

スキーマ進化に伴う XPath 式修正アルゴリズム

長谷川 数馬

XML(Extensible Markup Language) はあらゆるデータを記述するフォーマットとして様々なシステムで用いられている。大量のXMLデータを蓄積・管理する場合、DTD(Document Type Definition)などのスキーマ言語を用いてあらかじめ格納するデータの構造を定義しておき、スキーマの定義にしたがっている(妥当な)データを生成・蓄積するのが一般的である。スキーマは、時間の経過に伴い要素の削除や新たな要素の追加などの更新操作により内容が変化する。これをスキーマ進化と呼ぶ。また、XMLデータへの問合せ言語としてXPath(XML Path Language)がよく使われている。

ここで、DTD D に妥当なXMLデータに対しXPath式 p で問合せを行う場合を考える。また D がスキーマ進化したと仮定し、これを D' とする。このとき、 D に対し妥当であったXMLデータは D' に対し妥当になるように修正される必要がある。同様に、 D の構造が変化したことにより、 p は D' に対し妥当でなくなる可能性がある。この場合、 p を D' に対し妥当になるように修正する必要がある。また、 p はXMLデータの構造を利用して問合せを行う。そのため、スキーマ進化の際に行われる更新操作によりXMLデータの構造が変化する場合、XMLデータやそれに対する修正内容を参照せずとも、 D に対する更新操作のみから p に対する修正を推測できることが多い。しかし、 p の記述内容や D の構造が複雑な場合、 p を手作業のみで適切に修正するのは容易でない。

そこで本研究では、DTDに対して適用された更新操作に基づいてXPath式をスキーマ進化後のDTDに妥当となるよう修正するアルゴリズムを提案する。ただし、1回のスキーマ進化に対して、スキーマに対する更新操作の種類・回数及びXPath式の構文に制限を全く課さない場合、更新操作に対してXPath式に修正が必要か否かを判定することは決定不能であることを先行研究の結果から導くことができる。このため、更新操作及びXPath式の構文に次の制限を課す。更新操作は要素の削除、挿入、抜き取り、要素名の変更の4つとし、1回のスキーマ進化では4つの更新操作のうち1つしか使用しないとする。XPath式は軸をchild, descendant-or-self, parent, ancestor-or-self, preceding-sibling, following-siblingに限定し、ノードテストは要素のみかつ述語は使用しないとする。

本アルゴリズムをRubyで実装し、評価実験を行った。その結果本アルゴリズムを用いることで、スキーマ進化により妥当性を失ったXPath式を、スキーマ進化後のDTDに対し妥当となるように適切に修正できる見込みが得られた。

(指導教員 鈴木伸崇)