2F2-02

縮約可能変数つきタグ木パターンを用いた半構造データからの情報抽出

Information Extraction from Semistructured Data using 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

Information Extraction from semistructured data becomes more and more important. In order to extract meaningful or interesting contents from semistructured data, we need to extract common structured patterns from semistructured data. A tag tree pattern is a rooted tree pattern which has edge labels, ordered children, tree structures of tags and structured variables. An edge label is a tag, a keyword or a wildcard, and a variable can be substituted by an arbitrary tree. In particular, a contractible variable matches arbitrary subtrees, including a singleton vertex. A tag tree pattern is hence suited for representing common tree structured patterns in irregular semistructured data. We present a new method for extracting characteristic tag tree patterns from irregular semistructured data by using an algorithm for finding a minimally generalized tag tree pattern explaining given data. We report some experiments of applying this method to extracting characteristic tag tree patterns from HTML/XML files.

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．正データである不定形な半構造データから特徴的な木構造パターンを抽出するアルゴリズム [17] を利用して，木構造パターンを抽出する実験を行ったので [12]，本稿で報告する．

Object Exchange Model (OEM) [1] に基づき，半構造データを木構造データして取り扱う．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 [13]. これは，根付きの木構造パターンであり，順序つきの子と構造的な変数を持つ．その辺ラベルは，タグかキーワードかワイルドカードであり，頂点ラベルを持たない．木とは，根付きの木であり順序つきの子を持つが，変数を持たないものである．変数には，任意の木を代入できる．

多くの半構造データは，欠損や誤りなどの不定形性 (irregularity) を持つ．Object Exchange Model においては，重要なデータは，葉または部分木として表現されることが多い．よって，本研究では，タグ木パターンに，**縮約可能変数** (contractible variable) と**縮約不可変数** (uncontractible variable) の 2 種類の変数を導入する．縮約可能変数は，頂点数 1 以上の任意の部分木を表す．縮約不可変数は，頂点数 2 以上の任意の部分木を表す．

変数には任意の木を代入できるので，与えられた正データとしての木を説明できる過度に一般化されたパターンを求めることには，意味がない．半構造 Web 文書のような，不定形な，または不完全な木構造データから意味がある情報を抽出するためには，極小に一般化されたタグ木パターン，すなわち**極小一般化タグ木パターン** (minimally generalized tag tree pattern) をみつける必要がある．図 1 の例を考える．T_i から，“Sec1” や “SubSec3.1” のような辺ラベルを残して，それ以外の辺ラベルを無視することにより得られる木を，T_i' とする．タグ木パターン t_1 は，木 T_1', T_2', T_3' を説明している．つまり，T_1', T_2',T_3' は，変数に木を代入することにより，t_1 から得ることができる．t_1 は，極小一般化タグ木パターンである．タグ木パターン t_2 も，これらの 3 つの木を説明する．しかし，t_2 は，2 個以上の頂点を持つ任意の木を説明する．よって，t_2 は過度に一般化されたものであり，意味がない．

木構造データ，半構造データのパターンための知識表現として，regular path expression [6], tree-expression pattern [21], ordered gapped tree pattern [2], tree-association pattern [8] などが提案されている．半構造データからの構造的特徴を発見する方法が他にも提案されている [21, 3, 6]. 我々は，OEM データの意味的側面を重視し，順序のない子を持つような，項木，タグ木パターンを木構造パターンの表現とするデータマイニング，および，そのための計算学習理論の研究を展開している [10, 11, 14, 20]. タグ木パターンは，項木の特別なものとみなせる．タグ木パターンは，任意の木を代入できる構造的な変数を持ち，パターン全体のつながりを表現できる点で，木構造パターンの他の表現形式とは異なる．本研究では，情報抽出へ応用するため，OEM データの構文的側面を重視して，木構造パターンの表現として順序つきの子を持つタグ木パターンを用いる．

Web 文書からの情報抽出が盛んに研究されている [4, 9]. しかし，多くの研究は，テキスト文書に対するものである．半構造データや表のような構造的なデータからの情報抽出，ラッパー抽出は，Web データからの学習，Web マイニングにおいて，関心の高いテーマである [18, 19, 5, 7]. 我々の抽出法は，木構造データから特徴的パターンをみつける手法の応用であり，これは近年盛んに研究されている [3, 11, 13, 21].

