非線形項木パターンに対するマッチングアルゴリズムと頻出1変数項木パターン枚挙への応用

片山 悠[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．項木パターン中の全ての変数が異なるとき線形項木パターンといい，線形でない項木パターンを非線形項木パターンという．項木パターン中の全ての変数が同一のとき1変数項木パターンという．本研究では，項木パターン中に複数種類の変数が存在するがその中の1種類の変数の繰り返しを許す非線形項木パターンのマッチングアルゴリズムを提案する．さらに，提案したマッチングアルゴリズムを用いて，頻出1変数項木パターンの枚挙アルゴリズムの高速化を行う．

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 Matching Algorithm for Non-Linear Term Tree Patterns and its Application for Enumeration of Frequent One-Variable Term Tree Patterns

Yu Katayama[1]　Yusuke Suzuki[2,a)]　Tomoyuki Uchida[2,b)]　Tetsuhiro Miyahara[2,c)]

Abstract: Ordered trees are rooted trees with edge labels and ordered children. Term tree patterns are rooted ordered tree patterns having structured variables. A variable can be replaced with any rooted ordered tree. A term tree pattern is linear if all variables in the term tree pattern have mutually distinct variable labels. A term tree pattern is non-linear if the term tree pattern is not linear. A one-variable term tree pattern is a term tree pattern all of whose variable labels are the same. In this paper, we propose a polynomial time matching algorithm for a kind of non-linear term tree patterns, and propose an enumeration algorithm for frequent one-variable term tree patterns using proposed matching algorithm.

Keywords: graph algorithm, machine learning, data mining

1. はじめに

頻出アイテム集合発見問題は，データマイニングにおける基本的な問題である．頻出アイテム集合発見問題をグラフや木構造データに拡張し，グラフのデータベースに対して頻出なグラフ構造を発見する問題とそれを解くアルゴリズムが数多く提案されている [3], [6], [10]．本研究では，木構造データに頻出な木構造パターンを発見する問題を解く枚挙アルゴリズムの高速化を行う．

本研究で扱う木構造データは，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よぶ．順序木に共通する木構造の表現方法として，**項木パターン**を用いる [8]．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

[1] 広島市立大学情報科学部
Faculty of Information Sciences, Hiroshima City University, Japan
[2] 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University, Japan
a) y-suzuki@hiroshima-cu.ac.jp
b) uchida@hiroshima-cu.ac.jp
c) miyares23@hiroshima-cu.ac.jp

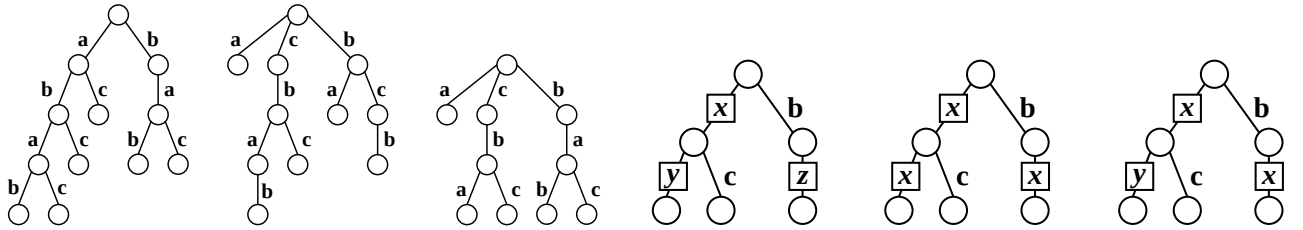

図 1 項木パターンと頻出項木パターン枚挙問題の例

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よぶ．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 は順序木 T とマッチするという．項木パターン中の全て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線形な項木パターン，または線形項木パターンとよぶ．線形でない項木パターンを非線形項木パターンとよぶ．

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．頻出項木パターン枚挙問題とは，与えられた順序木のデータベースと，閾値 σ に対して，データベース中の順序木でマッチするものの割合が σ 以上である項木パターンを全て枚挙する問題である．宮原ら [6] はタグ木パターンという線形項木パターンを拡張したパターンを提案し，頻出線形項木パターン枚挙問題を解く枚挙アルゴリズムを提案した．

