

数式処理を用いた TeX 文書への数式の解の自動挿入システムの作成

A development of an automatic insertion system of equations solution to TeX documents using computer algebra system.

11515 黒木浩聖
指導教員 大墨礼子

1. 緒言

一般的に TeX で文書を作成する際、テキストエディタを用いて作成する。文書中に式が出現する場合その解を記述する場合は解を求め、その答えを TeX 形式で記述する必要がある。

2. 研究のアプローチ

本研究では、TeX 文章中出现する式の解を、数式処理システムを用いて求め、その解を指定したファイルの位置に自動的に挿入するソフトウェアを開発する。数式処理システムは Maxima を使用する。

3. 結果

本研究において一番重要となるのが、数式処理ソフト Maxima に TeX の文章を送信する際に、数式処理ソフトの入力方式に合わせないといけないことである。本研究では、全ての式を対応する式に変換することは無理なので、作成した式変換の対応表を表1に示す。

表 1 式変換対応表

	TeX	Maxima
分数	$\frac{a}{b}$	a/b
ルート	\sqrt{a}	sqrt(a)
微分	$f'(x) = x^3$	diff(x^3,x)
二階微分	$f''(x) = x^3$	diff(x^3,x,2)
不定積分	$\int x^3 dx$	integrate(x^3,x)
定積分	$\int_2^3 x^3 dx$	integrate(x^3,x,2,3)

3.1 分数とルートの式変換アルゴリズム

例 1 のような式のように指定した行に「frac」または「sqrt」が存在した際に、変換するアルゴリズムを示す。

例) $\frac{a}{\sqrt{b}}$

- 「{」のある前の 4 文字が「frac」または「sqrt」だった場合「{」のある index 番号をスタックにプッシュする。
- 「}」が来たら最後に取得した「{」の位置の index 番号と「}」の index 番号と「{」の前の 4 文字を配列 pear に格納する。例1を例えると以下の式ようになる。
pear[]={{(5,7,frac),(14,16,sqrt),(8,17,frac)}}
- 上記の配列 pear を利用し要素の 0 番目から index 番号を利用し切り取り挿入していく。

- frac を切り取る場合はスイッチを使い二回に一回に「/」を挿入する。

3.2 微分と二階微分の変換アルゴリズム

問題の形式は「f'(x) = x^3」を使用する。このアルゴリズムは、上記の問題を最終的に「diff(x^3,x)」という形にする。下記にアルゴリズム手順を示す。text = [f'(x) = x^3]とする。

- 数式が二階微分かの判定を行う。
- 「=」の index 番号をスタックにプッシュする。
- 配列 X1 に「diff(」という文字列を格納。
- 配列 X1 に手順 2 で取得した index 番号に +1 をした値を text から切り抜き追加挿入し、3.1 の関数を呼び出すまた、「,」を挿入。
- 配列 X1 に 2 で取得した index 番号に -2 した場所を切り抜き挿入また、「)」を挿入

3.3 定積分と不定積分の変換アルゴリズム

問題の形式は「 $\int x^3 dx$ 」を使用する。このアルゴリズムは、上記の問題を最終的に「integrate(x^3,x)」という形にする。

- 式に存在する空白のスタックを行う。
- 「 \int 」, 「x^3」, 「dx」に切り取る。
- あらかじめ用意した配列 X1 に「integrate(」という文字列を格納する。
- 手順 2 で切り取った「x^3」を X1 に挿入し、「,」も挿入する。
- 手順 2 で切り取った「dx」から「x」を切り取り X1 に挿入する。また「)」を挿入する。

3.4 実行の様子(微分の場合)

図 2 に実行画面を示す。左側が実行前、右側が実行結果である。実行結果の 27 行目に計算結果が挿入されていることが分かる。

```

23 \begin{document}
24 {\bf 1.} 次の微分を求めよ。
25 \[
26 f'(x) = 6x^3
27 \]
28
29 \]
30
31 \end{document}
32
23 \begin{document}
24 {\bf 1.} 次の微分を求めよ。
25 \[
26 f'(x) = 6x^2
27 18x^2
28
29 \]
30
31 \end{document}
32

```

図 2 起動結果図

4. 今後の発展

現在実装できてない数式変換を追加する。またテキストエディタ SublimeText2 からプログラムを起動できるようにし利便性を向上させる。

文献

- [1] 横田博史, “はじめての Maxima,” 2006 年, 工学社