3F2-02

極大頻出な縮約可能変数つきタグ木パターンを用いた半構造文書からの情報抽出

Information Extraction from Semistructured Documents using Maximally Frequent Tag Tree Patterns with Contractible Variables

宮原 哲浩[*1] Tetsuhiro Miyahara
鈴木 祐介[*2] Yusuke Suzuki
正代 隆義[*2] Takayoshi Shoudai
内田 智之[*1] Tomoyuki Uchida
高橋 健一[*1] Kenichi Takahashi
上田 祐彰[*1] Hiroaki Ueda

[*1]広島市立大学 情報科学部 Faculty of Information Sciences, Hiroshima City University
[*2]九州大学大学院 システム情報科学府・研究院 Department of Informatics, Kyushu University

In order to extract meaningful and hidden knowledge from semistructured documents such as HTML or XML files, methods for discovering frequent patterns or common characteristics in semistructured documents have been more and more important. We propose new methods for discovering maximally frequent tree structured patterns in semistructured documents by using tag tree patterns as hypotheses. A tag tree pattern is an edge labeled tree which has ordered or unordered children and structured variables. An edge label is a tag or a keyword in such documents, and a variable can match an arbitrary subtree, which represents a field of a semistructured document. As a special case, a contractible variable can match an empty subtree, which represents a missing field in a semistructured document. We present algorithms for generating all maximally frequent ordered and unordered tag tree patterns with contractible variables.

1. はじめに

インターネットの発展に伴い，Web 文書も急速に増大している．本研究の目的は，HTML/XML ファイルのような木構造を持つ Web 文書から知識を発見することである．このような Web 文書は，半構造データ (semistructured data) と呼ばれている [1]. 半構造データから，意味がある知識や情報を抽出するためには，それらを特徴づける木構造パターンを発見することが必要である．半構造 Web 文書から特徴的な木構造パターン (縮約可能変数つきの極大頻出なタグ木パターン) をすべて生成するアルゴリズムを提案したので [10], 本稿で報告する．

Object Exchange Model (OEM) [1] に基づき，根つきの木を HTML/XML ファイルのような半構造データの表現として用いる．本稿では，“順序つき” または “順序” とは “子が順序を持つ” ことを意味し，“順序無し” または “無順序” とは “子が順序を持たない” ことを意味する．我々は，そのような木構造データに共通な木構造パターンを表現するため，**タグ木パターン** (tag tree pattern) を提案している [7, 8, 9, 10](2. 節). タグ木パターンは，辺ラベル，順序つきまたは順序無しの子，および構造的変数を持つ，根つきの木構造パターンである．辺ラベルは，タグ, キーワード, または特別な記号 “?”(文字列のワイルドカードを表す) であり，頂点ラベルを持たない．変数は，半構造文書のフィールドを表す任意の部分木にマッチすることができる．特別な場合として，縮約可能変数 (contractible variable) は，半構造文書における欠落フィールドを表す 1 頂点から成る木にマッチすることができる．多くの半構造データは，欠落フィールドなどの不定形性 (irregularity) を持つので，縮約可能変数つきのタグ木パターンは，半構造文書の木構造パターンを表現するのに適している．図 1 の木構造データとタグ木パターンの例において，タグ木パターン t_2 のラベル “x” を持つ変数は， 部分木 g_1 にマッチし， t_2 のラベル “y” を持つ縮約可能変数は，1 頂点から成る木 g_2 にマッチする．

グラフ構造や木構造に基づくデータマイニング，グラフ構造や木構造データからの頻出部分構造の発見が，近年盛んに研究されている [2, 3, 4, 6, 7, 8, 9, 14, 15]. 本研究における発見の目標は，単純な頻出パターンではなく，頂点数のような構文的なサイズに関する極大頻出パターンでもない．提案手法を，均質でない半構造 Web 文書からの情報抽出に応用するため，発見の目標は意味的なものであり，半構造文書に共通する構造的特徴を表現するもので，**極大頻出タグ木パターン** (maximally frequent tag tree pattern)(3. 節) と呼ばれる．“意味的” とは，極大性が木構造パターンの表現能力 (2. 節の言語のこと) で規定されることを示す．

