挿入を考慮した XML ラベリング手法の比較

Comparisons of XML Labeling Method Considering Insertions

小林 一仁♡ 横田 治夫♢

Kazuhito KOBAYASHI Haruo YOKOTA

近年 XML はその表現能力により様々な場面で使用されるようになった．そのため XML ドキュメントを検索する機会が増えている．効率的な検索のため，XML ドキュメントの各要素にラベルを付け，関係データベースに格納する手法が提案されている．しかし，自然数などを用いたラベリング手法では挿入時に問題がある．ノードの挿入が発生する度に多くのノードのラベル値を更新する必要が生じてしまい，挿入コストが非常に高くなる．我々は挿入ノードのコードを容易に作成できるコードを提案してきた．本論文ではこのコードを用いて XML ドキュメントをラベリンする．そしてシミュレーションにより，このコードを用いたラベリング手法と既存の挿入に考慮して数値の間を空けるラベリング手法とで比較を行い，提案手法の優位性を示す．

Recently, XML has commonly been used in many cases, because of its high ability of expression. It increases the situations of retrieving XML documents. For efficient retrieval, a number of methods labeling every node to store it in a relational database are proposed. But they have a problem of insertion. The labeling methods using continuous numbers make the cost of insertion very high. We proposed a code to make code assignment for newly inserted node of an XML document easy. In this paper, we compare our method with the other labeling method considering insert operations with sparse numbering. The simulation results indicate that our method is superior to the sparse numbering method.

1. はじめに

近年，様々な場面でデータを表現，交換するために XML が使われるようになってきている．これに伴い，XML ドキュメントを検索する機会が多くなり，効率のよい XML ドキュメント検索の実現が求められている．そのためには，XML をデータベースに格納することが必要とされている．中でも，関係データベースに XML ドキュメントを格納する手法が注目されている [1]．関係データベースに格納することで，関係データベースの様々な機能を利用することができる．

しかし，XML はタグにより表現された包含関係を持っていることから，XML ドキュメントを検索する際には，この包含関係に従った検索ができなければいけない．しかし，関係モデルのみではこの包含関係を表現することができない．そのため，包含関係を関係モデル上で表現する様々な手法が提案されている．主に，包含関係を木構造と見なして，木構造の先祖子孫判定手法を使うラベリング手法が使われている．図 1 に示す前順後順法 [2] や図 2 に示す Dewey Order[3] などがその例である．

♡ 正会員 東京工業大学大学院情報理工学研究科計算工学専攻
（現：(株) 日立製作所ソフトウェア事情部）
kkobayashi@de.cs.titech.ac.jp

♢ 正会員 東京工業大学 学術国際情報センター
yokota@cs.titech.ac.jp

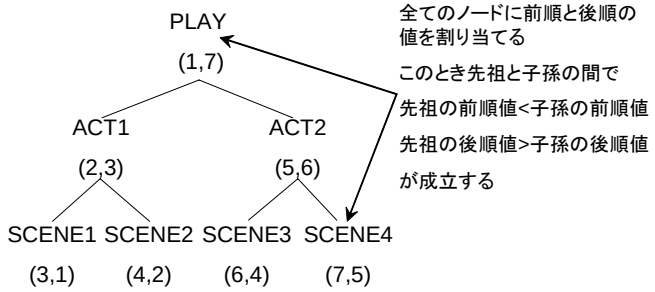

図 1: 前順後順法
Fig.1 Method of Preorder and Postorder

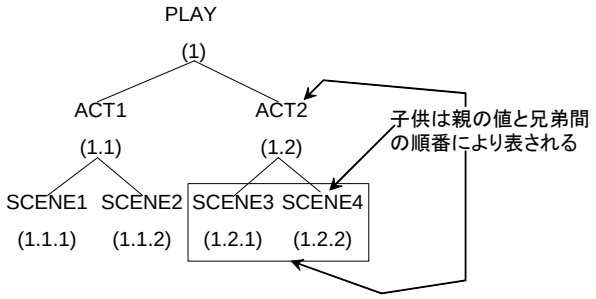

図 2: Dewey Order
Fig.2 Dewey Order

しかし，挿入が行われると，図 3 のように，多くのノードの値を割当し直す再構築をしなければいけない．関係データベースで多くの値をつけ直す処理はコストが非常に高い．そこで，本稿ではこれまでに提案してきた挿入ノードを容易に作成できるコードをを用いて実際に XML ドキュメントのラベリングをおこない，挿入を考慮した他手法と，検索と挿入の二つの値で比較を行う．

