

関数型日本語プログラミング言語“クロガネ”の開発

太田 大地[†] 笥 捷彦[‡]早稲田大学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

プログラマの世界では、ドキュメントを日本語で書くが、プログラムを日本語で書くことはほとんど無い。しかし日本人は、英語より日本語の方が慣れ親しんでいる。このことに疑問を持ち、プログラム全体を日本語で書ける事を目的とした日本語プログラミング言語が古くから考案されてきた。古くは和漢[1]が1983年に公開され、現在もプロデル[2]や、なでしこ[3]などの開発が続けられている。これらの言語では、プログラムを日本語の文章として表現する。それによりプログラム全体の可読性の向上、初心者に対する教育の優位さなどが長所として挙げられる。

2. 提案

本研究では、新しく関数型日本語プログラミング言語クロガネ[1]を開発した。関数型言語にすることで日本語らしくかつ簡潔に書ける表現を提案する。日本語プログラミング言語は、プログラムを文章らしく書くことで、可読性を上げている。そのためプログラムが冗長になる場合もある。関数型言語で記述されたプログラムは処理が抽象化されており、手続き型のものと比べ簡潔に表現できることを期待する。

2.1. 関数型言語としての特徴

関数型言語という言葉はあるが、どこからが関数型言語かという境界はあいまいである。Haskell[4]のような完全な参照透過性をもつ言語のみを指すという意見もあれば、JavaScriptのような高階関数が利用できさえすればよいという意見まで様々ある。クロガネでは、高階関数やクロージャといった機能に加え、対の容易な作成・展開や再代入を避ける構文を実装する。

Kurogane – Implement of a Japanese functional programming language

[†]Daichi OTA and [‡]Katsuhiko KAKEHI

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

[‡]Faculty of Science and Engineering, Waseda University

2.2. 日本語プログラミング言語としての特徴

関数型日本語プログラミング言語を作成する上で、必要だと考える構文を順に示す。

2.2.1. 重文が利用できる

プログラミングにおいて、ある式を計算し、さらにその結果を別の式で計算する、といった事はよく起こる。これに重文を用いることで簡潔に表現する。重文とは“雨が降り、地が固まる。”のようなそれぞれが単独の文として成り立つ文を複数重ねた文である。重文を用いて複数回関数を適用する場合、プログラムを“甲し、乙する。”と書いた場合の実行方法は、まず“甲する。”を実行した後、戻り値を“乙する。”の引数として実行するものとする。

2.2.2. 助詞“と”により対を作成する

関数型言語では対や連続した対であるリストはよく用いられるデータ構造である。この対は頻繁に用いられるため構文的に簡単に作れることが望ましい。誰もが“cons(a, cons(b, c))”のような長い関数呼び出しより“a : b : c”のような短い構文を望むだろう。日本語で物を並列に表現しようとした場合、助詞“と”を用いる。例えばaとbとcを連結するプログラムは“aとbとc”と書くことができる。そのため助詞“と”を、対を作るための特別な助詞として扱い、通常の助詞としては扱わないものとする。

“と”が標準の助詞として利用できない場合、“AとBを加算する。”のような関数が宣言できなくなる。そこで関数宣言に“と”が用いられている場合は、自動的に対を展開することで、まるで助詞“と”が通常での助詞あるかのように記述できる。さらに“AとBとCにDとEとFをGする。”の様な“と”が複数現れるような関数も宣言可能になる。

3. 実装

本研究ではクロガネを実装し、オープンソースで公開した。

クロガネはC#で書かれ、“.NET Framework 4 Client Profile”上で動作する。これは、既存の日本語プログラミング言語と連携することを

考慮したためであり、これによりプロデルと部分的に連携することが可能となった。

クロガネのソースコードの解析は字句解析、構文解析、意味解析の三つに分割して実装した。それぞれ Lexer クラス, Parser クラス, ExpressionGenerator クラスが担当している。これらのクラスは互いに参照していない。二つの観点からこのように設計している。一つは保守性のため、もう一つは可読性のためである。今後この言語を改良していく場合、互いに独立していれば、そのクラスのみを変更するだけで済む。また、追実験とまではいかななくても日本語プログラミング言語を実装したいと考える人が現れた場合、ソースコードを理解しやすくする。

4. 評価

クロガネにおいていくつかプログラムを記述し、各構文表現に対して評価を行った。

4.1. 重文

重文は有用であった。例としてソースコードをコンパイルするプログラムの例を挙げる。リスト 4.1 が従来のプログラムで記述したもので、リスト 4.2 が重文を用いて記述した例である。重文を用いることで、プログラムを短く書くことができる。さらにプログラム中に現れる括弧を減らすことができ、日本語らしさの向上につながる。

プログラムは、（（ソースコードを字句解析したもの）を構文解析したもの）を意味解析したもの。

リスト 4.1 重文を用いないプログラムの例

ソースコードを字句解析し、構文解析し、意味解析し、プログラムとする。

リスト 4.2 重文を用いないプログラムの例

重文を用いる場合、動詞は連用形として活用する。クロガネでは関数名をすべてサ変名詞とした。そのため重文の表現は常に“～し”という表現になる。もし関数名に自由な動詞を利用できる構文規則とする場合、従来での終止形に加え、連用形で表現される動詞を構文解析する必要が出てくる。

4.2. 助詞“と”

プログラムを記述する上で有用であった。プログラム中で二つの値を返したい場合やリストを簡単に作成するために、助詞“と”は欠かせないものだった。リスト 4.3 は、指定された日付から曜日を求めるプログラムの例である。このように適切な助詞が三つ見つからないような

場合に、簡単なデータ構造として、リストを利用することができる。

※関数宣言。

以下の定義で年と月と日から曜日計算する。

※関数の実装。

以上。

※関数呼び出し。

2011 と 1 と 14 から曜日計算する。

リスト 4.3 対を展開する関数呼び出しの例

クロガネでは対の前者への参照を“対の頭”、後者への参照を“対の体”と表現し、この“の”による演算の結合度は“と”に比べ高く設定した。対の一つ目の要素と二つ目を入れ替え、結果に代入するプログラムをリスト 4.4 に示す。

対の体と対の頭を結果とする。

リスト 4.4 対の各要素を参照するプログラム

5. 考察

本研究で新たな表現を提案することで、日本語プログラミング言語をさらに簡潔に表現することを可能にした。また、クロガネを実際に試してみることで、より少なくとも単純なプログラムであれば、ある程度日本語らしい表現の中で関数型言語を用いてプログラムを作成できることを確認した。しかし今回作成した言語を用いて大規模なプログラムを書いたわけではない。今後、日本語関数型言語を用いて大規模なプログラムが作成できるかを確認する必要がある。

また最近では、ソフトウェア開発をしやすいオブジェクト指向プログラミング言語に関数型言語の機能を部分的に取り入れるような言語が増えている。日本語プログラミング言語既存の言語に関数型言語の機能を取り入れた場合の表現も考えてみる必要があるかもしれない。

6. 参考文献

- [1] 日本語プログラミング言語「クロガネ」
<http://kurogane.codeplex.com/>
- [1] 鈴木孝則：日本語プログラミング言語『和漢』、情報処理学会マイクロコンピュータ研究会資料(29-2)、1-10、1983-11-28.
- [2] 日本語プログラミング言語”プロデル”
<http://rdr.utopiat.net/>
- [3] 日本語プログラミング言語”なでしこ”
<http://nadesi.com/>
- [4] Haskell - HaskellWiki
<http://www.haskell.org/>