本稿では，次のデータマイニング問題を考える．**MFOTTP** (**MFUTTP**) (3. 節) とは，順序 (無順序) 半構造データの集合から，ユーザが指示する閾値以上の頻度を持つような，極大頻出順序 (無順序) タグ木パターンをすべて生成するという問題である．図 1 の例を考える．半構造データの集合 $\{T_1, T_2, T_3\}$ に対して，t_2 は極大 $\frac{2}{3}$ 頻出な順序タグ木パターンである．実際，t_2 は T_2 と T_3 にマッチするが, T_1 にはマッチしない．タグ木パターン t_1 も T_2 と T_3 にマッチする．しかし，t_1 は，2 個以上の頂点を持つ任意の木にマッチするので，過度に一般化されたものであり，有用ではない．よって，求めるタグ木パターンの意味的な極大性は重要である．

タグ木パターンは，任意の木とマッチする構造的変数を持ち，木構造パターンの部分構造でなく全体構造を表現するという点で，他の木構造パターンの表現 [2, 4, 14] とは異なる．我々は以前の研究として，[5]([12]) の無順序 (順序) 木をすべて生成するアルゴリズムと我々の極大性テスト法を用いて，縮約可能変数を持たない極大頻出な無順序 (順序) タグ木パターンすべてを生成するアルゴリズムを与えた [7]([9]). また，[8] において，極小一般化である縮約可能変数つきの順序タグ木パターンひとつを多項式時間で発見するアルゴリズムを与えた．本稿で提案するアルゴリズムは，タグ木パターンの頻度を計算するために，縮約可能変数つきの順序 (無順序) 項木の多項式時間マッチング判定アルゴリズム [13] を使っている．本稿の結果として，4. 節において，縮約可能変数つきの極大頻出な無順序 (順序) タグ木パターンすべてを生成するアルゴリズム Gen-Mfuttp(Gen-Mfottp) を与える．これは，仮説生成の第 1 段階を改良して，縮約可能変数つきのタグ木パターンを対象とするように仮説の表現能力を豊かにして，[7]([9]) の結果を拡張したものである．

連絡先: 宮原哲浩, 広島市立大学情報科学部知能情報システム工学科, 〒 731-3194 広島市安佐南区大塚東 3-4-1, Email:miyahara@its.hiroshima-cu.ac.jp

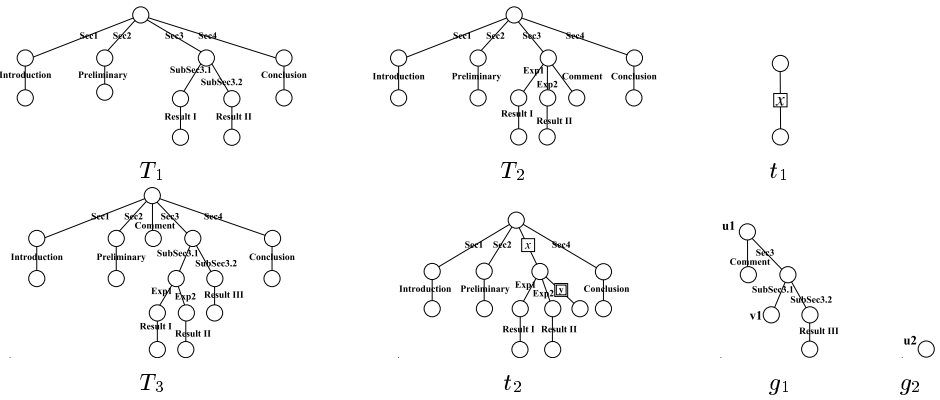


図 1: 順序タグ木パターン t_1, t_2 と順序木 T_1, T_2, T_3, g_1, g_2. 縮約不可 (可能) 変数は,1 本線 (2 本線) で囲まれた箱で表す. 変数の箱とその要素とは線で結ばれている. 箱の中のラベルは, 変数ラベルである.

