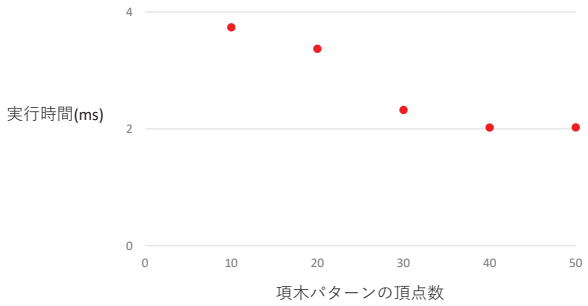

図 8 実験 (2) 順序木の頂点数 100，項木パターンと順序木がマッチする場合の項木パターンの頂点数に対する OneMatchingS の平均実行時間の変化

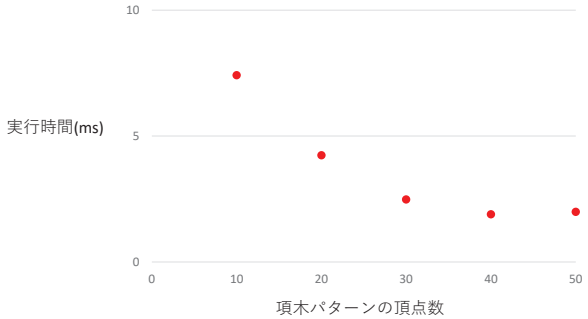

図 9 実験 (2) 順序木の頂点数 100，項木パターンと順序木がマッチしない場合の項木パターンの頂点数に対する OneMatchingS の平均実行時間の変化

実験 (2) の結果を図 8,9 に示す．実験 (2) の結果より，1 変数項木パターンの頂点数が増加すると，OneMatchingS の実行時間が減少することが確認できる．舛井ら [6] の OneMatching の実験結果においても，1 変数項木パターンの頂点数が増加すると同様に実行時間が減少している．1 変数項木パターンの頂点数が増加すると，変数に代入される 2-port 対応部分木の頂点数が小さくなる．これにより列挙される 2-port 対応部分木の数が減少するため，実行時間が減少すると考えられる．

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，2-port 対応部分木の列挙の途中でマッチする 2-port 対応部分木が出現すると，その時点で列挙が終了する．しかし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，舛井ら [6] が提案した 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時間計算量の改良を行った．舛井らは，1 変数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を $O(N^3)$ 時間で解くマッチングアルゴリズムを提案した．本研究では，このアルゴリズムを改良し，時間計算量を $O(N^2)$ に削減した新

しいマッチングアルゴリズムの提案を行った．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．実験結果より，提案アルゴリズムが舛井らのアルゴリズムよりも高速であることが確認できた．

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．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 [5]．入力順序木中の 2-port 対応部分木を列挙し，その 2-port 対応部分木を変数に代入して同型であるか確かめる提案アルゴリズムでは，大幅な計算量の削減は難しいと考えられる．そのため，異なるアプローチによるマッチングアルゴリズムの提案が高速化において重要と考えられる．

項木パターン中の変数ラベルの数が高々 k 個 (k は定数) である項木パターンを k 変数項木パターンとよぶ．本研究の発展としては，k 変数項木パターンと順序木のマッチング問題を解くマッチングアルゴリズムの開発が考えられる．また，1 変数項木パターンの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あげられる．

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, Inductive inference of formal languages from positive data, Information and Control, 45: 117–135, 1980.
[2] D. Angluin, Finding patterns common to a set of string, Journal of Computer and System Sciences, 21: 46–62, 1980.
[3] 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.
[4] 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.
[5] Y. Suzuki, T. Shoudai, T. Uchida,T. Miyahara, An Efficient Pattern Matching Algorithm for Ordered Term Tree Patterns, IEICE Trans. Fundamentals, Vol. E98-A(6): 1197–1211, 2015.
[6] 舛井 里帆, 池森 千尋, 鈴木 祐介, 内田 智之, 宮原 哲浩, 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, 火の国情報シンポジウム 2020, C4-2, 2020.