本稿の構成は次の通りである．まず 2 節で，これまでに提案されている関連研究を述べる．次に 3 節で VLEI コードの概略について説明を行う．4 節では包含関係を表現するラベリング手法を挙げ，それらに対してどのように VLEI コードを適用するのか説明を行う．5 節で実験とその考察を行い，6 節で本稿をまとめる．

2. 関連研究

XML ドキュメントを前順後順などの値を用いて包含関係を表現し，関係データベースに格納する場合，再構築のコストを下げるためにに前順後順などの値の間を開けることで，ある程度の挿入を容易に行う方法として [4] や [5] などがある．[4] では最初に予めノード間にスペースを作っておき，挿入によりスペースが狭くなってきたら動的にスペースを空ける手法を提案し，[6] で性能評価をしている．また [5] では挿入するノードを浮動小数点で表現する手法 (QRS) を提案している．また，オーバーフローを起こしたときにバルクロードを高速に行う手法も提案している．本稿では，提案手法と上記スペースを空ける手法との比較を行う．

包含関係を表現するための XML のラベリング方法として，前順後順法及び Dewey Order 以外では，[7] は XML の木構造をバイナリ tree にマッピングし，その値を用いて効率よく containment ジョインを行う手法を提案している．しかし，これは挿入に関してまったく考慮していない．

3. VLEI コードの概略

本節では我々の提案している VLEI(Variable Length Endless Insertion) コードの概略を述べる．詳しくは [8]

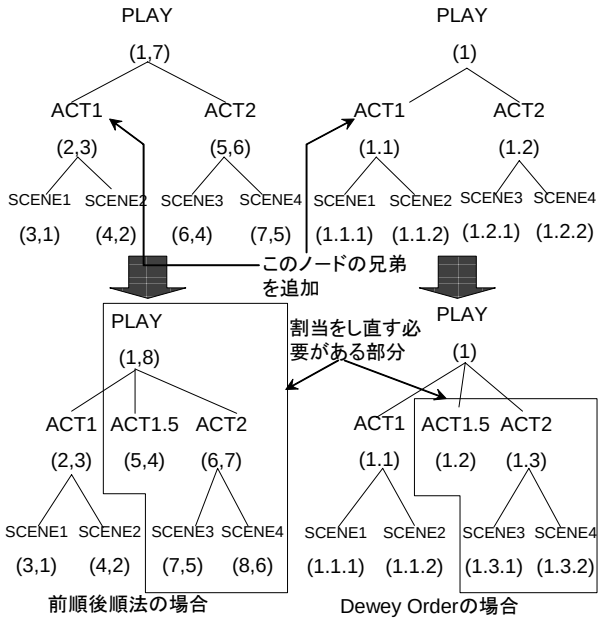

図 3: 挿入が行われたときの再構築の例
Fig.3 An Example of Reconstruction by Insertion

を参照されたい．VLEI コードは 1 から始まる 0,1 の可変長の bit 列で，次の大小関係を満たす．

定義 1. VLEI コードの大小関係
v をある VLEI コードとして，以下の大小関係が成立する．
$v \cdot 0 \cdot \{0|1\}^* < v < v \cdot 1 \cdot \{0|1\}^*$ □

1 を基準点としてこの値よりも小さいか大きいかを 0 と 1 をつけて表現する．例として，11 は 111 より小さく，110 よりは大きいという大小関係を持っている．

この VLEI コードを用いれば図 4 のアルゴリズムにより，あるコードとあるコードの間を示すコードを容易に作成することができる．たとえば，11 と 111 の間に要素を挿入するためこの間を表したい場合は図 4 のアルゴリズムにより 1110 が得られる．これは VLEI コードの定義から $11 < 1110 < 111$ が成立する．このように挿入する要素の値を他の要素の値を変更することなく容易に求めることができる．

4. VLEI コードを用いた XML ラベリング

VLEI コードを利用した Dewey Order で XML をラベリングすることを試みる．Dewey Order は兄弟間の順番を示す数字とその子供であることを示す識別子によって構成される．そこで Dewey Order の兄弟間の順番を表現しているところに VLEI コードを利用することで挿入が起こってもほかノードに影響を与えることなく挿入ノードの値を作成できる．そこで以下のように定義した識別子付き VLEI コードで Dewey Order を表現する．

定義 2. 識別子付き 8 進 VLEI コード
Dewey Order の識別子を 9 とする．兄弟間の順番を VLEI コードに変換したものを 8 進数の数値に変換する．識別子付き 8 進 VLEI コードを次のような 10 進数の数値で表現する．10 進数で表現することで関係データベースに数値として保存でき，また先祖子孫関係を判定する際に簡単な判定方法で求めることができる．