2. 項木とタグ木パターン

$T = (V_T, E_T)$ を, 頂点集合 V_T, 辺集合 E_T を持つ, 順序つきの子または順序無しの子を持つ, 根つき木とする. 順序つき (順序無し) の子を持つ根つき木を**順序木** (ordered tree)(**無順序木** (unordered tree)) と呼ぶ. E_g と H_g を E_T の分割とする, すなわち, $E_g \cup H_g = E_T$ かつ $E_g \cap H_g = \emptyset$ とする. さらに, $V_g = V_T$ とする. このとき, T が順序木であれば, 3 つ組 $g = (V_g, E_g, H_g)$ を**順序項木** (ordered term tree) と呼び, T が無順序木であれば, 3 つ組 $g = (V_g, E_g, H_g)$ を**無順序項木** (unordered term tree) と呼ぶ. V_g,E_g,H_g の要素をそれぞれ, **頂点**, **辺**, **変数**と呼ぶ.

以下では, "順序つき" と "順序無し" を区別する必要が無いときには, **項木**または**タグ木パターン**と言うことにする. 項木またはタグ木パターンのすべての変数ラベルは異なるものとする. Λ を辺ラベルの集合, X を変数ラベルの集合とし, $\Lambda \cap X = \phi$ であるものとする. $[v, v']$ は, v が v' の親である変数 $\{v, v'\} \in H_g$ を表す. このとき, v を $[v, v']$ の**親ポート** (parent port) といい, v' を $[v, v']$ の**子ポート** (child port) という.

X^c を X の部分集合とする. X^c に属する変数ラベルを**縮約可能変数ラベル**(contractible variable label) と呼ぶ. 縮約可能変数ラベルは, 子ポートが葉である変数にだけ付けることができる. 縮約可能変数ラベルを持つ変数を**縮約可能変数**(contractible variable) と呼ぶ. 縮約可能変数は, 後で述べるように, 1 頂点だけから成る木を代入することも許される. 縮約可能変数でない変数を, **縮約不可変数** (uncontractible variable) と呼ぶ. 変数 $[v, v']$ に対して, 変数の種類に注目しているときは, $[v, v']^c$, $[v, v']^u$ で, それぞれ, 縮約可能変数, 縮約不可変数を表すことにする.

順序項木 g の任意の内部頂点 u のすべての子は全順序を持つ. 順序項木 g の頂点 u の子に対する順序を $<_u^g$ で表す. $f = (V_f, E_f, H_f)$,$g = (V_g, E_g, H_g)$ を順序 (無順序) 項木とする. 次の条件 (1)-(4)((1)-(3)) を満たす,V_f から V_g への全単射 φ が存在するとき, f と g は**同型**であるという. (1) f の根は, φ により g の根に写される. (2) $\{u, v\} \in E_f$ と $\{\varphi(u), \varphi(v)\} \in E_g$ が同値であり, 対応する 2 つの辺が同じ辺ラベルを持つ. (3) $[u, v] \in H_f$ と $[\varphi(u), \varphi(v)] \in H_g$ が同値である. 特に, $[u, v]^c \in H_f$ と $[\varphi(u), \varphi(v)]^c \in H_g$ が同値である. (4) f と g が順序項木である場合, 2 つ以上の子を持つ,f の任意の内部頂点 u と, u の任意の 2 つの子 u',u'' に対して, $u' <_u^f u''$ と $\varphi(u') <_{\varphi(u)}^g \varphi(u'')$ が同値である.