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

2. タグ木パターンと情報抽出法

2.1 木構造パターンとしての項木

木構造パターンを表現する項木について説明する．$T = (V_T, E_T)$ を頂点集合 V_T，辺集合 E_T を持つ，子に順序があり，辺

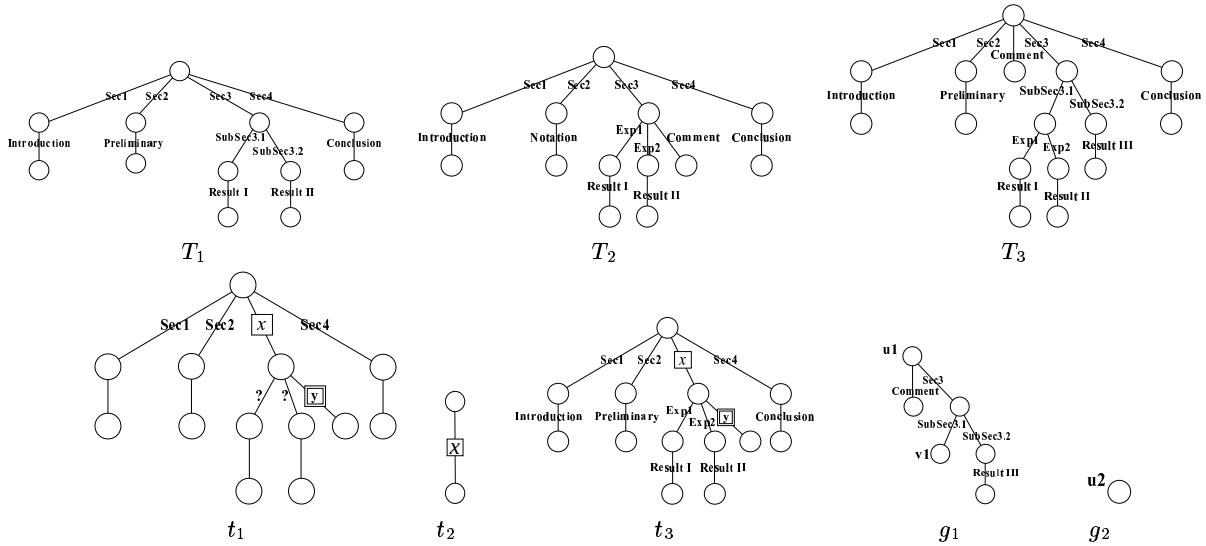


図 1: タグ木パターン t_1,t_2,t_3 と木 T_1,T_2,T_3. 変数は，その要素へ線を持つ箱で表される．縮約不可 (縮約可能) 変数は，1 本線 (2 本線) の箱で表される．箱の中のラベルは，変数ラベルである．記号 "?" は，辺ラベルのワイルドカードである．

ラベルを持つ根付き木 (以後, 木) とする．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，3 つ組 $g = (V_g, E_g, H_g)$ を**項木** (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．項木の辺と変数は，ある言語の語でラベル付けされているとする．変数のラベルを，**変数ラベル**と呼ぶ．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．項木 g の相異なる頂点の列 $v_1, v_2, \ldots, v_i$ が, v_1 から v_i への**道**であるとは，任意の $j(1 \leq j < i)$ に対して,v_j と v_{j+1} から成る辺または変数が存在するときにいう．頂点 v,v' から成る辺または変数が存在し，根から v' への道上に v が存在するとき，v は v' の**親**であるといい，v' は v の**子**であるという．$[v, v']$ 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．

項木 g の任意の内部頂点 u が，u のすべての子に対する全順序を持つとき，g は**順序項木** (ordered term tree) であるという．u の子に対する順序を $<_u^g$ で表す．順序項木 g が**正則** (regular) であるとは，H_g のすべての変数が X の相異なる変数ラベルを持つときにいう．

