# 二階正規パターンに対するマッチングアルゴリズムに関する研究

数理第七研究室　修士二年　松本嘉夫
指導教員：武市正人教授

## 1　はじめに

### 1.1　背景

XML は W3C [1] が仕様を規定している木構造をなすデータ形式であり、その汎用性の高さから近年広く使われるようになってきている。しかしその処理系は標準的と言えるほど広まっているものは未だなく、数多くの XML 変換処理系が研究されている。

特に現在最も多く使われているのが XSLT という言語であり、XSLT においてはノードの指定に XPath と呼ばれる言語が用いられている。XPath では URL やディレクトリ構造のようなパス形式で XML の木構造をたどることでノードを指定する。XPath には “//”(任意の子孫ノード)、“.”(カレントノード)、“..”(親ノード) といった略記法が準備されている他、“ancestor”(任意の先祖ノード) という記法も使える。これらにより木構造の親子関係を直接的に表すことができる。一方 XML 文書中に出現するタグの種類やその出現回数を規定するのによく正規表現が用いられるが、XPath にはこれに対応する記法が無いので兄弟要素の連接関係やタグの出現回数を表すのは難しい。また XPath はノードを抽出する際に、ルートノードからそのノードまでのパスなどの、抽出するノード以外の情報 (これを “文脈”(context) と呼ぶ) を捨ててしまう。だから例えば XML 文書の一部のみを削除するような変換を行う際には “//” と “*”(ワイルドカードを表す) を用いれば、削除するノードは簡単に指定できるがそれ以外の部分をそのまま再構成してやらなければならない。

### 1.2　目的

本研究では XML をはじめとする木構造から一部を抽出する、あるいはデータ変換を行う際のノードの指定方法として二階正規パターンと呼ばれる手法を用いるものを提案した。

一階パターン変数が木構造に対する部分木を抽出するのに対し、二階パターン変数は部分木以外の残りの部分、即ち文脈を抽出する。また二階パターンは引数の出現する位置については制限しないので、任意の深さのノードを簡単に抽出することもできる。

更に正規二階パターンは二階パターン変数の引数に正規表現の出現を許すものである。よって抽出したいノードの指定に際して、兄弟要素の連接関係やその出現回数を直感的に表すことができる。

このように二階正規パターンは XPath よりも変換に関する記述力に勝っており、これを用いることでより直感的な部分木の指定やより簡潔な XML の変換が可能になると考えられる。そこで本研究では二階正規パターンの構成法 (すなわちシンタックスとセマンティクス) を定め、二階正規パターンを用いたマッチング判定のアルゴリズムを示した。またこのアルゴリズムの効率が現実的であることを示すため、実際に二階正規パターンのマッチャーを実装しマッチングが現実的な時間で行われることも示した。

### 1.3　関連研究

本研究と最も深く関連するのが Hosoya と Pierce の研究である。彼らは正規表現を許した型システム及びこれを用いた一階正規パターンによって XML 要素を抽出する技法を提案し [3, 5]、これを実装する処理系 XDuce を公開した [4]。また Benzaken らは、XDuce に高階関数、関数のオーバーロードなどを導入した処理系 CDuce [2] を公開した。

Temur Kutsia は横方向 (兄弟要素) だけでなく、縦方向 (親子要素) にも正規表現を導入し表現力を増している。またマッチングには、項書換系の観点に基づいた Rule Based な手法を用いている [6]。

二階パターンはもともとプログラム変換の分野において発展してきた手法である。Yokoyama らは二階パターンのうちマッチングを行うと解が高々1 つしか存在しないパターンのクラスを定め、またそのマッチング判定が線形時間で終るための戦略を導入した [7]。本研究における二階パターン変数に代入される関数の制限、マッチングの戦略はこれを参考にしている。

## 2　二階正規パターンマッチ

### 2.1　マッチングアルゴリズム

本研究では Hosoya らの (一階) 正規パターンを拡張し、二階正規パターンを構成した。そしてこれに対するマッチングアルゴリズムを考案した。このアルゴリズムは複数のルールからなっており、これらのルールを再帰的に項とパターンの組に施してゆくものである。また複数のルールが適用可能の場合はどのルールを優先的に適用するかによってアルゴリズムが返す代入が異なるが、ユーザーにとって利便性の高いと思われる正規表現に対する first match と longest match を実現するルールの適用順序も考案した。