g を項木, x を X の変数ラベルとする. u を g の根, u' を g の葉とし, $\sigma = [u, u']$ をこの 2 頂点のリストとする. このとき, $x := [g, \sigma]$ という形の表現を x に対する**束縛** (binding) と呼ぶ. もし x が X^c に属する縮約可能変数ラベルであれば,g は 1 頂点 u から成る木であってもよく, このときは $\sigma = [u, u]$ となる. この場合が, 束縛に対して,1 頂点から成る木が許される唯一の場合である. f,g を 2 つの順序 (無順序) 項木とする. 新しい順序 (無順序) 項木 $f\{x := [g, \sigma]\}$ は, 束縛 $x := [g, \sigma]$ を f に, 次のように適用して得られる. $e = [v, v']$ を, 変数ラベル x を持つ f 中の変数とする. g' を g のコピーとし, g' の頂点 w, w' は, それぞれ g の頂点 u, u' に対応するとする. 変数 $e = [v, v']$ に対して, e を H_f から削除し, 頂点 v, v' を, それぞれ g' の頂点 w, w' と同一視することにより, g' を f に追加する. もし g が 1 頂点から成る木であれば, つまり $u = u'$ であれば, 束縛を適用した後で v と v' を一致させる. **代入** (substitution) θ とは, 束縛の有限集合 $\{x_1 := [g_1, \sigma_1], \cdots, x_n := [g_n, \sigma_n]\}$ のことである. ここで, x_i は, X の相異なる変数ラベルとする. 代入 θ を項木 f に適用して得られる項木 (**代入例** (instance) という) $f\theta$ とは, f に対して θ 中のすべての束縛 $x_i := [g_i, \sigma_i]$ を同時に適用して得られる項木のことである. f の根を $f\theta$ の根とする. f が順序項木であるとき, $f\theta$ の任意の頂点 v に対して, 新しい全順序 $<_v^{f\theta}$ を自然に定めることができる. 例として, 図 1 の木 g_1, g_2, T_3 と項木 t_2, を考える. $\theta = \{x := [g_1, [u_1, v_1]], y := [g_2, [u_2, u_2]]\}$ を代入とする. θ による t_2 の代入例 $t_2\theta$ は, 木 T_3 に同型である.

Λ_{Tag} と Λ_{KW} を無限または有限個の語から成る言語とし, $\Lambda_{Tag} \cap \Lambda_{KW} = \emptyset$ であるとする. さらに, $\Lambda = \Lambda_{Tag} \cup \Lambda_{KW}$ とする. Λ_{Tag},Λ_{KW} の語をそれぞれ, **タグ** (tag), **キーワード** (keyword) と呼ぶ. **順序 (無順序) タグ木パターン** (ordered (unordered) tag tree pattern) とは, 辺ラベルがタグかキーワードか特別な記号 "?" であるような, 順序 (無順序) 項木のことである. $\Lambda_?$ を Λ の部分集合とする. 記号 "?" は, $\Lambda_?$ の任意の語に対するワイルドカードである. 変数を持たないタグ木パターンを, **基礎タグ木パターン** (ground tag tree pattern) という.

タグ木パターンの辺 $\{v, v'\}$ と, 木の辺 $\{u, u'\}$ について, 次の条件 (1)-(3) を満たすときに, $\{v, v'\}$ が $\{u, u'\}$ と**マッチする** (match) という. (1) $\{v, v'\}$ の辺ラベルがタグであれば, $\{u, u'\}$ の辺ラベルは, 同じタグであるか, または, $\{u, u'\}$ 上

