XPath 充足可能性問題の多項式時間可解な部分問題に関する研究*

福島雄司（学籍番号 200721554）
研究指導教員：阪口哲男
副研究指導教員：鈴木伸崇

１．まえがき

DTD D と XPath 式 p に対して，p の問合せ結果が空でなく, かつ D に関して妥当な XML データが存在し得るか否かを決定する問題を XPath 充足可能性問題という．充足可能でない XPath 式は，その問合せ結果が常に空となるので，実行を回避することが望ましい．

しかし，XPath 充足可能性問題は一般に決定不能であることが知られている[1]．このため，効率よく解くことのできる，同問題のできるだけ広い部分問題を発見し，その部分問題を効率よく解くためのアルゴリズムを構成することは重要な課題である．本研究では，このような部分問題を発見し，それを解くための多項式時間アルゴリズムを構成する．更に，アルゴリズムに関する評価実験を行う．

２．諸定義

本研究では，XPath 式として，XPath の仕様[3]における絶対ロケーションパスで，軸を child，parent，descendant，following-sibling，preceding-sibling の 5 種に限定し，ノードテストは要素名のみかつ述語を用いないものを考える．D を DTD，p を XPath 式とする．D に関して妥当，かつ，p の問合せ結果が空でない XML データが存在するとき, p は D の下で充足可能であるという．XPath 充足可能性問題とは，p が D の下で充足可能であるか否かを決定する問題のことをいう．D の各要素の内容モデルに同じ要素名が 2 回以上出現しないとき，D は duplicate-free[2]であるという．

* “A Study on Polynomial Time Solvable Subclass of XPath Satisfiability Problem ”
by Yuji FUKUSHIMA

３．XPath 充足可能性問題の計算複雑さ

まず, descendant 軸を用いない場合における XPath 充足可能性問題の計算複雑さについて, 3SAT からの帰着により以下の 2 つの結果を示した．

定理 1：XPath 充足可能性問題は，XPath 式の軸を child と parent のみに限定しても NP 完全である． □

定理 2：XPath 充足可能性問題は，XPath 式の軸を child，following-sibling，preceding-sibling のみに限定しても NP 完全である． □

次に, descendant 軸を含む場合について考察し，上と同様 3SAT からの帰着により，次の結果を示した．

定理 3：XPath 充足可能性問題は，XPath 式の軸を child，parent，descendant のみに限定し，かつ DTD を duplicate-free かつ*を含まないものに限定しても NP 完全である． □

４．XPath 充足可能性問題の検証アルゴリズム

本研究で構成したアルゴリズムは，XPath 式のロケーションステップを前から順に処理して木を構成しながら，ノードの妥当性を DTD に照らし合わせて検証していく．ロケーションステップの最後まで木を構成することができれば yes，そうでなければ no を返す．descendant 軸は 1 つの軸から複数の異なるノードに到達できるので，到達可能なノード数に応じて木は複数構成される場合がある．D を DTD，p を XPath 式とする．このアルゴリズムの時間計算量は, descendant 軸を含まない場合，D が duplicate-free であるとの仮定の下で $O(|p|(|p| + |D|))$である．また，descendant

軸を含む場合，D が duplicazte-free，*を含まず，かつ非再帰であるとの仮定の下で $O(|\Sigma|^2 \cdot |D|^2 \cdot |p|)$である．ここで，$\Sigma$ は D に出現する要素名の集合を表す．例として，次の DTD と XPath 式 p を考える．

<!ELEMENT list (item | list)* >
<!ELEMENT item (a)>
<!ELEMENT a (#PCDATA)>

p = /child::list/child::item/parent::list/child::item /parent::list/descendant::a

アルゴリズムにより木が構成される過程を図 1 に示す．次のロケーションステップの処理対象となるノードをカレントノードと呼ぶ．ロケーションステップの最後まで木を構成できたので，アルゴリズムは yes を返す．

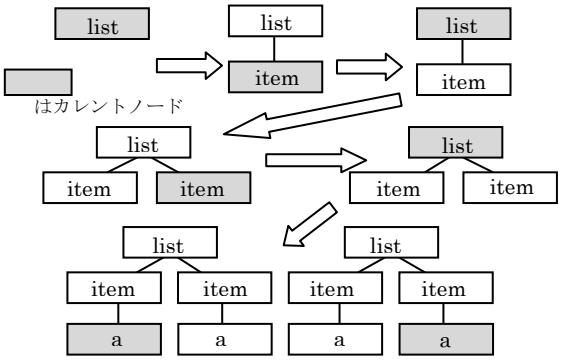

図 1：木の構成過程

５．評価実験

アルゴリズムを Java で実装し，評価実験を行った．まず，XMark を用いて異なるサイズの XML データを生成する．次に，充足可能な XPath 式と充足可能でない XPath 式をそれぞれ 15 個用意し，充足可能な XPath 式に対して Overhead Ratio[4]，充足可能でない XPath 式に対して Saving Ratio[4]をそれぞれ求める．XPath 式の充足可能性の検証に要した時間を c，検索を実行するのに要した時間を e とすると，Overhead Ratio は$(c+e)/e$，SavingRatio は$(e-c)/e$と定義され，いずれも 1 に近いほど望ましい．

実行環境は CPU：Intel Core2 Duo 1.60GHz，メモリ：2GB，OS：Windows Vista Business，使用言語：Java 2 SDK 1.6.0，XPath 実行環境：eXist 1.2.2 および SAXON b.9.1 である．

結果を図 2 に示す．XPath 式の実行にある程度の時間を要する状況下においては，XPath 式の実行前に充足可能性検証を行うことは有効であると考えられる．

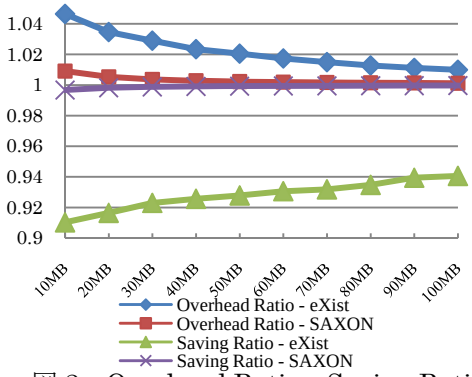

図 2：Overhead Ratio，Saving Ratio

６．むすび

本研究では，XPath 充足可能性問題について，時間計算量に関する複雑さについて考察し，同問題を解くための多項式時間アルゴリズムを構成した．今後の課題として，より多くの軸や DTD 以外のスキーマ言語に対応したアルゴリズムの構成が挙げられる．

文献

[1] M. Benedikt, W. Fan, and F. Geerts, *XPath satisfiability in the Presence of DTDs*, Journal of the ACM, Vol.55, Issue 2, Article 8, 79 pages, May 2008.

[2] M. Montazerian, P. T. Wood, and S. R. Mousavi, “XPath Query Satisfiability is in PTIME for Real-World DTDs,” Proc. Xsym 2007, pp.17-30, 2007.

[3] J. Clark and S. DeRose, eds., XML Path Language (XPath) Version 1.0, http://www.w3.org/TR/xpath/.

[4] L. V. S. Lakshmanan, G. Ramesh, H. Wang, and Z. Zhao, “On Testing Satisfiability of Tree Pattern Queries”, Proc. VLDB, pp.120-131, 2004.