科学研究費助成事業　　研究成果報告書

平成 ２６ 年　 ６ 月　 ４ 日現在

機関番号：１２１０２
研究種目：基盤研究(C)
研究期間：2011～2013
課題番号：２３５００１１０

研究課題名（和文）正規木文法のための差分抽出アルゴリズムの開発

研究課題名（英文）A Difference Extraction Algorithm for Regular Tree Grammar

研究代表者
鈴木　伸崇（SUZUKI, Nobutaka）

筑波大学・図書館情報メディア系・准教授

研究者番号：６０３０５７７９

交付決定額（研究期間全体）：（直接経費）　　2,700,000円、（間接経費）　　810,000円

研究成果の概要（和文）：XMLのスキーマ定義は，現実の利用状況の変化に応じて更新される．このような場合，データの妥当性の維持等の理由から，スキーマの更新内容を適切に把握することが必要である．そこで本研究では，XMLのスキーマ言語としてよく用いられる正規木文法を対象に，スキーマの差分抽出アルゴリズムの開発・評価を行った．更に，その応用可能性を確認するため，XPath式をスキーマ更新に応じて修正するアルゴリズムの開発を行った．

研究成果の概要（英文）：Schemas of XML documents are continuously updated according to changes in real world. In such a case, we have to precisely know how a schema is updated to keep the validity of the XML documents. Thus, in this study we consider constructing a difference extraction algorithm for regular tree grammar. Moreover, to verify the availability of the algorithm, we construct an algorithm for transforming XPath Expressions according to schema evolution.

研究分野：情報科学

科研費の分科・細目：情報学・メディア情報学・データベース

キーワード：構造化文書　XML

１．研究開始当初の背景

スキーマとそれに関して妥当な XML データを蓄積・管理する場合，時間の経過と共に格納すべきデータの構造や種類は変化し，それに応じてスキーマ定義も更新されることが多い．このような場合，スキーマ更新履歴の管理，スキーマの更新に応じた XML データの修正等が必要となるため，スキーマの更新内容を適切に把握することが重要である．特に，管理者が複数で更新内容の共有が必要な場合や，スキーマが複雑で更新内容が多岐にわたる場合等は，スキーマの更新内容を把握することがより重要となる．このためには更新前後のスキーマ間で差分抽出を行う必要があるが，これを適切に行える手法はこれまでほとんど提案されていない．

これまで，文字列間の差分(編集操作列)抽出に関しては基本的なアルゴリズムが確立しており，順序木や XML データ間の差分抽出に関しても，順序木間の木編集操作列を求めるアルゴリズムがいくつか提案されている．しかし，これら既存のアルゴリズムは木文法の意味を解することができないため，正規木文法の適切な差分抽出を行うことは困難である．また，Leonardi らは DTD の差分抽出を行うアルゴリズムが提案している．しかし，これはヒューリスティックに基づく手法であり，最適解(コスト最小の編集操作列)が得られるとは限らない．また，正規木文法等，表現力が DTD より真に高いスキーマ言語も存在する．そのため，そのようなスキーマ言語に対してこのアルゴリズムを適用することは困難である．

２．研究の目的

本研究は，スキーマ定義言語としては最も表現力の高い正規木文法を対象とし，正規木文法のための差分抽出アルゴリズムの開発を行う．なお，正規木文法は RLEAX NG の形式モデルであり，その表現力は局所木文法(DTD に相当)や単一型木文法(W3C XML Schema に相当)よりも真に高いことが分かっている．したがって，本アルゴリズムは XML の主要なスキーマ言語に適用可能である．

文字列間や順序木間の差分抽出は多項式時間可解であるが，正規木文法の差分抽出は文法の表す意味も絡むより複雑な問題である．このことから，正規木文法の差分抽出問題は計算困難であると予想される．したがって，そのままでは効率の良いアルゴリズムを開発することは困難である．そこで本研究では，以下の 3 点を主な研究目的とする．

(1) 正規木文法の差分抽出が効率よく行えるための十分条件を求める
(2) その十分条件の下で，正規木文法の差分抽出を行う効率の良いアルゴリズムを構成する
(3) 以上で開発したアルゴリズムに関する評価実験を行い，有効性を検証する

３．研究の方法

正規木文法は，終端記号の集合，非終端記号の集合，開始記号，および生成規則の集合から構成される．例えば，正規木文法 $G = (N,T,S,P)$ を考える．ここで，