のタグと等しいとみなされる別のタグである．(2) $\{v, v'\}$ の辺ラベルがキーワードであれば，$\{u, u'\}$ の辺ラベルもキーワードであり，$\{v, v'\}$ の辺ラベルが $\{u, u'\}$ の辺ラベルの部分文字列である．(3) $\{v, v'\}$ の辺ラベルが "?" であれば，$\{u, u'\}$ の辺ラベルは $\Lambda_?$ の要素である．基礎順序 (無順序) タグ木パターン $\pi = (V_\pi, E_\pi, \emptyset)$ が，順序 (無順序) 木 $T = (V_T, E_T)$ に**マッチする**とは，次の条件 (1)-(4)((1)-(3)) を満たすような V_π から V_T への全単射 φ が存在するときにいう．(1) π の根は，φ により T の根に写される．(2) $\{v, v'\} \in E_\pi$ と $\{\varphi(v), \varphi(v')\} \in E_T$ が同値である．(3) 任意の $\{v, v'\} \in E_\pi$ について，$\{v, v'\}$ は $\{\varphi(v), \varphi(v')\}$ とマッチする．(4) π と T が順序つきである場合，π の 2 つ以上の子を持つ任意の内部頂点 u と，u の任意の 2 つの子 u', u'' に対して，$u' <_u^\pi u''$ と $\varphi(u') <_{\varphi(u)}^T \varphi(u'')$ が同値である．タグ木パターン π が木 T に**マッチする**とは，ある代入 θ があって，$\pi\theta$ が基礎タグ木パターンであり，$\pi\theta$ が T とマッチするときにいう．

$\mathcal{OTT}_\Lambda$ ($\mathcal{UT}_\Lambda$) は，Λ の辺ラベルを持つ順序 (無順序) 木の全体集合を表すとする．$\mathcal{OTTP}_\Lambda^c$ ($\mathcal{UTTP}_\Lambda^c$) は，縮約可能変数，縮約不可変数，および Λ のタグとキーワードを持つ順序 (無順序) タグ木パターンの全体集合を表すとする．タグ木パターン $\pi \in \mathcal{OTTP}_\Lambda^c$ ($\mathcal{UTTP}_\Lambda^c$) に対して，**言語** $L_\Lambda(\pi)$ は，$L_\Lambda(\pi) = \{$ 木 $T \in \mathcal{OT}_\Lambda(\mathcal{UT}_\Lambda) \mid \pi$が T とマッチする $\}$ と定義され，π の表現能力を表すものである．

3. データマイニング問題

データマイニング設定:　順序 (無順序) 半構造データの集合 $\mathcal{D} = \{T_1, T_2, \ldots, T_m\}$ とは，順序 (無順序) 木の集合である．$\Lambda_\mathcal{D}$ を $\mathcal{D}$ の木のすべての辺ラベルの集合とする．順序 (無順序) タグ木パターン π の $\mathcal{D}$ に関する**マッチ数**とは，π とマッチするような順序 (無順序) 木 $T_i \in \mathcal{D}$ $(1 \leq i \leq m)$ の個数であり，$match_\mathcal{D}(\pi)$ と表される．$\mathcal{D}$ に関する π の**頻度**は，$supp_\mathcal{D}(\pi) = match_\mathcal{D}(\pi)/m$ と定義される．σ を，$0 < \sigma \leq 1$ である実数とする．タグ木パターン π は，$supp_\mathcal{D}(\pi) \geq \sigma$ であるとき，$\mathcal{D}$ に関して σ **頻出** (σ-frequent) であると言う．Π は，$\mathcal{OTTP}_\Lambda^c$ または $\mathcal{UTTP}_\Lambda^c$ を表すとし，$\Lambda' \subseteq \Lambda_{Tag} \cup \Lambda_{KW} \cup \{?\}$ とする．$\Pi(\Lambda')$ は，そのすべての辺ラベルが Λ' に属するようなタグ木パターン $\pi \in \Pi$ の全体集合を表す．Tag を Λ_{Tag} の有限部分集合とし，KW を Λ_{KW} の有限部分集合とする．順序 (無順序) タグ木パターン $\pi \in \mathcal{OTTP}_\Lambda^c(Tag \cup KW \cup \{?\})$ ($\mathcal{UTTP}_\Lambda^c(Tag \cup KW \cup \{?\})$) は，次の条件 (1) および (2) が満たされるとき，$\mathcal{D}$ に関して**極大** σ **頻出** (maximally σ-frequent) であると言う．(1) π は $\mathcal{D}$ に関して σ 頻出である．(2) 任意のタグ木パターン π' in $\mathcal{OTTP}_\Lambda^c(Tag \cup KW \cup \{?\})$ ($\mathcal{UTTP}_\Lambda^c(Tag \cup KW \cup \{?\})$) に対して，もし $L_\Lambda(\pi') \subsetneq L_\Lambda(\pi)$ であれば，π' は $\mathcal{D}$ に関して σ 頻出ではない．