変数を持たない順序項木を**基礎項木** (ground term tree) と呼び，順序つきの子を持つ木とみなす．$\mathcal{OT}_\Lambda$ は，Λ の辺ラベルを持つ基礎項木全体の集合を表す．辺ラベルの集合 Λ に対して．$\mathcal{OTT}_\Lambda^c$ は，縮約可能変数，または縮約不可変数を持ち，Λ の辺ラベルを持つ順序項木全体の集合を表す．本稿では，縮約可能変数，または縮約不可変数を持つ正則順序項木のみを取り扱うので，これらを単に**項木**と呼ぶ．

$f = (V_f, E_f, H_f)$,$g = (V_g, E_g, H_g)$ を項木とする．次の条件 (1)-(4) 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**であるといい，$f \equiv g$ と表す．(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) 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．

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．新しい項木 $f\{x := [g, \sigma]\}$ 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．$f\theta$ 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，t_3 を図 1 の項木とし，$\theta = \{x := [g_1, [u_1, v_1]], y := [g_2, [u_2, u_2]]\}$ を代入とする．ここで，g_1,g_2 は図 1 の木である．θ による t_3 の代入例 $t_3\theta$ は，図 1 の木 T_3 に同型である．

2.2 タグ木パターン

タグ木パターンについて説明する．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．Λ_{Tag},Λ_{KW} の語をそれぞれ，**タグ** (tag), **キーワード** (keyword) と呼ぶ．**タグ木パターン** (tag tree pattern) とは，辺ラベルがタグかキーワードか特別な記号 "?" (ワイルドカード) であるような，項木のことである．つまり，タグ木パターンとは，辺ラベルの集合 Λ を $\Lambda_{Tag} \cup \Lambda_{KW} \cup \{?\}$ とする項木のことである．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．

本稿では，木構造データとは，辺ラベルがタグかキーワードである基礎タグ木パターンのことであり，単に木と呼ぶ．木の辺ラベルのキーワードは，テキストデータに相当する．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，タグ間の適当な同等関係の下で等しいとみなされるタグである．(2) $\{v, v'\}$ の辺ラベルがキーワードであれば，$\{u, u'\}$ の辺ラベルも同じキーワードである．(3) $\{v, v'\}$ の辺ラベルが "?" であれば，$\{u, u'\}$ の辺ラベルは任意でよい．

基礎タグ木パターン $\pi = (V_\pi, E_\pi, \emptyset)$ が，木 $T = (V_T, E_T)$ にマッチするとは，次の条件 (1)-(4) 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 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．タグ木パターン π の言語 $L_\Lambda(\pi)$ は，$L_\Lambda(\pi) = \{$ 木 $T \in \mathcal{OT}_\Lambda \mid \pi$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．ここで，$\Lambda = \Lambda_{Tag} \cup \Lambda_{KW}$ とする．

2.3 極小一般化タグ木パターンの抽出

本節では，正データである木構造データとみなされる，不定形な半構造データから，特徴的なタグ木パターンを抽出する方法について説明する．タグ木パターン π が，与えられた正データである木の集合 S を説明する，**極小一般化タグ木パターン** (minimally generalized tag tree pattern) であるとは，次の条件 (1) および (2) を満たすときにいう．(1) $S \subseteq L_\Lambda(\pi)$ (π が S を説明する)．(2) $S \subseteq L_\Lambda(\pi') \subsetneq L_\Lambda(\pi)$ を満たすようなタグ木パターン π' は，存在しない．与えられた木の集合に対する，極小一般化タグ木パターンをみつける問題は，計算学習理論の分野においては，極小言語問題 (minimal language problem, MINL) として議論されている [15, 17]．極小言語問題を解く多項式時間アルゴリズム [17] を使って，我々の抽出法は，与えられた木の集合 S を説明する極小一般化タグ木パターンをみつける．このアルゴリズムは，まず，S を説明するような，縮約不可変数だけから成る極小一般化タグ木パターン t をみつける．次に，t の変数を辺ラベルかワイルドカードを持つ辺，または縮約可能変数で置き換えてみて，得られるタグ木パターン t' が S を説明するのならば，t から t' を得る．この置き換えが適用できなくなるまで，この置き換えを繰り返して適用する．最後に，アルゴリズムは，この結果得られたタグ木パターンを出力する．この抽出法は，仮説チェックのために，タグ木パターンが木にマッチするかどうかを判定する多項式時間マッチングアルゴリズム [17] を使っている．このマッチングアルゴリズムは，縮約可能変数を含まない項木に対する多項式時間マッチングアルゴリズム [15, 16] の拡張である．

