I_046

アスペクト指向に基づく ソフトウェアの非侵襲的利用法を用いた数式認識ツールの開発

Development of Tool for Recognition of Mathmatical Expression by Non-Invasive Software Reuse based on Aspect Oriented Programming

木村 裕樹† Yuki Kimura
古賀 雅伸† Masanobu Koga

1. はじめに

近年, 論文等の文書を作成する際のファイルフォーマットとしてPDFが多く用いられている. この理由として, PDFは情報漏洩防止策が充実しており, 描画環境を選ばずに美しい描画が可能であることが挙げられる. PDFは複数のオブジェクトから構成され, 各オブジェクトは, 表示するためのツールがアクセスしやすいように配置されており, 適宜圧縮がかけてある [1].

近年, PDF文書中の情報を利用しようとする研究も多く, テキストや画像などを抽出するといった試みがある. 例えばAdobe Acrobatを用いてPDFファイルから情報の抽出を行う. しかし, PDF中の数式は数学的な意味が失われた形となる. PDF文書中に存在する数式情報を数学用アプリケーションで利用できる形式で出力できれば, その論文の数学的データベースとしての価値が高まる.

そこで, 本研究ではPDFファイル中に含まれる数式情報をMathML[2]に変換できる, PDF2MathML[3]を開発した（図1）.

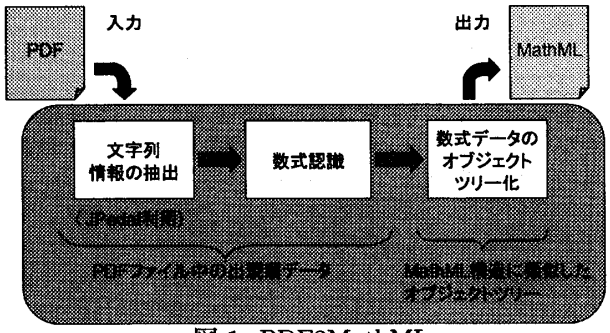

図 1: PDF2MathML

2. 研究で使用した技術

2.1 MathML

本研究は, PDFから抽出した数式情報をMathMLのPresentation Markup形式で出力する. MathMLとはXML[4]ベースの言語であり, 数学的な情報の表現に特化した言語である. 以下に $a+b=c$ の例を示す.

MathML(Presentation Markup)

```
<math xmlns=
"http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>a</mi><mo>+</mo><mi>b</mi>
    <mo>=</mo><mi>c</mi>
  </mrow>
</math>
```

†九州工業大学

2.2 JPedal

本研究では, PDFの解析のためにJPedal[5](GPL版)を用いた. JPedalは, PDFファイルをJava言語で処理することができるツールである. JPedalを利用して, PDFファイル中のテキスト表示に必要な以下の情報を抽出する.

- テキスト (本文・表示座標・フォント・文字サイズ)
- 分数などに用いられる直線の始点, 終点の座標

このようなデータを読み出すためにJPedalへの強い依存が発生している. さらに, 5つのサンプルの実行結果より, 上記以外の抽出しない情報に対する処理時間 (2.29[s]) が, 全体の処理時間 (3.35[s]) の約70%にまで及びボトルネックとなっている.

2.3 AspectJ

アスペクト指向プログラミング (Aspect Oriented Programming:以下AOP) では, 関心事によってプログラムコードの分離を行う. AOPはオブジェクト指向プログラミングのモジュール化の限界を広げることができる技術である. オブジェクト指向は呼ばれる側のモジュールを再利用するための技術であるが, アスペクト指向は呼ぶ側のモジュールを再利用するための技術と言える. [6]

AOPをJava言語で行うためのツールがAspectJである. AspectJでは, ポイントカットがアスペクトを織り込む「とき」, アドバイスが織り込む「内容」を指す.

例えば, 以下のようなAspectJの記述例の場合

アスペクトの記述

```
@Before("call(void Test.*(..))")
public void beforeSetTest{
   ...
}
```

「call(void Test.*(..))」がポイントカット部分である. これは「Testクラス内で, 任意の引数を持ち, 返り値がvoidであるメソッドを呼び出したとき」を示す. 「*」や「..」は任意のものを示すワイルドカードである.

メソッド内には, ポイントカットで指定した「とき」に織り込みたい処理の「内容」を記述する. アスペクト指向プログラミングは, 分離した状態で機能を記述するだけで, 処理したい「とき」にアスペクトを織り込む作業を自動的に行う.

3. PDF2MathML

PDF2MathMLは, 含まれている数式1つにつきMathMLファイルを1つ生成する. 抽出対象となる数式は1行立ての数式である. また, 任意のページ範囲を指定できる.