Angluin[1] は正例から推論可能な言語族としてパターン言語を導入した．さらに，パターン中に 1 種類の変数しか現れない 1 変数パターンを提案した [2]．**1 変数項木パターン**とは，1 変数パターンの概念を項木パターンに拡張したもので，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．舛井ら [5] 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．酒井ら [7] は，舛井らの研究を発展させ $O(N^2)$ 時間のマッチングアルゴリズムを提案した．また，田中ら [10] は，宮原ら [6] の枚挙アルゴリズムを発展させ，頻出 1 変数項木パターン枚挙問題を解く枚挙アルゴリズムの提案と実装を行った．

変数の種類や出現回数に制限がない非線形項木パターンと順序木に対するマッチング問題は NP 完全な問題と知られている [9]．しかし非線形な項木パターンであっても，1 変数項木パターンのような変数ラベルの種類数に制限があるような項木パターンと順序木に対するマッチング問題は多項式時間で計算可能である．本研究では，項木パターン中に複数の変数ラベルの変数が存在するが，ある 1 種類の変数ラベルを持つ変数のみ繰り返し出現することを許す非線形な項木パターンを考える．このような項木パターンを**1rep項木パターン**（または，**1 種類の変数ラベルの繰り返しを許す非線形項木パターン**）とよぶ．

本論文で扱う 1rep項木パターンと順序木に対するマッチング問題とは，1rep項木パターン t と順序木 T が与えられたときに，t と T がマッチするか判定する問題である．本論文では，1rep項木パターンと順序木に対するマッチング問題を多項式時間で解くマッチングアルゴリズムを提案する．

さらに，田中ら [10] の提案した頻出 1 変数項木パターンの枚挙アルゴリズムを発展させ，提案したマッチングアルゴリズムを用いることで田中らの手法より高速な頻出 1 変数項木パターンの枚挙アルゴリズムの提案を行う．提案したマッチングアルゴリズムおよび枚挙アルゴリズムを計算機上に実装し，評価実験を行った結果を報告する．

図 1 に線形項木パターン，1 変数項木パターン，1rep項木パターンと頻出項木パターン枚挙問題の例を示す．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．図では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．順序木集合 $\{T_1, T_2, T_3\}$ を順序木のデータベースとする．線形項木パターン t_1 は順序木 T_1, T_2, T_3 とマッチし，1 変数項木パターン t_2 は順序木 T_1 とマッチし，1rep項木パターン t_3 は順序木 T_1, T_2 とマッチする．このとき閾値 $\sigma = 0.5$ とすると，t_1, t_3 は $\{T_1, T_2, T_3\}$ に対して頻出な項木パターンである．

2. 準備

本節では，項木パターン，線形項木パターン [8] および 1 変数項木パターン [5] 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．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，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**とよぶ．

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，下記の (1)–(4) の 4 つの条件を満たす全単射 $\varphi : V_f \to V_g$ が存在するときをいう．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．(3)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 $[\varphi(u), y, \varphi(v)] \in H_g$, $[\varphi(u'), y', \varphi(v')] \in H_g$ に対し，$y \neq y'$ $(y, y' \in X)$ である．(4)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．また上記の条件 (4) と次の条件 (5) を満たす全単射 $\varphi : V_f \to V_g$ が存在するとき，f と g が辺と変数ラベルを無視して同型であるという．(5)$(u, a, v) \in E_f$ または $[u, x, v] \in H_f$ のとき，その時に限り $(\varphi(u), b, \varphi(v)) \in E_g$ または $[\varphi(u), y, \varphi(v)] \in H_g$ 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 $\theta = \{x_1 := [g_1, \sigma_1], \ldots, x_n := [g_n, \sigma_n]\}$ を代入とよぶ．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 に全て適用して得られる新しい項木パターンを $f\theta$ と書く．