アルゴリズム makeInsertValue(v_l, v_r)

入力: 挿入する要素の左側のコード v_l,
右側のコード v_r
(但し $v_l < v_r$)

出力: 挿入する要素のコード v_i

```
if length(v_l) ≤ length(v_r)
    v_i = v_r · 0
else
    v_i = v_l · 1
endif
return v_i
```

図 4: アルゴリズム 1. 挿入する要素の決定
Fig.4 Algorithm1. Making Value of Inserting Node

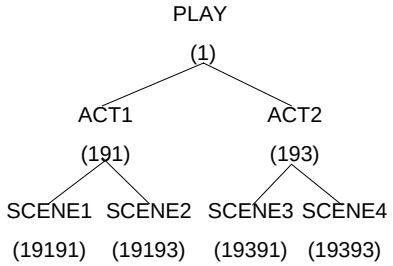

図 5: Dewey Order への VLEI の適用
Fig.5 Applying VLEI code to Dewey Order

ノードの値=”親の値”+9+”「VLEI コードで兄弟間の順番を表現した bit 列」を 8 進数に変換した値”
□

このように定義することで Dewey Order を 10 進数で表現でき，int などの整数型の格納領域に格納できる．また VLEI コードで兄弟間の順番を表現した bit 列は図 6 のアルゴリズムで容易に求めることができる．兄弟間の VLEI コードを 8 進数で表現するため識別子は 8 か 9 を使うことができる．今回は 9 を識別子として使用する．図 2 に対してこの定義で番号付けを行うと図 5 のようになる．

この Dewey Order で子孫を探すときには先祖の VLEI コードに 9 を付けたもので始まる VLEI コードを探せばよい．逆に親を検索する際には子供の VLEI コードから一番下の桁の 9 から一番下の桁の数を取った VLEI コードを検索すればよい．またその上の親を探すときには下から 2 番目の桁の 9 から一番下の桁の数まで取った VLEI コードを検索すればよい．例えば，図 5 の中で，ACT1 の子孫を捜すときは VLEI コードが 1919 で始まるものを探せばよい．また SCENE3 の親を探すときは VLEI コードが 193 のものを探せばよい．さらに SCENE3 の 2 つ上の親を探すときは VLEI コードが 1 のものを探せばよい．

挿入する値は図 7 のアルゴリズムで決める．親の子供の中で一番長さが小さいものから順にまだ使われているかどうかを調べ，使われていない値を探す．使われていたらその値よりも小さく，長さが 1 大きい値について再帰的に調べていく．

5. 実験

識別子付き 8 進 VLEI コードを用いた Dewey Order と既存手法の前順後順法で値と値の間をあける手法のそれぞれの包含関係を表現するための値を関係データベースに格納し，

アルゴリズム NNum2VLEI(a_i,N)

入力: 自然数 a_i,
　　要素の数 N

出力: VLEI コード v_i

int $m = \lceil log_2 N \rceil$
int $P = 2^m$
variable bit $v_i = 1$
int $x = a_i - P$
while($x \neq 0$)
　　$P = \frac{1}{2}P$
　　if($x > 0$)
　　　　$v_i = v_i \cdot 1$
　　　　$x = x - P$
　　elseif($x < 0$)
　　　　$v_i = v_i \cdot 0$
　　　　$x = x + P$
　　endif
endwhile
return v_i

図 6: アルゴリズム 2. 自然数から VLEI コードへのマッピング
Fig.6 Algorithm2. Mapping from Natural Number to VLEI code

検索と挿入の性能を測定し，比較をする．以降識別子付き 8 進 VLEI コードを用いた Dewey Order を VLEI(Dewey) と略し，既存手法の値と値の間をあける手法を Sparse と略す．また今回データベースには包含関係を表現する値と，ノードの名前，値，そして各ノードを特定するための ID を格納した．格納対象の XML ドキュメントは文書構造を持っている The Plays of Shakespea [9] の hamlet.xml を関係データベースに格納した．また実験環境は以下の通りである．

CPU	Celeron 1.7Ghz
HDD	Seagate ST340016A
Memory	PC-2100 512MB
OS	Windows XP Professional
DBMS	MySQL
Java	J2RE1.4.2

5.1 検索

hamlet.xml に対してそれぞれの手法で 1 回の先祖子孫関係を判定する XPath[10]/PLAY//LINE を検索した時と同じ結果になるような SQL を 100 回発行し結果が得られるのに要した時間を測定した．結果は図 8 のようになった．