3. 実現と実験結果

我々は，与えられた半構造データを説明する極小一般化タグ木パターンをみつける，2.3 節の抽出法を実現した．実現は，GCL2.2 で行い，Sun workstation Ultra-10 clock 333MHz 上で実験した．サンプルの半構造データに対する実験結果を，図 2 に示す．これらの実験では，入力の木は，HTML/XML ファイルの構文解析木の部分木を表す．入力の木において，構文解析木の HTML/XML タグから成る木構造は，保持されている．属性と属性値は無視する．タグに対する同値関係は設定していない．タグの作る木構造に注目するため，構文解析木のタグでないテキストデータは同一のダミーのキーワードに変換する．

Exp. 1 から 3 では，提案した抽出法を評価するために，人工的な HTML ファイルからサンプルを作成した．Exp. 1 の入力ファイルは，すべて，約 40 個の頂点を持つ木から構成されている．Exp. 2 の入力ファイルは，90%が約 40 個の頂点を持つ木であり，10%が約 20 個の頂点を持つ木である．Exp. 3 の入力ファイルは，90%が約 40 個の頂点を持つ木であり，10%が約 70 個の頂点を持つ木である．Exp. 1 から 3 に対するグラフは，これらの 3 実験に対するデータ数を変化させたときの，本手法の実行時間 (run time, 秒) を示している．これらの 3 実験に対して，得られたタグ木パターンの頂点数は，ほとんど同じであった．

Exp. 4 においては，サンプルの HTML ファイルは，ローカルな検索機能付きの Web サイトの検索エンジンの検索結果である (http://www.ael.org)．サンプルファイルは，約 18 個の頂点を持つ木 10 個から成る．このサンプルファイルの木は，文献データのレコードである．図 2 (Exp. 4) のタグ木パターンは，このサンプルファイルを説明する極小一般化タグ木パターンである．辺ラベルの無い辺は，ダミーのキーワードを持つ辺を表している．この極小一般化タグ木パターンは，このような木構造データに対するラッパーとみなすことができる．

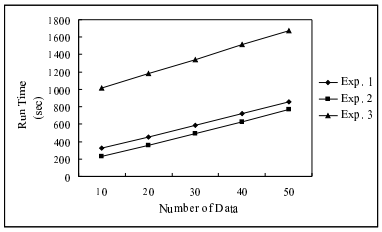
Exp. 5 においては，サンプルの XML ファイルは，DBLP 文献データベース (http://dblp.uni-trier.de/xml/dblp.xml) から作成した．このサンプルファイルは，約 18 個の頂点を持つ木 50 個から成る．Exp. 5(a),5(b) のグラフは，データ数を変化させたときの，得られたタグ木パターンの頂点数と本手法の実行時間 (run time, 秒) を示している．42 個以上のデータに対する，得られたタグ木パターンは同一であった．これにより，本手法を用いて，このような文献データから特徴的なタグ木パターンを抽出するには，少ないデータで十分であることがわかる．

4. おわりに

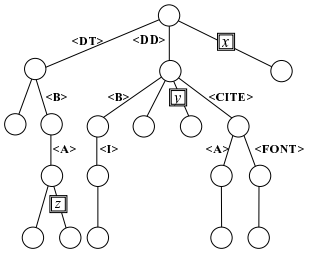
本稿では，半構造データからの情報抽出について研究した．不定形な半構造データから特徴的なタグ木パターンを抽出する方法を提案した．さらに，HTML/XML ファイルから極小一般化タグ木パターンを抽出する実験について述べた．本研究の一部は，科学研究費基盤研究 (C)(13680459), および広島市立大学特定研究費 (2101) の助成による．

参考文献

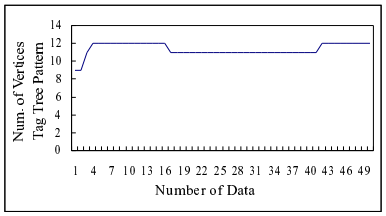
[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan



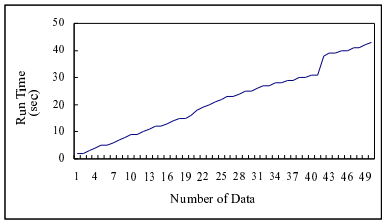
Exp. 1-3



Exp. 4



Exp. 5(a)



Exp. 5(b)

図 2: 半構造データから極小一般化タグ木パターンを抽出する実験結果.