極大頻出な順序タグ木パターンをすべて生成する問題 (All Maximally Frequent Ordered Tag Tree Patterns, **MFOTTP**)
入力: 順序半構造データの集合 $\mathcal{D}$, 閾値 σ $(0 < \sigma \leq 1)$, タグの有限集合 Tag, キーワードの有限集合 KW.
仮定: $\Lambda_\mathcal{D} \subseteq \Lambda_? \subsetneq \Lambda$.
問題: $\mathcal{D}$ に関して極大 σ 頻出である，$\mathcal{OTTP}_\Lambda^c(Tag \cup KW \cup \{?\})$ の順序タグ木パターンをすべて生成する．

極大頻出な無順序タグ木パターンをすべて生成する問題 (All Maximally Frequent Unordered Tag Tree Patterns, **MFUTTP**)
入力: 無順序半構造データの集合 $\mathcal{D}$, 閾値 σ $(0 < \sigma \leq 1)$, タグの有限集合 Tag, キーワードの有限集合 KW.
仮定: $\Lambda_\mathcal{D} \subsetneq \Lambda_? \subsetneq \Lambda$, かつ，$\Lambda - \Lambda_?$ および　$\Lambda_? - \Lambda_\mathcal{D}$ は無限集合.
問題: $\mathcal{D}$ に関して極大 σ 頻出である，$\mathcal{UTTP}_\Lambda^c(Tag \cup KW \cup \{?\})$ の無順序タグ木パターンをすべて生成する．

4. すべての極大頻出タグ木パターンの生成

本節では，問題 MFOTTP を解くアルゴリズム，つまり，すべての極大 σ 頻出な順序タグ木パターンを生成するアルゴリズム Gen-Mfottp の概要を示す．$\mathcal{D}$ を順序半構造データの入力集合とする．手続き Sub-Mfottp において，[2] のすべての順序木を生成するアルゴリズムを使っている．縮約不可変数だけから成るタグ木パターンとは，頂点と縮約不可変数だけから成るタグ木パターンのことを言う．縮約不可変数だけから成る順序タグ木パターンを，同じ木構造を持つ順序木とみなすことができる．[2] のと同じ親子関係を用いることにより，一般的なパターンから特殊なパターンへと進む深さ優先探索とバックトラックをする方法で，縮約不可変数だけから成る順序タグ木パターンすべてを，重複無く枚挙することができる．木構造パターンと木構造データのマッチの意味は [2] と異なるが，縮約不可変数だけから成る順序タグ木パターンを生成する過程において，親のパターンは子のパターンよりも一般的である関係は同様に成立する．Gen-Mfottp と同様にして，問題 MFUTTP を解くアルゴリズム，つまり，すべての極大 σ 頻出な無順序タグ木パターンを生成するアルゴリズム Gen-Mfuttp を与えることができる．Sub-Mfottp に対応する手続 Sub-Mfuttp において，[4, 11] のすべての無順序木を生成するアルゴリズムと親子関係を使用して，縮約不可変数だけから成る無順序タグ木パターンを生成する過程において，仮説空間の深さ優先探索とバックトラックを実現している．アルゴリズム Gen-Mfottp，Gen-Mfuttp の詳細および正当性については，[10] を参照のこと．

Algorithm Gen-Mfottp;
begin
  $\Pi(\sigma) := \emptyset$; $\pi := (\{v, v'\}, \emptyset, [v, v']^u)$;
  Sub-Mfottp(π); **return** $\Pi(\sigma)$
end.

