

数式処理システムの高度な対話機能の開発

ー音声による数式読み上げ機能の開発ー

中村 勝則[†], 平井 綾乃[†]

武庫川女子大学 情報メディア学科[†]

1. はじめに

本研究の目的は、高度な対話機能を持つ数式処理システムの実現であり、ここでは数式を音声で読み上げる機能の実装について報告する。また、日本語コマンドによる柔軟な対話機能についても開発中であり、現状の報告と、システム開発の展望についても述べる。

2. 研究の動機と目的

近年、商用の数式処理システムは性能、機能の両面で大きく進歩しており、更には Wolfram Alpha といった、数式処理機能を持った高度な検索エンジンも利用できるようになってきた。しかし、日本語による対話機能を持った数式処理システムが一般的に利用できる状況ではなく、筆者らは日本語による操作が可能な数式処理システムの開発を目指している。本研究は、数式処理アルゴリズムの追求ではなく、数式処理システムにおけるユーザインターフェースに関するものである。最終的には、自然言語（日本語）による対話機能と、利用者の処理の脈絡を把握する機能を構築して利用者の問題解決をサポートするシステムを実現したいと考えている。

今回は、数式を日本語で読み上げる機能について報告する。

3. 構築したシステム

自然言語による対話機能を持った数式処理システムを実現する方法としては、Maxima や Mathematica といった既存のシステム上で開発する方法が最初に挙げられる。ただしそのような方法では、使用するプラットフォーム固有の処理系に依存する開発スタイル（Maxima の場合は Common Lisp, Mathematica の場合は Wolfram 言語と MathLink）が強制されるため、他の処理系（日本語形態素解析ツールや、推論系言語処理システム）との連携処理を実現する際の手間が多く発生する。そこで本研究では、開発プラッ

トフォームの中心に Python を採用して簡便な開発スタイルを取っている。Python には数式処理のためのフリーのソフトウェアライブラリも公開されているだけでなく、その他多くの有用なライブラリが利用でき、更に、他の言語処理系やツールとの連携処理が簡単に実現できるため、研究活動を円滑に進めることができる。

本研究で使用する Python 処理系は 3.6.4, GUI 構築ライブラリは Kivy organization が公開している Kivy 1.10.0, 数式処理ライブラリは SymPy Development Team (<http://www.sympy.org/>) が公開している SymPy 1.1.1 である。また、日本語の読み上げ処理には名古屋工業大学が公開している Open JTalk 1.07 を使用した。また、利用者が入力した数式を整形表示する機能も TeX と連携することによって実現した。

開発したシステムでは、数式は SymPy ライブラリ独自のデータタイプとして扱うが、これを Python のリストに変換し、更にそれを日本語テキストに変換して読み上げる。このために次に示す機能を実装した。

3-1. SymPy の式をリストに変換する機能

SymPy 独自のデータタイプを Python のリストの構造に変換する関数 `sympy2list` を実装した。これにより数式を扱いやすいリストの形式にする。（次の例参照）

例. SymPy の式を Python のリストに変換

```
a+b      → ['+', 'a', 'b']
a*b      → ['*', 'a', 'b']
f(x,y)   → ['f', 'x', 'y']
```

この他にも、冪乗や各種の関数、微分・積分の演算表現を解析する構文規則を実装しており、多くの種類の数式の変換に対応している。

3-2. リストを日本語テキストに変換する機能

Python のリストを日本語テキストに変換する

A development of intelligent function for computer algebra system

- A development of speech function for CAS -

[†] Katsunori Nakamura, Ayano Hirai

Mukogawa Women's University, Dept. of Informatics and Mediology

関数 sympy2jtx を実装した。(次の例参照)