Kaufmann, 2000.

[2] H. Arimura, H. Sakamoto, and S. Arikawa. Efficient learning of semi-structured data from queries. *Proc. ALT-2001, Springer-Verlag, LNAI 2225*, pages 315–331, 2001.

[3] 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.

[4] C.-H. Chang, S.-C. Lui, and Y.-C. Wu. Applying pattern mining to web information extraction. *Proc. PAKDD-2001, Springer-Verlag, LNAI 2035*, pages 4–15, 2001.

[5] W.W. Cohen, H. Mathew, and S.J. Lee. A flexible learning system for wrapping tables and lists in HTML documents. *Proc. WWW 2002*, pages 1–21, 2002.

[6] M. Fernandez and D. Suciu. Optimizing regular path expressions using graph schemas. *Proc. Int. Conf. on Data Engineering (ICDE-98)*, pages 14–23, 1998.

[7] K. Fukuda, A. Ishino, M. Takeda, and F. Matsuo. A proposal of information extraction based on maximal common hedge (in Japanese). *IPSJ SIGNotes Fundamental Infology*, No.066:151–158, 2002.

[8] K. Furukawa, T. Uchida, K. Yamada, T. Miyahara, T. Shoudai, and Y. Nakamura. Extracting characteristic structures among words in semistructured documents. *Proc. PAKDD-2002, Springer-Verlag, LNAI 2336*, pages 356–367, 2002.

[9] N. Kushmerick. Wrapper induction: efficiency and expressiveness. *Artificial Intelligence*, 118:15–68, 2000.

[10] T. Miyahara, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda. Polynomial time matching algorithms for tree-like structured patterns in knowledge discovery. *Proc. PAKDD-2000, Springer-Verlag, LNAI 1805*, pages 5–16, 2000.

[11] 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.

[12] 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.

[13] 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.

[14] T. Shoudai, T. Uchida, and T. Miyahara. Polynomial time algorithms for finding unordered tree patterns with internal variables. *Proc. FCT-2001, Springer-Verlag, LNCS 2138*, pages 335–346, 2001.

[15] Y. Suzuki, R. Akanuma, T. Shoudai, T. Miyahara, and T. Uchida. Polynomial time inductive inference of ordered tree patterns with internal structured variables from positive data. *Proc. COLT-2002, Springer-Verlag, LNAI 2375*, pages 169–184, 2002.

[16] Y. Suzuki, K. Inomae, T. Shoudai, T. Miyahara, and T. Uchida. A polynomial time matching algorithm of structured ordered tree patterns for data mining from semistructured data. *Proc. ILP-2002, Springer-Verlag, LNAI 2583*, pages 270–284, 2003.

[17] Y. Suzuki, T. Shoudai, T. Miyahara, T. Uchida, and S. Hirokawa. Polynomial time inductive inference of ordered term trees with contractible variables from positive data. *Proc. LA Winter Symposium, Kyoto, Japan*, pages 13–1 – 13–11, 2003.

[18] T. Taguchi, K. Koga, and S. Hirokawa. Integration of search sites of the World Wide Web. *Proc. of CUM, Vol.2*, pages 25–32, 2000.

[19] K. Taniguchi, H. Sakamoto, H. Arimura, S. Shimozono, and S. Arikawa. Mining semi-structured data by path expressions. *Proc. DS-2001, Springer-Verlag, LNAI 2226*, pages 378–388, 2001.

[20] T. Uchida, T. Miyahara, and T. Shoudai. Discovery of tree structured patterns from semistructured text documents. *Proc. PAKDD 2002 Workshop on Text Mining*, pages 5–14, 2002.

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