5.2 挿入

hamlet.xml を関係データベースに格納し，ある要素の子供に人工的に作成した要素を 1 個追加する挿入を 1000 回行った．要素を選ぶときに theta=1 の Zipf 分布 [11] により作成した乱数を用いた．Zipf 分布は確率分布の一種であり，データベースのアクセス分布のモデル化によく用いられる．引数として theta をとり，この値が大きくなればなるほど偏りが大きくなるような確率分布である．各手法でデータ長

アルゴリズム:insertNodeInDewey(a,b)

入力: Dewey Order における VLEI コードの配列 a
　　挿入のタイプ b

出力: 挿入するノードの VLEI コードの値 c

if(b==子供)
　c="a"+91;
else if(b==兄弟)
　c="a"+1;
endif
while(true)
　if(c がまだ使われていない値である)
　　break;
　endif
　c="c"+0;
endwhile

図 7: アルゴリズム 3. 識別子付き 8 進 VLEI コードにおける挿入する値の決定
Fig.7 Algorithm3. Making Value of Insertion Node in Case of Ocatal VLEI code with suffix

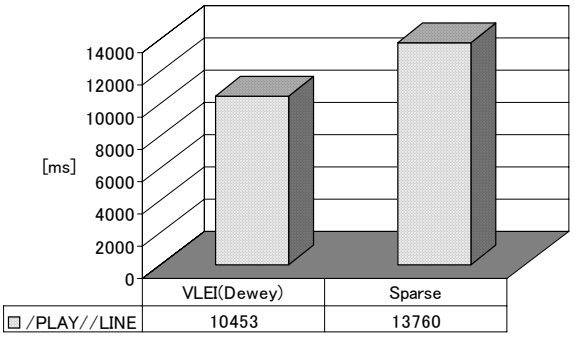

図 8: 1 回の先祖子孫関係の判定を 100 回行ったときに要した時間
Fig.8 Time of one ancester-descendant detection × 100

を超え，オーバーフローした場合はデータベースを再構築をした．Dewey Order に格納する際には BIGINT を利用した．また比較のため予め前順と後順に間を空けて格納する前順後順法を比較として測定した．[4] によると間を空ける際に間隔が $[\frac{Width}{|T|+|X|+1}]$($Width$ は使用可能数の最大値，T は更新される XML 木，X は挿入 XML 木) である時が最もよい均衡状態なので，この値で間隔を空けた．また動的に値の割当をし直すことはしなかった．また VLEI コードに偏りが生じても偏り制御をしなかった．結果は図 9 のようになった．

5.3 考察

検索において VLEI(Dewey) が Sparse よりも 24% 速い．これは Dewey Order を用いたラベリング手法では親子関係を判定する際に Dewey Order の値のみを評価するのに対し，前順後順法では前順と後順の 2 つの値を評価しなければいけない．よってこのコストが減ったため速くなった．

挿入に関して VLEI(Dewey) が要した時間は Sparse の $\frac{1}{5}$ と凌駕している．これは再構築の数が関係している．それぞれの手法における再構築の数は下の表の通りである．

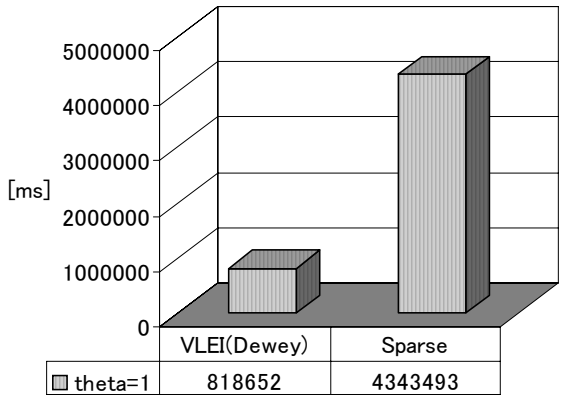

図 9: Zipf 分布 theta=1 の時 1000 回の挿入に要した時間
Fig.9 Time of 1000 Insertion Choosing Position by Zipf's low theta=1

Sparse	6
VLEI(Dewey)	3

挿入で要した時間の 99% を再構築に要した．再構築の数が減った理由として提案手法では値格納に 64bit 整数型を利用した．間をあける手法では 32bit 整数型を利用したが前順と後順の 2 つの値を利用しなければいけないため結果として同じ記憶領域を必要することとなる．よって同じ記憶領域を利用すると仮定した場合，提案手法では間をあける手法より記憶領域を超えにくくすることができる．