3.1 JPedal 箇所の問題解決策

アスペクト指向を利用することによって, JPedal への強い依存関係を完全に分離することが可能となる．つまり, JPedal のような外部ソフトウェアの非侵襲的な利用が可能となる．

以下に示すのは実装したアスペクトクラスの例である．

実装アスペクト例

```
@Aspect
public void PdfAspect(){

  @Pointcut("call(void *.lineTo(..))
          && args(x,  y)")
  public void lineTo(float x, float y) {}

  @After("lineTo")
  public void setLXY(float x, float y) {
    pdfFile.setLX(x);
    pdfFile.setLY(y);
  }
}
```

このクラスは，直線描画処理における直線の終了点の座標 (x, y) を取得している．pdfFile.setLX(x), pdfFile.setLY(y) の記述が JPedal 側の指定した位置に織り込まれる．

さらに，この非侵襲的利用法を応用することで，JPedal に依存することなく数式抽出に不要な処理を省く，バイパス処理が可能となる．

3.2 数式情報の抽出

JPedal を用いて抽出した情報から, 以下のような条件を用い数式とテキストを判別する．

A. 演算子を含む

B. 数式フォントが存在する

C. 数字を含む

D. 普通のテキストで使用するフォントが存在しない

上の条件で $A \cap (B \cup C) \cap D$ のときに数式とみなす．

sin，cos など,「数式フォントで表記されないが，数式を構成する数式要素である」要素は，この条件に含まれないため，あらかじめ例外として設定する．

3.3 オブジェクトモデルから MathML の生成

選別された数式情報を MathML の Presentation Markup に準拠したオブジェクトモデルとして表現する．オブジェクトモデルは数式を構築する各数式要素ごとに用意する．

このオブジェクトモデルに沿って XML ドキュメントツリーを生成し，そこから MathML を得る．XML ドキュメントツリーの生成には JDOM[7] を用いた．

4. 例題

本研究で作成したプログラムで，PDF ファイルから MathML を抽出する例を示す．まず数式を含む TEX 文書を作成し，TEX から生成された dvi ファイルを ps ファイルに変換する．次に ps ファイルから Adobe Distiller を用いて生成した PDF ファイルを本ツールに読み込む．以下，$f(x) = 4x^3 - 2x^2 + 5x$ を含む PDF ファイルを実行したときの GUI 表示を図 2 に示す．

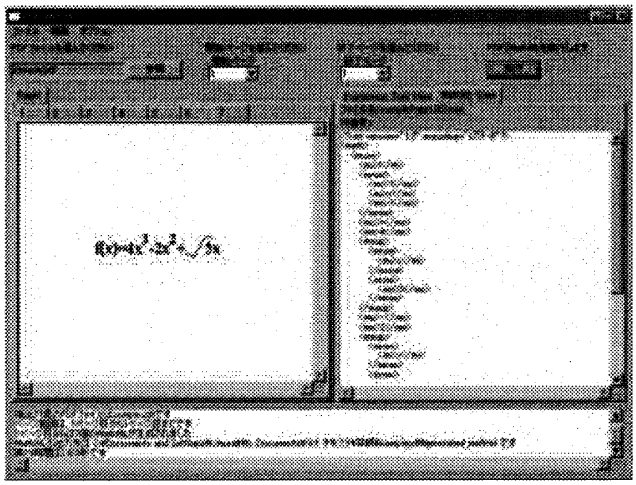

図 2: 例題の実行結果

5. まとめ

本研究では, 学会や Web 上で論文などを公開するために使われている PDF ファイル中の数式情報を, MathML 形式で抽出できるツール PDF2MathML を開発し，AspectJ の利用によって，JPedal 利用箇所の問題点を改善した．つまり，JPedal との関係を分離し，そこで不要となる処理をバイパス処理することによって，その処理分だけ高速化することが出来た．

しかし，抽出された情報から数学的オブジェクトモデルを作成するまでの処理が逐次的であり，新たな数式要素を追加すると，情報抽出後の認識処理において，判別が困難になる可能性が高い．

本ツールに新しく認識できる数式要素を追加するために，認識方法を評価関数による認識に変更し，曖昧になりがちな認識条件に対応する必要がある．さらに人間の査読を考慮した自動学習システムが導入できれば，数式要素の追加作業のコストが軽減されると考えている．

本研究の一部は科学研究費補助金 (基板 C：17560396) による助成を受けて行われた。ここに謝意を表す。

参考文献

[1] アドビシステムズ. PDF リファレンス 第 2 版. ピアソン・エデュケーション, 2001.

[2] W3C. *MathML*. http://www.w3.org/Math/.

[3] 松島一平. Pdf 文書中の数式情報抽出に関する研究. *FIT*, 2004.

[4] W3C. *XML*. http://www.w3.org/XML/.

[5] *JPedal*. http://www.jpedal.org/.

[6] 天野まさひろ, 鷲崎弘宣, 立堀道昭. AspectJ によるアスペクト指向プログラミング入門. ソフトバンク・パブリッシング, 2004.

[7] *JDOM*. http://jdom.org/.