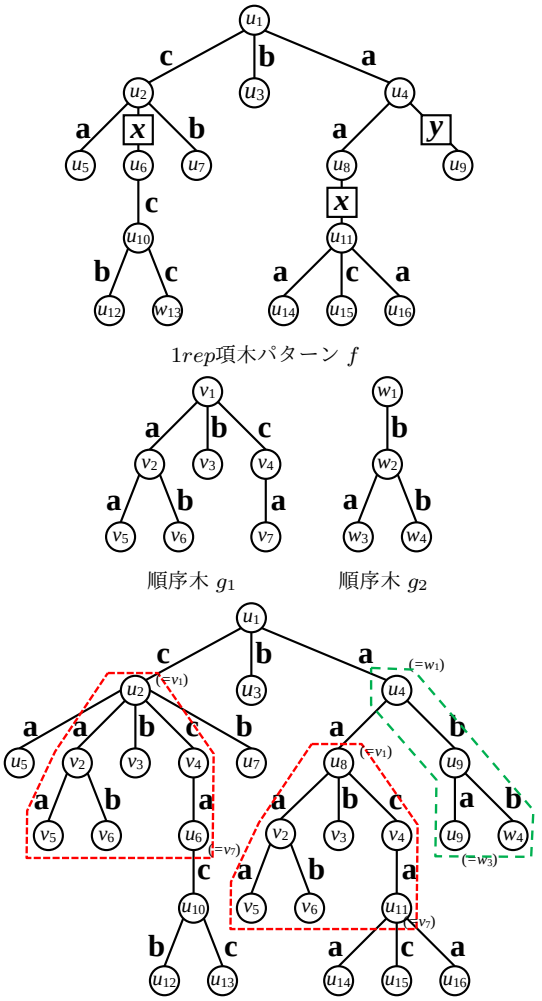

図 2　1*rep*項木パターン f への代入と 2-port 対応部分木の例．

定義 2　$\Lambda \cup X$ 上の項木パターン t に対し，t の項木パターン言語を $L(t) = \{$ 順序木 $T \mid$ ある代入 θ に対し $T \cong t\theta\}$ と定義する．

順序木 T と項木パターン t に対し，$T \in L(t)$ ならば t は T とマッチするという．

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 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．

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線形な項木パターン，または**線形項木パターン**とよぶ．線形でない項木パターンを**非線形項木パターン**とよぶ．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**とよぶ．本論文では，項木パターン中に複数の変数ラベルの変数が存在するが，ある 1 種類の変数ラベルを持つ変数のみ繰り返し出現することを許す非線形な項木パターンを考える．このような項木パターンを **1*rep*項木パターン**（または，**1 種類の変数ラベルの繰り返しを許す非線形項木パターン**）とよぶ．1*rep*項木パターンのクラスは線形項木パターンのクラスおよび 1 変数項木パターンのクラスを包含する．本論文では，簡単のために 1*rep*項木パターンと 1 変数項木パターンにおいて変数ラベル x を持つ変数のみ繰り返し出現するものとする．

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．1*rep*項木パターン f の変数ラベル x を持つ変数に対し順序木 g_1 を，変数ラベル y を持つ変数に対し順序木 g_2 を代入し，順序木 $f\{x := [g_1,[v_1,v_7]], y := [g_2,[w_1,w_3]]\}$ を得る．順序木 $f\{x := [g_1,[v_1,v_7]], y := [g_2,[w_1,w_3]]\}$ 中の 2-port 対応部分木 $g_1(v_1,v_7)$ を赤色の破線の枠で，2-port 対応部分木 $g_2(w_1,w_3)$ を緑色の破線の枠で示す．

$D = \{T_1, T_2, \ldots, T_m\}$ を空でない順序木の有限集合，t を項木パターンとする．$match_D(t)$ を，t とマッチする $T_i \in D$ $(1 \leq i \leq m)$ の個数と定義し，D に対する t の支持率 $supp_D(t)$ を $supp_D(t) = match_D(t)/m$ と定義する．実数 σ $(0 < \sigma \leq 1)$ に対して，$supp_D(t) \geq \sigma$ ならば，項木パターン t は D に対して **σ 頻出**であるという．実数 σ は，頻出と判断するために閾値としてユーザが与える値であり，最小支持率とよばれる．

3. 1*rep*項木パターンのマッチングアルゴリズム

1*rep*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*rep*項木パターンに対するマッチング問題
入力： 1*rep*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定せよ

Suzuki ら [9] は頂点数 n の線形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 $O(nN)$ 時間で解くマッチングアルゴリズム Matching を提案した．舛井ら [5] は頂点数 n の 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．本論文では，これらのアルゴリズムをもとに，1*rep*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 xMatching1，xMatching2 を提案する．