$N = \{R, T, M, E, Pcdata\}$
$T = \{\text{staffs, staff, name, email, pcdata}\}$
$S = \{R\}$
$P = \{R \rightarrow \text{staffs}(T^*), T \rightarrow \text{staff}(M\ E),$
$M \rightarrow \text{name}(Pcdata), E \rightarrow \text{email}(Pcdata),$
$Pcdata \rightarrow \text{pcdata}(\varepsilon)\}$

N が非終端記号の集合で，個々の非終端記号が要素の型を表す．T が終端記号の集合で，個々の終端記号が要素を表す．また，S が開始記号，P が生成規則の集合である．正規木文法の差分抽出は，2 つの正規木文法 G と G' が与えられた時に，G を G' に更新するために必要なコスト最小の編集操作列を求めることをいう．ここで，編集操作列とは「生成規則の追加・削除」等の編集操作の系列であり，各編集操作にはコストが付与される．例えば，正規木文法 $G' = (N',T',S',P')$ を考える．ここで，

$N' = \{R, T, M, E, F, L, Pcdata\}$
$T' = \{\text{staffs, staff, name, email, first, last, pcdata}\}$
$S' = \{R\}$
$P' = \{R \rightarrow \text{staffs}(T^*), T \rightarrow \text{staff}(M\ E),$
$M \rightarrow \text{name}(F\ M), E \rightarrow \text{email}(Pcdata),$
$F \rightarrow \text{first}(Pcdata), M \rightarrow \text{last}(Pcdata),$
$Pcdata \rightarrow \text{pcdata}(\varepsilon)\}$

このとき，上記 G と G'の差分は次のようになる．

- N において，F と L を追加
- T において，first と last を追加
- P において，(a) 生成規則 $M \rightarrow \text{name}(Pcdata)$ の右辺を $\text{name}(F\ L)$ に変更し，(b) 2 つの生成規則 $F \rightarrow \text{first}(Pcdata), M \rightarrow \text{last}(Pcdata)$ を追加

なお，2 つの正規木文法 G と G' が与えられた時に，G の内容をすべて削除して G' の内容を追加すれば G を G' に更新するための編集操作列が得られるが，そのような差分を抽出するのは無意味である．そこで，本研究ではコスト最小の編集操作列を差分として抽出する．

以上を踏まえて，本研究では，以下の手順で，正規木文法の差分抽出アルゴリズムを開発する．

(1) 正規木文法に対する編集操作(要素の追加・削除，要素名の変更など)を形式的に定義する．
(2) 得られた編集操作に基づいて，正規木文法の差分抽出問題の計算困難性について考察し，その証明を行う．
(3) 正規木文法の差分抽出問題が多項式時間可解となる十分条件を求める．そして，その条件の下で動作する，正規木文法の差分抽出を行うための多項式時間アルゴリズムを開発する．

(4) 得られたアルゴリズムを実装し,評価実験を行う．アルゴリズムの実装には，迅速な開発が可能な Ruby 言語を用いる．

４．研究成果

まず，正規木文法に対する編集操作として，以下を定義した．

- 非終端記号の追加・削除
- 生成規則の追加・削除
- 非終端記号の変更(リネーム)
- 非終端記号の置換
- 生成規則の右辺(正規表現)を更新するための操作．正規表現を木として表し，木に対する頂点の追加・削除・置換として定義する

次に，この編集操作に基づいて，NP 完全問題である 3-partition 問題からの帰着により，正規木文法の差分抽出問題が NP 困難であることを証明した．次に，この問題が多項式時間可解となる十分条件を求めた．その条件下では，生成規則 r の左辺の非終端記号を他の非終端記号 A に変更・置換できるのは，以下の条件が成り立つ場合のみである．

- r の右辺は更新されない
- 他の生成規則が左辺に A を用いていない

そして，この十分条件の下で，正規木文法のための差分抽出アルゴリズムを開発した．$G = (N,T,S,P)$ と $G' = (N',T',S',P')$ を正規木文法とする．本アルゴリズムによる，G と G' の差分抽出に要する時間計算量は次の通りである．

$$O(|P|\cdot|P'|\cdot|r_{max}| + |P\cap P'|\cdot|r_{max}|^3)$$

ここで，r_{max} は $P\cup P'$における生成規則の右辺のうち，最もサイズの大きなものを表す．以上で開発したアルゴリズムを Ruby 言語を用いて実装し，以下のスキーマを用いて評価実験を行った．

- relaxng.rng (http://relaxng.org/relaxng.rng)
- VoiceXML10full.rng (http://www.kohsuke.org/relaxng/voicexml.html.)