6. まとめ

我々の提案している VLEI コードを XML ラベリング手法である Dewey Order に適用するため，識別子付き 8 進 VLEI コードを提案し，実際に XML ラベリング手法に適用して，検索時間と挿入時間の計測実験を行った．この実験により既存の間をあける手法と比べ要素の挿入に対して強いことが示された．また VLEI コードを用いた手法の XPath 検索速度は前順後順法によるものより速いことを示すことができた．以上より，識別子付き 8 進 VLEI コードを用いた Dewey Order が既存手法よりも優れていることを示すことができた．

今後の課題として，今回の実験は実際の XML ドキュメントの検索や更新について考慮しなかった．そこで [12] などのような XML ベンチマークなどの研究に基づくベンチマークを行い VLEI コードが実際の更新に対しても強いことを示す．また今回は DeweyOrder を識別子付き 8 進 VLEI コードで表現し 64bit の格納領域に格納したが，XML 木において根から葉までの長さが深くなると 64bit では表現できなくなる．そこで DeweyOrder を識別子付き 16 進 VLEI コードで表現し，BLOB などのバイナリデータに格納する．また，Dwey Order 以外のインデックス構造またはラベリング手法を検討する．

[謝辞]

VLEI コードの Dewey Order への適用に関して奈良先端科学技術大学の天笠 俊之先生に助言を頂いた．ここに感謝の意を表します．また本研究の一部は，文部科学省科学研究費補助金特定領域研究 (15017233)，独立行政法人科学技術振興機構 CREST，及び 21 世紀 COE プログラム「大規模知識資源の体系化と活用基盤の構築」の助成により行われた．

[文献]

[1] Igor Tatarinov, Stratis Viglasand Kevin S. Beyer, Jayavel Shanmugasundaram, Eugene J. Shekita, and Chun Zhang. Storing and querying ordered xml using a relational database system. In *Proc. of SIGMOD Conf.*, pages 204–215, 2002.

[2] Paul F. Dietz. Maintaining order in a linked list. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 122–127, 1982.

[3] Online Computer Libraly Center. Introduction to the dewey decimal classification. http://www.oclc.org/oclc/fp/about/about_the_ddc.htm.

[4] 江田毅晴，天笠俊之，吉川正俊，植村俊亮．Xml 木のための更新に強い節点ラベル付け手法．In *DBSJ Letters,No.1 in Vol.1,2002*, 2002.

[5] Toshiyuki Amagasa, Masatoshi Yoshikawa, and Shunsuke Uemura. Qrs: A robust numbering scheme for xml documents. In *19th International Conference on Data Engineering (ICDE 2003)*, pages 705–707, 2002.

[6] 江田毅晴，天笠俊之，吉川正俊，植村俊亮．XML のための動的範囲ラベル付け手法:その評価および XRel への適用について．In **情処研報**, **情報処理学会**, *DBS-129-16*, 2002.

[7] Wei Wang, Haihng Jiang, Hongjun Lu, and Jeffrey Xu Yu. Pbitree coding and efficient processing of containment join. In *International Conference on Data Engineering*, pages 391–402, March 2003.

[8] 小林一仁，小林大，横田治夫．挿入制限のない範囲ラベリング用コード．In **信学技報**, **電子情報通信学会**,*DE2003-13*, 2003.

[9] Jon Bosak. The plays of shakespeare in xml. http://www.oasis-open.org/cover/bosakShakespeare200.html.

[10] World Wide Web Consortium. Xml path language . http://www.w3.org/TR/xpath.

[11] William J. Reed. The Pareto, Zipf and other power laws. http://linkage.rockefeller.edu/wli/zipf/reed01_el.pdf.

[12] Kanda Runapongsa, Jignesh M.Patel, H.V.Jagadish, Yun Chen, and Shurug Al-Khalifa. Michigan benchmark: Towards xml query performance diagnostics. In *Proceedings of the 29th VLDB Conference*, 2003.

小林 一仁　Kazuhito KOBAYASHI
平 16 東工大大学院・情報理工・計算工・修士課程了．(株) 日立製作所 ソフトウェア事業部．日本データベース学会正会員．

横田 治夫　Haruo YOKOTA
昭 55 東工大・工・電物卒．昭 57 同大大学院・情報・修士課程了．同年富士通 (株)．同年 6 月 (財) 新世代コンピュータ技術開発機構研究所．昭 61(株) 富士通研究所．平 4 北陸先端大・情報・助教授．平 10 東工大・情報理工・助教授．平 13 東工大・学術国際情報センター・教授．工博．電子情報通信学会，情報処理学会，人工知能学会，IEEE，ACM 各会員．