提案したマッチングアルゴリズム xMatching1 を Algorithm 1 に示す．入力として頂点数 n の 1*rep*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 xMatching1 の動作を説明する．x を 1*rep*項木パターン中に繰り返し出現することを許す変数ラベルとする．xMatching1 は順序木 T の全ての 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 の変数ラベル x の変数に代入し，線形項木パターン t' を得る．そして t' と T がマッチするか否かをアルゴリズム Macthing を用いて判定する．もし t' と T がマッチするような 2-port 対応部分木が存在すれば，その時点で 2-port 対応部分木の列挙を終了し，xMatching1 は “yes” を返す．全ての 2-port 対応部分木を列挙しても t' と T がマッチしなければ，xMatching1 は “no” を返す．順序木 T の全ての 2-port 対応部分木は $O(N^4)$ 個存在する．t' と T がマッチするか否かを判定するのに $O(N^2)$ 時間かかるため，xMatching1 は 1*rep*項木パターンと順序木に対するマッチング問題を $O(N^6)$ 時間で計算する．

t を 1*rep*項木パターンとする．t を幅優先探索したときに t 中の変数で変数ラベル x を持つ変数が最初に出現するならば，そのような t を ***bfs*1*rep*項木パターン** とよぶ．*bfs*1*rep*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 xMatching2 を Algorithm 2 に示す．xMatching2 は xMatching1 よりも高速にマッチング問題を解くことが可能である．頂点数 n の *bfs*1*rep*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 xMatching2 の動作を説明する．xMatching2 は xMatching1 と同様に，順序木 T の 2-port 対応部分木を列挙し，Matching を用いてマッチするか否かを判定する．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，t 中の変数ラベル x の変数の出現位置に基づき高々 N^2 個の 2-port 対応部分木を列挙する．よって xMathing2 は *bfs*1*rep*項木パターンと順序木に対するマッチング問題を $O(N^4)$ 時間で計算する．

補題 1 t を項木パターン，T を順序木とする．$x_1, \ldots, x_n$ を t 中に出現する変数ラベルとする．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，$t\theta = t\{x_1 := [S_1(u_1,v_1),[u_1,v_1]], \ldots, x_n :=$

Algorithm 1 XMATCHING1(t, T)

Input: 1rep項木パターン t, 順序木 T
Output: "yes" or "no"

```
for each T の頂点 u do
  c_1, ..., c_k を u の全ての子とする
  for i := 1 to k do
    for j := i to k do
      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
      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）
        t' := t{x := [S, [u, v]]} とする
        if MATCHING(t', T)[9] = "yes" then
          return "yes"
        end if
      end for
    end for
  end for
end for
return "no"
```

Algorithm 2 XMATCHING2(t, T)

Input: bfs1rep項木パターン t, 順序木 T
Output: "yes" or "no"

```
h を幅優先探索して最初に到達する t の変数ラベル x の変数，u' を変数 h の親ポート，f をその順番とする
u を幅優先探索で f 番目の T の頂点とする
c_1, ..., c_k を u の全ての子とする
i を u の子の順番で変数 h の子ポートの順番とする
for j := i to k do
  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
  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）
    t' := t{x := [S, [u, v]]} とする
    if MATCHING(t', T)[9] = "yes" then
      return "yes"
    end if
  end for
end for
return "no"
```

$[S_n(u_n, v_n), [u_n, v_n]]\} \cong T$ を満たす T の 2-port 対応部分木の列 $S_1(u_1, v_1), \ldots, S_n(u_n, v_n)$ が存在する．

証明： $(\Longrightarrow)$ t と T がマッチするので，マッチの定義より，$t\{x_1 := [Q_1, [w_{11}, w_{12}]], \ldots, x_n := [Q_n, [w_{n1}, w_{n2}]]\} \cong T$ を満たす順序木の列 $Q_1, \ldots, Q_n$ が存在する．ここで w_{i1} は Q_i の根，w_{i2} は Q_i のある葉である $(1 \leq i \leq n)$．代入の定義より，Q_i 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_i と同型な T の部分木 S_i が存在する．V_{Q_i} を Q_i の頂点集合，V_{S_i} を S_i の頂点集合とし，ξ_i を V_{Q_i} から V_{S_i} への同型写像とする．w_{i1}, w_{i2} 以外の Q_i の頂点 w は，代入の前後で次数が変化しないため，S_i の頂点 $\xi_i(w) \in V_{S_i} \setminus \{\xi_i(w_{i1}), \xi_i(w_{i2})\}$ の S_i 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_i 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_i 上で $\xi_i(c_1)$ のすぐ隣の兄弟が $\xi_i(c_2)$ であるなら，T 上でも $\xi_i(c_1)$ のすぐ隣の兄弟が $\xi_i(c_2)$ である．よって部分木 S_i と頂点の組 $(\xi_i(w_{i1}), \xi_i(w_{i2}))$ は T の 2-port 対応部分木である．このとき，$t\{x_1 := [Q_1, [w_{11}, w_{12}]], \ldots, x_n := [Q_n, [w_{n1}, w_{n2}]]\} \cong T$ なので，$t\theta = t\{x_1 := [S_1(\xi_1(w_{11}), \xi_1(w_{12})), [\xi_1(w_{11}), \xi_1(w_{12})]], \ldots, x_n := [S_n(\xi_n(w_{n1}), \xi_n(w_{n2})), [\xi_n(w_{n1}), \xi_n(w_{n2})]]\} \cong T$ が成り立つ．