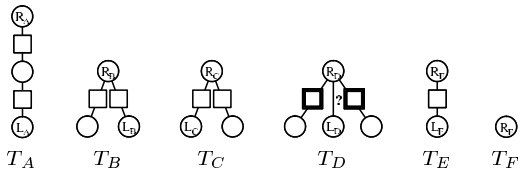
Procedure Sub-Mfottp(π);
begin
  if π is not σ-frequent w.r.t. $\mathcal{D}$ **then return**
  else Basic-Mfottp(π);
  foreach child tag tree pattern π' of π **do** Sub-Mfottp(π')
end;

Procedure Basic-Mfottp(π);
begin

Step 1. **Generate σ-frequent tag tree patterns**:
Let $H_\pi = \{h_1, \ldots, h_k\}$ be the variable set of π. We perform procedure Substitution-Ot(π, h_1, k).

Step 2. **Eliminate redundancy**:
For each $\pi \in \Pi(\sigma)$, if there exists a pair of contractible variables $[u, v]^c$ and $[u, v']^c$ such that v' is the immediately right sibling of v, then we remove π from $\Pi(\sigma)$.

Step 3. **Maximality test1**:
$\theta_X(x) = \{x := [T_X, [R_X, L_X]]\}$
$X \in \{A, B, C, D, E, F\}$



For each $\pi \in \Pi(\sigma)$, if there exists an uncontractible (resp. contractible) variable x in π such that $\pi\theta_X(x)$ is σ-frequent w.r.t. $\mathcal{D}$ for any $X \in \{A, B, C, D\}$ (resp. $X \in \{E, F\}$), then π is not maximally σ-frequent w.r.t. $\mathcal{D}$, and we remove π from $\Pi(\sigma)$.

Step 4. **Maximality test2**:

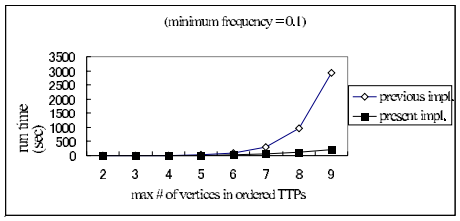
If there exists an edge with "? " in π such that a tag tree pattern obtained from π by replacing the edge with an edge which has a label in $Tag \cup KW$ is σ-frequent w.r.t $\mathcal{D}$, then π is not maximally σ-frequent w.r.t. $\mathcal{D}$, and we remove π from $\Pi(\sigma)$.

end;

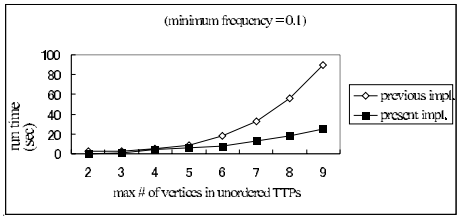
Procedure SUBSTITUTION-OT(π, h_i, k);
begin
if $i = k + 1$ **then begin** $\Pi(\sigma) := \Pi(\sigma) \cup \{\pi\}$; **return end**;
If the child port of h_i is not a leaf
then SUBSTITUTION-OT(π, h_{i+1}, k);
VARIABLE-REPLACING-OT(π, h_i, k);
return
end;