これらのスキーマに対して，ランダムに生成した編集操作列を適用し，更新されたスキーマを作成した．適用した編集操作列の長さを表 1 に示す．

表 1: スキーマと編集操作列長

	スキーマ	編集操作列の長さ
1a	relaxng.rng	12
1b	relaxng.rng	22
2a	VoiceXML10full.rng	21
2b	VoiceXML10full.rng	34

本アルゴリズム，および，既存の XML 差分抽出ツール(X-Diff および Diffmk)を用いて，元のスキーマと更新されたスキーマの差分抽出を行い，抽出された差分を比較した．その結果を表 2 に示す．

表 2：抽出された差分長

	本手法	X-Diff	Diffmk
1a	12	24	7(+5)
1b	22	187	14(+12)
2a	21	176	16(+9)
2b	34	278	26(+9)

この結果，本アルゴリズムにより抽出された差分は表 1 の編集操作列長と一致し，冗長さを含まない差分が抽出できているが，X-Diff や Diffmk では冗長な差分が抽出されていることが分かった．なお，Diffmk は，要素の削除を検出する機能を備えておらず，要素の追加と置換のみ検出することができるツールである．そのため，表 2 では Diffmk により抽出された要素の追加と置換の数を括弧外に記載し，要素の削除については報告者が手作業で数えたものを括弧内に記載している．実際の差分抽出において，削除を手作業で発見するのは非常に手間のかかる作業であるため，Diffmk をスキーマ間の差分抽出に用いるのは困難であると考えられる．

また，このアルゴリズムの応用として，スキーマ更新に応じて XPath 式を修正するアルゴリズムを開発した．開発したアルゴリズムを Ruby を用いて実装し,評価実験を行った．評価実験では,スキーマとして MSRMEDOC DTDs (versions 2.1.1 and 2.2.2) および NLM Journal Publishing Tag Set Tag Library DTDs (versions 2.3 and 3.0)を用い，XPath 式には XPath 式の生成ツールである XQGen から生成したものを用いた．その結果，実際のスキーマ更新において，XPath 式の修正が適切に行われていることを確認した．これらの成果により，スキーマ更新が生じた場合の XML データの管理がより容易に行えると考えられる．

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 2 件）

① Kazuma Hasegawa, Kosetsu Ikeda, and Nobutaka Suzuki, An Algorithm for Transforming XPath Expressions According to Schema Evolution, 査読有 , Proceedings of the First International Workshop on Document Changes: Modelling Detection, Storage and Visualization (DChanges 2013), 2013,8p, http://ceur-ws.org/Vol-1008/paper4.pdf

② Kazuma Horie and Nobutaka Suzuki, Extracting Difference between

Regular Tree Grammars, 査読有, Proceedings of the 28th ACM Symposium on Applied Computing (SAC 2013), 2013, pp.859-864
DOI:10.1145/2480362.2480527

〔学会発表〕(計 4件)

① 長谷川数馬, 池田光雪, 鈴木伸崇, スキーマ進化に伴う XPath 式修正アルゴリズムの提案, 第 5 回データ工学と情報マネジメントに関するフォーラム (DEIM2013), B2-2, 2013 年 3 月 3 日, 福島県ホテル華の湯
② 堀江和磨, 鈴木伸崇, 正規木文法間の差分抽出アルゴリズムの提案, 情報処理学会研究発表会報告, DBS-154-12, 2012 年 8 月 8 日, 名古屋大学
③ 長谷川数馬, 池田光雪, 鈴木伸崇, スキーマ進化に伴う XPath 式修正アルゴリズム, 情報処理学会第 74 回全国大会, 4P-6, 2012 年 3 月 7 日, 名古屋工業大学
④ 堀江和磨, 鈴木伸崇, 正規木文法の差分抽出問題に関する研究, 情報処理学会研究発表会報告, DBS-152-12, 2011 年 8 月 3 日, 立命館大学朱雀キャンパス

６．研究組織
(1)研究代表者
鈴木 伸崇 (SUZUKI, Nobutaka)
筑波大学・図書館情報メディア系・准教授
研究者番号：60305779