$(\Longleftarrow)$ マッチの定義より，T のある 2-port 対応部分木の列 $S_1(u_1, v_1), \ldots, S_n(u_n, v_n)$ に対し，$t\theta = t\{x_1 := [S_1(u_1, v_1), [u_1, v_1]], \ldots, x_n := [S_n(u_n, v_n), [u_n, v_n]]\} \cong T$ を満たすならば，t と T がマッチする． □

定理 1 アルゴリズム XMATHING1 は 1rep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 $O(N^6)$ 時間で正しく解く．

証明： t と T がマッチするので，$t\theta \cong T$ を満たす代入 θ が存在する．ここで補題 1 より代入 $\theta = \{x := [S, [u, v]], x_1 := [S_1(u_1, v_1), [u_1, v_1]], \ldots, x_n := [S_n(u_n, v_n), [u_n, v_n]]\}$ とする．x を 1rep項木パターンに繰り返し出現する変数ラベルとする．代入 θ を変数ラベル x に対する束縛からなる代入 $\theta' = \{x := [S, [u, v]\}$ とそれ以外の変数ラベルに対する束縛からなる代入 $\theta'' = \{x_1 := [S_1(u_1, v_1), [u_1, v_1]], \ldots, x_n := [S_n(u_n, v_n), [u_n, v_n]]\}$ にわけて考える．このとき $t\theta = t(\theta'\theta'') = (t\theta')\theta'' \cong T$ である．θ' は変数ラベル x に対する束縛からなる代入なので，$t\theta'$ は線形項木パターンである．線形項木パターン $t\theta'$ と順序木 T に対して，$(t\theta')\theta'' \cong T$ となる代入 θ'' が存在するか否か，つまり $t\theta'$ と T がマッチするか否かはアルゴリズム MATCHING$(t\theta', T)$ によって判定できる．これにより，x 以外の変数ラベルに対して補題 1 の条件を満たす 2-port 対応部分木の列が存在するか否かを判定できる．よって T の全ての 2-port 対応部分木 $S(u, v)$ に対して，代入 $\theta' = \{x := [S, [u, v]\}$ を作成し，$t\theta'$ と T がマッチするか判定すれば，全ての変数ラベルに対して，補題 1 の条件を満たすような 2-port 対応部分木の列が存在するか否かを確かめることができる．よって，アルゴリズム XMATHING1 は 1rep項木パターンと順序木に対するマッチング問題を正しく解く．

T の 2-port 対応部分木は高々 N^4 個であり，列挙にかかる時間は $O(N^4)$ 時間である．変数ラベル x の変数に代入後の線形項木パターン $t\theta'$ の最大サイズは N である．よって，線形項木パターン $t\theta'$ と順序木 T がマッチするか否かの判定にかかる時間は高々 $O(N^2)$ 時間である．これより，アルゴリズム XMATCHING1 は，1rep項木パターンに対するマッチング問題を $O(N^6)$ 時間で正しく解く． □

舛井ら [5] の定理 1 より，T の 2-port 対応部分木 $S(u, v)$ の根 u と，$S(u, v)$ において u の子の先頭となる頂点の 2 つを一意に決めたとき，$S(u, v)$ の数は高々 N^2 個しかない．

よって，定理 1 より次の定理が示せる．

定理 2 アルゴリズム xMathing2 は $bfs1rep$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 $O(N^4)$ 時間で正しく解く．