例. 日本語テキストに変換

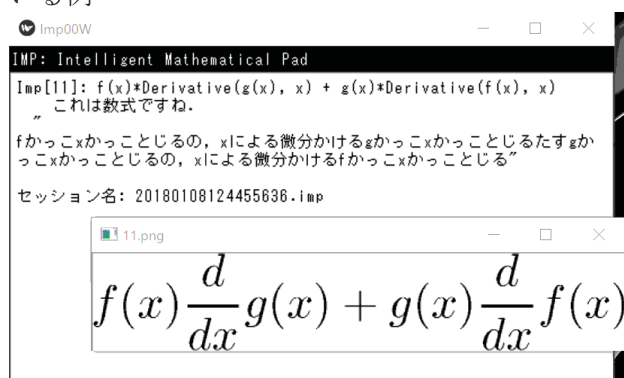
`['+', 'a', 'b']` → “a たす b”
`['*', 'a', 'b']` → “a かける b”
`['f', 'x']` → “f かつ x かつことじる”

これらの関数によって得られた日本語テキストを Open JTalk に渡して読み上げる. 入力した数式を日本語に変換し, それを TeX を介して清書して表示する様子を次に示す.

入力例: $f(x)*g(x)$ を変数 x で微分する.

```
User[11]: diff( f(x)*g(x), x )
Imp[11]: sympy
          f(x)*Derivative(g(x), x) + g(x)*Derivative(f(x), x)
User[12]:
```

出力例: 上記の入力を処理した結果を表示している例



(TeX により清書された数式が表示されている)

数式の清書表示と同時に, 数式を日本語で読み上げる

4. 展望

今回は, 数式のデータ構造を解析して日本語テキストに変換する機能について報告した. 今後も引き続き, 数式処理システムのユーザインタフェースの開発を続けてゆく予定であるが, 今後の展望について述べる. 以下に, 現在筆者らが考えている優先度に沿って述べる.

1) 日本語で与えたコマンドの処理

Python には既に MeCab (日本語形態素解析プログラム) と連携するパッケージが公開されており, この連携機能の動作を確認している. MeCab を用いると品詞の属性を伴った形で日本語の単語のリストが得られるので, これを「～は…である」という形の述語論理の式に変換することは難しくない. また「P ならば Q」という推論規則の式にすることで, 与えた日本語の文を命令文 (コマンド) として解釈することもできる. 今回の発表に引き続き

て, 日本語で処理コマンドを与える機能の開発のための準備を筆者らは始めている. 具体的には, Python と Prolog 言語処理系との連携をするための準備をしている. 想定している Prolog 言語処理系はアムステルダム大学 (蘭) が開発した SWI-Prolog であるが, これと Python を連携するためのパッケージ PySwip が Python3 に対応していないので, 当面は同等の連携機能を筆者らで開発する必要がある, 現在作業を進めている.

2) 数式のプロパティの扱い

入力した数式のタイプ (代数式, 解析的演算, 関係式など) や, 式の形状を判別することで, 次に行うべき処理 (グラフのプロットや方程式の求解など) を利用者に促すなどの機能が実現できると考えている. 既に Wolfram Alpha でもかなりの機能が実現されているが, 検索エンジンとして開発するのではなく, 日本語による問題解決の支援機能という形のシステムを目指す.

3) 利用者の意図解析

利用者とシステムのインタラクションを解析することで, 利用者の処理の意図を認識して, 利用者に対して助言を生成する機能が実現できる可能性があり, 今後取り組む予定である.

4) 日本語音声コマンドの処理

名古屋工業大学が公開している Julius や Google の音声認識 API を用いると, 日本語の音声をテキストに変換することができる. この機能を利用すると, 利用者とシステムが日本語でインタラクションすることが可能になるので今後取り組む予定である.

5. 終わりに

筆者が編纂した Python に関する技術資料をインターネットサイト

<https://qiita.com/KatsunoriNakamura/items/b465b0cf05b1b7fd4975>

で公開しており自由に入手, 閲覧できる. (フリーソフト)

学部生向けの Prolog のテキスト (フリーソフト) も公開している.