表 1: 実行時間

| ノード数 | $P1$ | $P2$ | $P3$ | $P4$ |
|---|---|---|---|---|
| 1000 | 0.125 | 0.171 | 0.334 | 0.166 |
| 5000 | 0.257 | 0.487 | 1.826 | 1.852 |
| 10000 | 0.445 | 1.293 | 6.592 | 9.561 |
| 15000 | 0.695 | 2.381 | 12.933 | 23.223 |
| 20000 | 0.920 | 3.946 | 21.449 | 47.689 |

### 2.2 アルゴリズムの性質

まずアルゴリズムの停止性を証明した。即ち任意の項 $t$ とパターン $P$ に対し、$t$ の $P$ に対するマッチング判定は必ず停止する。

またアルゴリズムの健全性を証明した。即ち $t$ の $P$ に対するマッチング判定でアルゴリズムが $\sigma$ なる代入を返したとすると、その代入は必ず $t = V\sigma$ を満たす。

二階パターンマッチは NP 困難に属す問題でありパターンの大きさ、より具体的には二階パターン変数の引数の個数とネスト回数に対して、マッチング判定は指数時間になることが分かっている。しかしユーザーが用いるパターンは小さいものであると考えられる。

## 3 評価実験

我々の提案した二階正規パターンのマッチングルールの有用性を評価するために、これを実装するプログラムを作成し実行時間に対する評価実験を行った。

実験ではノード数 $n$ を指定しランダムに生成した項に対してマッチングを行い、その実行時間を調べた。パターンは

$P1$ = `(p (<a>[]+, <z>[]))`、
$P2$ = `(p (<a>[]+, <z>[]) (<b>[]+, <y>[]))`、
$P3$ = `(p (<a>[]+, <z>[]) (<b>[]+, <y>[])`
`(<c>[]+, <x>[]))`、
$P4$ = `(p (<a>[],(q (<b>[]+, <z>[]))))`

の 4 通りを試した。これらはいずれも「二階パターンの引数に"+"のついた正規パターンが含まれる」という本研究が提案する新規性を反映しており、$P2$, $P3$ では二階パターンの引数が順に増えており、$P4$ では二階パターンがネストしている。

結果は表 1 の通りであり、いずれも現実的な時間でマッチング判定は終了した。

## 4 結論

### 4.1 まとめ

XML 変換を目的とすることを想定した二階正規パターンを定義し、その構成とマッチングアルゴリズムを提案した。本研究における二階正規パターンの役割は、項の中で二階パターン変数の引数が現れる箇所を探し出しそこを抽象化することで、引数とそれ以外の部分 (文脈) とに分けることである。これにより直感的な変換の記述が可能になった。またこのアルゴリズムの停止性と健全性を証明した。

更に、このアルゴリズムを用いる二階正規パターンのマッチャーを実装し、評価実験によってマッチングが現実的な時間で行われることを確かめた。

### 4.2 今後の課題

項 $t$ とパターン $P$ に対し $t = P\sigma$ なる代入 $\sigma$ が存在したときにアルゴリズムは必ずその代入を見つけられるかという性質を完全性と考えることができる。本アルゴリズムは完全性も満たすと予想しているが、今回その証明を与えることはできなかった。これは今後の課題である。

関連研究の [2] や [3] においてはパターンに対し型が定義されており静的な型チェックが可能になっている。本研究では型付けを考慮してこなかったが、二階パターンにも型付けが可能か、更に型同士の演算 (和、差、積) の計算が可能か、更にその計算効率はどれほどなのか、等を考えることは今後の課題である。

## 参考文献


[1] W3C (World Wide Web Consortium) `http://www.w3.org/`.

[2] V. Benzaken, G. Castagna, and A. Frisch. CDuce: An XML-Centric General-Purpose Language. In *Proceedings of 2003 ACM SIGPLAN International Conference on Functional Programming*. ACM Press, 2003.

[3] H. Hosoya and B. C. Pierce. Regular Expression Pattern Matching for XML. *ACM SIGPLAN Notices*, 36(3):67–80, 2001.

[4] H. Hosoya and B. C. Pierce. XDuce: A Statically Typed XML Processing Language. *ACM Transactions on Internet Technology*, 3(2):117–148, 2003.

[5] H. Hosoya, J. Vouillon, and B. C. Pierce. Regular Expression Types for XML. *ACM SIGPLAN Notices*, 35(9):11–22, 2000.

[6] T. Kutsia. Context Sequence Matching for XML. In *Proceedings of the 1st International Workshop on Automated Specification and Verification of Web Sites*, pages 103–119, Valencia, Spain, 2005.

[7] T. Yokoyama, Z. Hu, and M. Takeichi. Deterministic second-order patterns. *Information Processing Letters*, 89(6):309–314, 2004.