4. 頻出 1 変数項木パターンの枚挙アルゴリズムの高速化

頻出 1 変数項木パターン枚挙問題は次のような問題である [10]．

頻出 1 変数項木パターン枚挙問題

入力：空でない順序木の有限集合 D，最小支持率 σ $(0 < \sigma \leq 1)$

問題：D に対して σ 頻出な 1 変数項木パターンを全て枚挙せよ

D を空でない順序木の有限集合，実数 σ $(0 < \sigma \leq 1)$ を最小支持率とする．田中ら [10] は，D に対して σ 頻出な 1 変数項木パターンを全て枚挙するアルゴリズム Gen-F1TTP を提案した．本論文では，1 変数項木パターンの枚挙アルゴリズム Gen-F1TTP を発展させ，D に対して σ 頻出な 1 変数項木パターンを高速に全て枚挙するアルゴリズム Gen-F1TTPX(Algorithm 3) を提案する．アルゴリズム Gen-F1TTPX の動作を説明する．手続き enumFreq はアルゴリズム Gen-F1TTP[10] 中で用いられた手続きと同一である．手続き enumFreq は，順序木の列挙手法である最右拡張 [3], [11] を用いて変数のみからなる線形項木パターンを枚挙する．これにより，D に対して σ 頻出な変数のみからなる線形項木パターンを全て枚挙する．本研究では，アルゴリズム Gen-F1TTP の replaceEdge と replaceOneVariable をもとに，手続き replaceEdgeX(Procedure 4) を提案する．x を $1rep$ 項木パターンと 1 変数項木パターンに繰り返し出現する変数ラベルとする．手続き replaceEdgeX は，手続き enumFreq の出力である変数のみからなる線形項木パターンに対して，brute-force method を用いて各変数に辺または変数ラベル x の変数を代入し，D に対して σ 頻出となるか調べていく．この手続きの途中で作成される項木パターンは $bfs1rep$項木パターンであるため，頻出か否かを判定するために前節で提案した xMatching2 を用いる．

田中ら [10] の補題 1 より，次の補題が示せる．

補題 2 項木パターン f, g に対し，$f \cong g\theta$ となる代入 θ が存在するならば $L(f) \subseteq L(g)$ である．

定理 3 アルゴリズム Gen-F1TTPX は，D に対して σ 頻出な 1 変数項木パターンを過不足なく枚挙する．

証明： アルゴリズム Gen-F1TTPX の手続き enumFreq は変数のみからなる σ 頻出な線形項木パターンを過不足なく枚挙する [10]．g を手続き enumFreq の出力である変数のみからなる線形項木パターンの一つとする．手続き replaceEdgeX は，g に対し，brute-force method を

Algorithm 3 Gen-F1TTPX

Input: 順序木の集合 $\mathcal{D}$，最小支持率 σ $(0 < \sigma \leq 1)$
Output: D に対して σ 頻出な 1 変数項木パターンの全集合 $\Pi(\sigma)$

1: $\Pi_1(\sigma)$:=enumFreq$(\mathcal{D}, \sigma)$ [10]
2: $\Pi(\sigma)$:=replaceEdgeX$(\mathcal{D}, \sigma, \Pi_1(\sigma))$ (Procedure 4)
3: **return** $\Pi(\sigma)$

Procedure 4 replaceEdgeX

Input: 順序木の集合 $\mathcal{D}$，最小支持率 σ $(0 < \sigma \leq 1)$，変数のみからなる線形項木パターンの集合 Π_{in}
Output: 1 変数項木パターンの集合 Π_{out}

1: $\Pi_{out} := \emptyset$
2: $pos := 1$
3: **for** $\pi \in \Pi_{in}$ **do**
4: $\quad \Pi_{out} := \Pi_{out} \cup$ replaceEdgeXSub$(\mathcal{D}, \sigma, \pi, pos)$ (Procedure 5)
5: **end for**
6: **return** Π_{out}

Procedure 5 replaceEdgeXSub

Input: 順序木の集合 $\mathcal{D}$，最小支持率 σ $(0 < \sigma \leq 1)$，$1rep$項木パターン π，正数 pos
Output: $1rep$項木パターンの集合 Π_{out}

1: **if** $pos > |E_\pi \cup H_\pi|$ **then**
2: $\quad$ **return** $\{\pi\}$
3: **end if**
4: $\Pi_{tmp} := \emptyset$
5: 変数 h を π の辺と変数の BFS 順で pos 番目の位置にある変数とする
6: **for** D に頻出な辺ラベル k **do**
7: $\quad \pi_k$ を π の変数 h を k でラベル付けされた辺で置き換えた項木パターンとする
8: $\quad$ **if** π_k が D に対して σ 頻出である **then**
9: $\quad\quad \Pi_{tmp} := \Pi_{tmp} \cup \{\pi_k\}$
10: $\quad$ **end if**
11: **end for**
12: π_x を π の変数 h を変数ラベル x の変数で置き換えた項木パターンとする
13: **if** π_x が D に対して σ 頻出である **then**
14: $\quad \Pi_{tmp} := \Pi_{tmp} \cup \{\pi_x\}$
15: **end if**
16: $\Pi_{out} := \emptyset$
17: **for** $\pi' \in \Pi_{tmp}$ **do**
18: $\quad \Pi_{out} := \Pi_{out} \cup$replaceEdgeXSub$(\mathcal{D}, \sigma, \pi', pos + 1)$
19: **end for**
20: **return** Π_{out}