5. 実現と実験結果

アルゴリズム GEN-MFOTTP(GEN-MFUTTP) における，縮約不可変数だけから成る σ 頻出な順序 (無順序) タグ木パターンを探索する過程の性能を評価するため，以前の実現 ("previous impl.") と今回の実現 ("present impl.") による，そのようなすべてのパターンを生成する 2 種類の実験を行った．以前の実現 [9] ([7]) は，この過程における仮説空間探索の枝刈りをすることができない．今回の実現 GEN-MFOTTP (GEN-MFUTTP) は，仮説空間探索の枝刈りをすることができる．実現は，Sun workstation Ultra-10 (clock 333MHz) 上で，GCL2.2 により行った．サンプルファイルは，衣服の販売データに関する XML ファイルから変換されたものである．このサンプルファイルは,172 個の木構造データから成る．このファイルの木の頂点数の最大値は 11 である．仮説空間における順序 (無順序) タグ木パターンの頂点数の最大値 ("max # of vertices in ordered (unordered) TTPs") を設定できるようにしている．Exp.1 (Exp.2) は，最小支持率 (minimum frequency) を 0.1 として，仮説空間の順序 (無順序) タグ木パターンの最大頂点数を変化させたときの 2 つの実現の実行時間 (run time, 秒) を示している．これらの実験により，上記過程における仮説空間探索の枝刈りが有効であることがわかる．



Exp. 1



Exp. 2

図 2: 実験結果

6. おわりに

HTML/XML ファイルのような半構造 Web 文書からの知識発見について研究し, 縮約可能変数つきの極大頻出な順序 (無順序) タグ木パターンすべてを生成するアルゴリズムを与えた．本研究の一部は，科学研究費基盤研究 (C)(13680459), および広島市立大学特定研究費 (2101) の助成による．

参考文献

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 2000.

[2] T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto, and S. Arikawa. Efficient substructure discovery from large semi-structured data. *Proc. 2nd SIAM Int. Conf. Data Mining (SDM-2002)*, pages 158–174, 2002.

[3] T. Asai and H. Arimura. Algorithms for mining semistructured data (in Japanese). *IEICE Trans. Inf. Syst.*, J87-D-I(2):79–96, 2004.

[4] T. Asai, H. Arimura, T. Uno, and S. Nakano. Discovery of frequent substructures in large unordered trees. *Proc. DS-2003, Springer-Verlag, LNAI 2843*, pages 47–61, 2003.

[5] T. Beyer and S. Hedetniemi. Constant time generation of rooted trees. *SIAM J. Comput.*, 9:706–712, 1980.

[6] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *Proc. PKDD-2000, Springer-Verlag, LNAI 1910*, pages 13–23, 2000.

[7] T. Miyahara, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda. Discovery of frequent tree structured patterns in semistructured web documents. *Proc. PAKDD-2001, Springer-Verlag, LNAI 2035*, pages 47–52, 2001.

[8] T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, S. Hirokawa, K. Takahashi, and H. Ueda. Extraction of tag tree patterns with contractible variables from irregular semistructured data. *Proc. PAKDD-2003, Springer-Verlag, LNAI 2637*, pages 430–436, 2003.

[9] T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda. Discovery of frequent tag tree patterns in semistructured web documents. *Proc. PAKDD-2002, Springer-Verlag, LNAI 2336*, pages 341–355, 2002.

[10] T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda. Discovery of maximally frequent tag tree patterns with contractible variables from semistructured documents. *Proc. PAKDD-2004, Springer-Verlag, LNAI (to appear)*, 2004.

[11] S. Nakano and T. Uno. Efficient generation of rooted trees. *NII Technical Report, NII-2003-005E, National Institute of Infomatics, Japan*, 2003.

[12] W. Skarbek. Generating ordered trees. *Theoretical Computer Science*, 57:153–159, 1988.

[13] Y. Suzuki, T. Shoudai, S. Matsumoto, T. Uchida, and T. Miyahara. Efficient learning of ordered and unordered tree patterns with contractible variables. *Proc. ALT-2003, Springer-Verlag, LNAI 2842*, pages 114–128, 2003.

[14] K. Wang and H. Liu. Discovering structural association of semistructured data. *IEEE Trans. Knowledge and Data Engineering*, 12:353–371, 2000.

[15] T. Washio and H. Motoda. State of the art of graph-based data mining. *SIGKDD Explorations*, 5:59–68, 2003.