

日本語プログラミング言語における、より日本語らしいプログラム記述の実現

伊東一樹[†] 寛 捷彦[‡]早稲田大学大学院基幹理工学研究科情報理工学専攻[†] 早稲田大学 理工学術院[‡]

1. 序論

1.1. 背景

近年、多くの日本語プログラミング言語が開発・研究されている。日本語プログラミング言語とは、プログラムを自然な日本語で書く事を目指して設計されたプログラミング言語である。代表的な日本語プログラミング言語として、プロデル[1]、なでしこ[2]などが存在する。本研究ではプロデルを題材として研究を行っていく。

1.2. 日本語プログラミング言語「プロデル」

プロデルではメソッドの定義・呼び出しを行うことができる。そのプログラムの例を図 1.1 及び図 1.2 に示す。

【配列】 から、**【内容】** を、探す手順
処理内容
終わり

図 1.1 プロデルにおけるメソッド定義

果物リストは、{「リンゴ」、「なし」、「スイカ」}
果物リストから「リンゴ」を探す

図 1.2 図 1.1 のメソッドを呼び出すプログラムの例

図 1.1 では“探す”というメソッドを定義している。その際、“配列”と“内容”という引数を取るように、“【”と“】”で指定している。引数を指定する際には、引数それぞれに補語を付与する必要がある。ここでは、“配列”に“から”、“内容”に“を”という補語をそれぞれ指定している。このメソッドを呼び出すには、図 1.2 のようにに引数部分に変数または値を指定し、定義されている補語とメソッド名を用いて記述する。ここでは、“配列”引数に

“果物リスト”という変数を、“内容”引数に“リンゴ”という文字列を指定し呼び出しを行っている。

1.3. 問題点・目的

節 1.2. で示したようにプロデルにおいてメソッドの呼び出しを行うプログラムは、決められたキーワードを用いて記述する必要がある。そのため、日本語文章としての自由度が狭められてしまっている。結果的に日本語プログラミング言語としての利点を大きく損なってしまっているのである。

本研究では、上記の問題点を解決するため、より自然で幅広く記述することのできる日本語プログラムの実現を目的とする。

具体的には、入力された日本語文章(プロデルプログラムとしてコンパイルできないもの)を解析しその文章の意味に最も近いと推測されるプロデルのプログラムを出力することを行う。

現実的なシステム開発の場面では、本研究によって次に挙げるようなことが実現される。

- コンパイルエラーとなるプログラムを、即座に正しいプログラムへと自動修正を行う。
- 実際のプログラムよりも短い文章から正しいプログラムを生成できる。

2. 提案手法

本研究では幅広い記述法を実現するため、生成するメソッドの候補を用意しておく。その際、全ての候補メソッドに対して情報を持たせておく。情報の内容は次の通りである。

- メソッドの処理を表すキーワード
- 引数の情報
 - キーワード
 - 取りうる型
 - 取りうる補語のパターン
- 引数の取りうる型

キーワードと補語にはスコアを割り当てておく。これらを利用し、全ての候補メソッドの情報と、入力された日本語文章の間で類似度を求める。最も類似度の高いプログラムを出力する。

これによって幅広い記述法に対応しつつ、的確なプログラムを選び出すことができると思わ

Realization of the Program notation more naturally Japanese On Japanese Programming Language.

[†] Kazuki Ito, Department of Computer Science and Engineering, Graduate School of Fundamental Science and Engineering, Waseda University.

[‡] Katsuhiko KAKEHI, Faculty of Science and Engineering, Waseda University.

れる。

3. 提案システム

3.1. 概要

本システムは、プロデルデザイナー(プロデル専用のIDE)のアドインとして実装を行う。システムの実行は次の手順で行う。

1. プロデルデザイナーに日本語文章を入力する
2. 入力した日本語文章上にカーソルを置く
3. Ctrl+Enter を入力する。

システムを実行すると、入力された日本語文章はプロデルで動作する正しいプログラムに置換される。

3.2. 実現方法

入力された日本語文章と候補プログラム群から出漁プログラムを生成する際の手順を示す。

1. 入力された日本語文章を単語の組の形に構文解析しクエリとする。構文解析はプロデルの構文解析器を主に使用し、それに対応できない場合 MeCab[3]を使用する。
2. 全ての候補プログラムのキーワード情報とクエリとの間の類似度を求める。最も類似度の高い候補プログラムを出力プログラムのベースとする。
3. 2. で選択したプログラムのベースへ引数を割り当てる。割り当てる引数は、プログラムの引数情報を利用して決定する。

4. 評価

実際のシステム開発において、本システムがどのように利用できるのかで評価する。

4.1. 評価方法

次の手順で評価実験を行う。

1. プロデルのリファレンスを一切見ずにシステムの開発を行う。
2. 記述したプログラムがコンパイルエラーとなる時、本システムを実行する。
3. 出力されたプログラムのうち正常に動作するプログラムの割合を計測する。

4.2. 題材とするプログラム

今回のシステムでは、似た単語を含む変数名やクラス名などが多く存在するほど精度が悪くなると思われる。よって、そのようなプログラムの例としてシューティングゲームを題材とする。その逆で簡素なプログラムであれば精度が上昇すると思われるので、題材として四則演算のみを行える電卓を採用する。また、本システムはプログラマ自身が定義を行っていないプログラムの記述時に力を発揮すると思われる。こ

こではDB操作を簡潔に行う事のできるライブラリを用いたシステムとしてToDoリストを採用する。

4.3. 評価結果

出力したプログラムの内、およそ6割は想定通りの動作を行うものとなった。残りの4割は異なったプログラムになってしまった。しかしそのほとんどのプログラムは、引数として割り当てられたものが異なっているというものであり、全く異なるプログラムが出力されたのは、全体から見て1割程度であった。

5. 結論

日本語プログラミング言語においてこれまで問題となっていた記述の自由度の狭さという問題点の解決法として、補語とメソッド名と変数の型を複数指定しておくという手法を提案した。その結果として日本語プログラムの記述の自由度を広めることに成功した。

参考文献

- [1] ゆうと: “日本語プログラミング言語「プロデル」”, <http://rdr.utopiat.net/>, 2013, 2013年7月24日閲覧。
- [2] クジラ飛行機: “日本語プログラミング言語「なでしこ」公式ページ”, <http://nadesi.com/>, 2013, 2013年7月24日閲覧。
- [3] 京都大学情報学研究所: “MeCab: Yet Another Part-of-Speech and Morphological Analyzer”, <http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>