用いて各変数に辺または変数ラベル x を持つ変数を代入する．これにより，g と辺と変数ラベルを無視して同型な全ての 1 変数項木パターンを全て枚挙する．補題 1 より，項木パターン f, g が，ある代入 θ が存在して $f \cong g\theta$ となるとき，D に対する f の支持率は g の支持率以下となる．よって，項木パターン g の D に対する支持率が最小支持率 σ を下回ったならば，g に代入をして作成される全ての項木パターンの支持率は必ず σ よりも小さくなる．この性質を用いて，手続き replaceEdgeX は，ある項木パターンが最小支持率 σ を下回ったならば，その項木パターンに代入して作成される項木パターンの作成を行わない．よって，手続き replaceEdgeX は，σ 頻出な変数のみからなる線形項木パターン g に対して，辺と変数ラベルを無視して同型な全ての σ 頻出な 1 変数項木パターンを過不足なく枚挙する．

以上より，アルゴリズム Gen-F1TTPX は，D に対して σ 頻出な 1 変数項木パターンを過不足なく枚挙する．□

5. 評価実験

前節で提案した頻出 1 変数項木パターン枚挙問題を解く枚挙アルゴリズム Gen-F1TTPX と田中ら [10] が提案した枚挙アルゴリズム Gen-F1TTP を主記憶メモリ:16.0GB, CPU:11th Gen Intel(R) Core(TM) i7-1165G7@2.80GHz, OS:Windows 10 の計算機上に開発環境:Eclipse，開発言語:Java を用いて実装した．以降，枚挙アルゴリズム Gen-F1TTPX を提案手法，枚挙アルゴリズム Gen-F1TTP を従来手法とよぶ．評価実験では，田中ら [10] の実験と同様に実データ集合 D_{leu} と人工データ集合 D_{syn} の 2 種類の順序木集合を入力として用いた．

実データ集合 D_{leu} として，KEGG-GLYCAN データベース [4] に登録されている白血病に関する糖鎖データ 177 個を順序木とみなして用いた．人工データ集合 D_{syn} として，次の (1),(2) のように作成した頂点数 30 の順序木 100 個を用いた．(1) ランダムに作成した頂点数 30 の順序木 10 個．(2)1 変数項木パターン t_1, t_2, t_3 を $L(t_3) \subseteq L(t_2) \subseteq L(t_1)$ となるように作成し，t_1, t_2, t_3 の変数にランダムに作成した順序木を代入して得られた順序木をそれぞれ 30 個ずつ．

評価実験として，入力を (1) 実データ集合 D_{leu} と (2) 人工データ集合 D_{syn} の 2 通り，最小支持率 σ を，$\sigma \in \{1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3\}$ の 8 通りに対して，提案手法 Gen-F1TTPX と従来手法 Gen-F1TTP を実行し実行時間を比較した．実データ集合 D_{leu} を入力としたときの実行時間と頻出 1 変数項木パターンの数を表 1 に示す．人工データ集合 D_{syn} を入力としたときの実行時間と頻出 1 変数項木パターンの数を表 2 に示す．

実データ集合 D_{leu}，人工データ集合 D_{syn} のいずれの入力に対しても，最小支持率が下がると提案手法 Gen-F1TTPX の実行時間が増加することが確認できる．これは従来手法 Gen-F1TTP と同様であり，最小支持率が下がるに伴い途中で生成される項木パターンの数が増えるためであると考えられる．

実データ集合 D_{leu} に対して，最小支持率 σ が 1.0 から 0.6 にかけては実行時間にほとんど差がないが，σ が 0.5 以下の時は実行時間が削減できていることが確認できる．これは，従来手法に比べ提案手法では途中で生成される項木パターン数の削減が行われているためだと考えられる．

