

DTD グラフに基づく XPath 充足可能性問題の判定手法

塚田 貴裕

XML(Extensible Markup Language) はデータの生成が容易で可読性も高く、近年急速に普及が進んでいる。大量の XML データを蓄積管理する場合、格納するデータの構造を DTD(Document Type Definition) などのスキーマを用いてあらかじめ定義しておき、スキーマに妥当なデータを生成・蓄積するのが一般的である。また、XML データへの問合せ言語として XPath(XML Path Language)がよく使われる。

DTD に妥当な XML データに対して XPath 式を用いて検索を行っても、XPath 式を満たす結果が存在し得ない可能性がある。例えば、DTD で指定されていない要素を含む XPath 式や、DTD とは異なる順序で要素を指定している XPath 式による問合せはその結果が必ず空になる。このように、DTD に妥当などの XML データに問合せをしても結果が必ず空になる XPath 式のことを充足不能な式という。充足不能な XPath 式を実行しても、検索結果は常に空となるため実行しないことが望ましい。そこで、XPath 式を実行する前に充足可能性の判定を行い、その XPath 式が充足可能な式であるか否かを判定することで、充足不能な XPath 式の実行を回避し無駄な処理時間を減らすことが可能である。

しかし、XPath 式の充足可能性問題は一般に決定不能であり、XPath 式に様々な制約を加えても効率よく解けないことが知られている。そこで、本研究では一部の充足不能な XPath 式に関して検出漏れが生じることを許した上で、多項式時間で動作するアルゴリズムを考案した。本研究のアルゴリズムで対象とする XPath 式は、軸を child 軸、parent 軸および descendant-or-self 軸に限定し、ノードテストは要素のみに制限する。また、述語は child 軸のみを用いた相対ロケーションパスのみ使用できるものとする。

アルゴリズムを Ruby を用いて実装し、それに関する評価実験を行った。実験内容は、DTD に対して充足可能な XPath 式と充足不能な XPath 式を用意し、それぞれの式に対して充足可能性判定時間及び XPath 式実行時間を測定するものである。実験の結果、特に XML データのサイズがある程度大きい場合 XPath 式の実行にかなりの時間を要するが、充足可能性判定はわずかな時間で行うことが可能である。よって、充足可能性判定を事前に行うことによって、充足不能な XPath 式の実行を回避し無駄な実行時間を減らすことが可能であることが示せた。

(指導教員 鈴木伸崇)