また実データ集合 D_{leu} に対して，最小支持率 σ が 0.3 のときに実行時間が削減できているが，それ以外ではあまり差がないことが確認できる．これは，従来手法において頻出か否かの判定に用いるマッチングアルゴリズム Matching と OneMatching に比べ，提案手法で頻出か否かの判定に用いるマッチングアルゴリズム xMatching2 が遅いためだと考えられる．つまり，σ がある程度小さくなり途中で生成される項木パターン数の削減量が大きくならないと，マッチングアルゴリズムの実行時間の差によって，総実行時間が速くならないと考えられる．この考察が正しいかどうか確認するために，途中で生成される項木パターン数と，それらが頻出か否かの判定を行う回数を調べる必要があると考えられる．

6. おわりに

本研究では，項木パターン中に複数の変数ラベルが存在するが，ある 1 種類の変数ラベルを持つ変数のみ繰り返し出現することを許す非線形な項木パターンを 1rep項木パターンと定義した．そして，1rep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アルゴリズム xMatching1 と xMatching2 を提案した．さらに，田中ら [10] の提案した頻出 1 変数項木パターンの枚挙アルゴリズムを発展させ，マッチングアルゴリズム xMatching2 を用いることでより高速に枚挙を行う枚挙アルゴリズム Gen-F1TTPX を提案した．提案したマッチングアルゴリズムおよび枚挙アルゴリズムを計算機上に実装し，実データ集合と人工データ集合を用いて評価実験を行った．

項木パターン中に定数 k 個の変数ラベルが存在し，それらが繰り返し出現することを許す非線形な項木パターンを k 変数項木パターンとよぶ．k 変数項木パターンは 1 変数項木パターンと比べ，データ中のいくつかの構造が繰り返し出現することを表現できるため，糖鎖データなどの実データの表現により適していると考えられる．また圧縮木という項木パターンを用いた順序木の可逆圧縮手法にも応用できると考えられる．今後の課題としては，本研究で提案した枚挙アルゴリズムを発展させ，頻出 k 変数項木パターンの枚挙アルゴリズムの開発を行うことが挙げられる．また，頻出 1 変数項木パターンの枚挙アルゴリズムの逐次出力も重要な課題である．

表 1 実データ集合 D_{leu} を入力としたときの実行結果

最小支持率 σ	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3
σ 頻出な 1 変数項木パターンの数	1	1	1	4	10	10	12	35
従来手法 Gen-F1TTP の実行時間 (ms)	87	116	301	11,776	22,153	379,812	523,240	3,043,844
提案手法 Gen-F1TTPX の実行時間 (ms)	26	94	400	15,036	21,683	149,938	194,767	597,817

表 2 人工データ集合 D_{syn} を入力としたときの実行結果

最小支持率 σ	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3
σ 頻出な 1 変数項木パターンの数	1	3	3	3	6	6	9	12
従来手法 Gen-F1TTP の実行時間 (ms)	201	2,591	8,789	23,927	77,853	247,645	806,160	3,477,453
提案手法 Gen-F1TTPX の実行時間 (ms)	219	4,264	16,024	41,615	125,993	329,385	873,499	2,573,719

謝辞

本研究は JSPS 科研費 JP19K12102,JP19K12103, JP21K12021 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] T. Asai, K. Abe, S. Kawasoe, H. Sakamoto, H. Arimura, and S. Arikawa, Efficient substructure discovery from large semi-structured data, IEICE Trans. Inf. Syst., Vol. E87-D(12), 2754–2763, 2004.
[4] KEGG GLYCAN Database, https://www.genome.jp/kegg/glycan/ (accessed 22 February 2022)
[5] 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.
[6] T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, and T. Kuboyama, Enumeration of Maximally Frequent Ordered Tree Patterns with Wildcards for Edge Labels, IPSJ Trans. Math. Model. Appl.(TOM), Vol. 10(2), 59–69, 2017.
[7] 酒井 笑理, 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マッチングアルゴリズムの改良, 第 182 回アルゴリズム研究会, AL182-11, 2021.
[8] 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.
[9] 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.
[10] 田中 知希, 鈴木 祐介, 内田 智之, 宮原 哲浩, 頻出 1 変数項木パターンの枚挙アルゴリズム, 第 120 回人工知能基本問題研究会, SIG-FPAI-120, 24–29, 2022.
[11] M. Zaki, Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications, IEEE Trans. Knowledge and Data Engineering, Vol. 17(8), 1021–1035, 2005.