

JavaScript を対象にした関数の視覚化による理解支援

情報・通信工学科 学籍番号：1311196 寺田研究室 村松啓寛

1 背景

ソフトウェアの視覚化はプログラミング教育、理解、実際にコードを書く上で有効な技術である。最近では UML ^{*1} 等が挙げられる。ソフトウェアの視覚化はコード自体やデータ構造の視覚化、静的または動的な視覚化と分類できる。

JavaScript はブラウザで採用されている言語であり、最近ではサーバーや様々なアプリケーションでも採用されており、広く利用されている言語である。JavaScript の関数は関数型言語と同様に引数や戻り値として扱うことができる第一級オブジェクトであり、重要な役割を担っている。しかし、その柔軟性によって複数の関数が複雑に関係してくると、コードをみて、その関数がどのような役割で、どのように動作するのか理解しづらくなる問題が存在している。

2 目的

本研究は JavaScript プログラムの関数とそれらの関係を図形的に視覚化して理解支援を行うことを目的とする。

3 提案システム

3.1 基本方針

JavaScript において関数は重要な要素の一つである。よって、関数と関数に関する要素に注目して視覚化を行うことを基本方針としている。関係しない要素は視覚化された図を見やすくするために基本的に表現を行わない。JavaScript の基本的文法を理解している、初心者ではないユーザーの利用を想定している。

3.2 関数に関する要素

関数に関する要素とは以下のようなことである。

- 関数宣言
- 関数が代入されている変数
- 関数呼び出し
- 関数呼び出しの戻り値が代入されている変数
- 関数を指している識別子を持つオブジェクト
- 関数を指している識別子を持つオブジェクトが代入されている変数
- 内側の関数が外側の関数の変数を参照している場合の参照されている変数

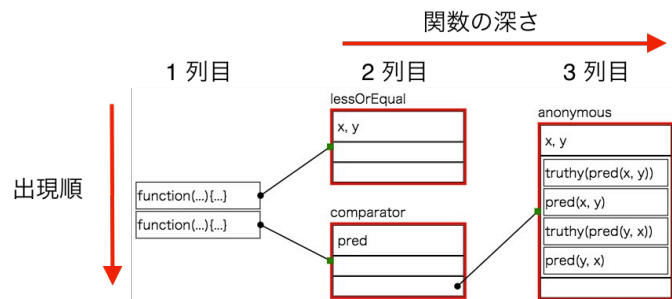


図1 視覚化の例

3.3 関数の表現

関数において、引数や戻り値は重要な要素なので明示する。引数に関しては、すべての要素を表現する。戻り値に関しては、関数や関数呼び出し、関数を値に持つプロパティを持つオブジェクト、内側の関数が外側の関数の変数を参照している場合の参照されている変数を表現する。また、無名関数も JavaScript では重要な要素であるので、関数名が存在する関数と同様に扱う。

3.4 基本ルール

図1は視覚化した例である。関数は三つの区域からなる長方形で、上段が引数、中段が本体の処理、下段が戻り値を示している。視覚化された図の横方向は左から2列目以降が関数の深さを表している。左から右にいくほど、深い関数であることを表している。各列は以下のようにになっている。

- 1列目
グローバル領域を表していて、関数宣言の要素、関数や関数呼び出し及びその戻り値が代入されている変数が表現されている。
- 2列目
グローバル領域内で定義された関数または無名関数や無名関数呼び出しの関数が表現されている。
- 3列目以降
前の列の関数の中で定義された関数または無名関数や無名関数呼び出しの関数が表現されている。

縦方向は要素がコード上で出現した順番を表している。下にいくほど、後から出現した要素であることを表している。

4 評価

JavaScript の参考書 [4][5] に掲載されている複数の関数が関わる関数宣言や無名関数呼び出し、再帰関数、内側の関数が外側の関数の変数を参照する場合などのサンプルコー

*1 <http://www.uml.org/>

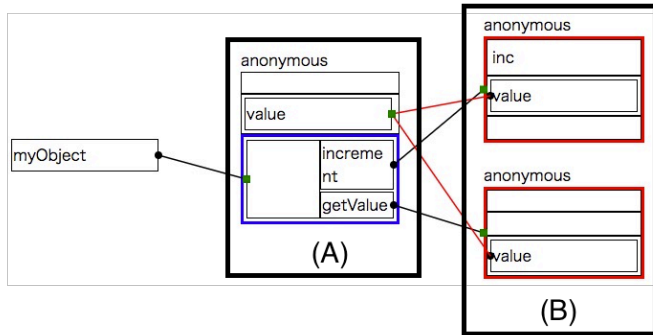


図 2 無名関数呼び出し及び内側の関数が外側の関数の変数を参照する場合のサンプルコードの視覚化

ドを例にし、視覚化することによってどのようにわかりやすくなっているかや逆にどのような部分がわかりにくくなっているかをまとめた。以下の点の評価を行った。

- 関数の関係を表すことができているか
- 関数の関係がわかりやすくなっているか
- 関数の関係以外でどのような部分が表せていたり、わかりやすくなっているか

無名関数呼び出し及び内側の関数が外側の関数の変数を参照する場合のサンプルコードの視覚化に対して評価を行った。図 2 はそのサンプルコードを視覚化した図である。(A) は無名関数呼び出しを表現している。(B) の関数は (A) の中で定義された関数であり、その中の変数 `value` から (A) の変数 `value` に引かれていた矢印が内側の関数が外側の関数の変数を参照する表現である。

幾つかのサンプルコードを視覚化した結果、以下の点が表せており、わかりやすくなっていると考えられる。

- 関数定義なのか無名関数呼び出しなのか
- 関わっている関数の数
- 関数の返り値に関数が含まれているか
- 内側の関数が外側の関数の変数参照
- 引数の有無、その数

また、逆に表せていなかったり、わかりにくくなったのは以下の点である。

- 関数の関係のみの視覚化のため関数自体がどのような動作をするのか表せない
- 関数が何を行っているかわかりにくい
- 矢印の始点や終点が近くに描画されるとどちらを指しているかわかりにくい

5 関連研究

5.1 Visual Haskell: A First Attempt[1]

Reekie の研究では関数型言語の Haskell を基にした Visual Haskell というビジュアルプログラミング言語を提案した。彼の研究は Visual Haskell を使用することでプログ

ラムを静的に視覚化することを可能にしている。

5.2 Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education[2]

Philip の研究では Web ベースで Python のプログラムの実行を視覚化し、学習支援を行うシステムを提案した。彼のシステムはユーザーが入力した Python のプログラムを動的解析し、視覚化している。実行を幾つかに分け、コードのどの状態が表現されているかユーザーに提示する。実行の様子は自動的にアニメーションのように再生されるのではなく、ユーザーが自ら進めたり、戻したりできるようになっている。また彼は、JavaScript でも同様なシステム JavaScript Tutor[3] を作成している。

6 結論

JavaScript を対象とした関数の関係に対し、それぞれの要素に対し、視覚化表現の定義を行い、視覚化を自動的に行うシステムを作成した。様々なサンプルコードを視覚化し、関数の関係を図に表すことができ、それらがわかりやすく表現できていると考えられる。そのため、本システムを使用することにより、JavaScript を対象とした関数の関係についての理解支援ができると考えられる。

7 今後の課題

本研究で作成したシステムには以下のような課題が残っていると考えられる。

- 被験者実験の実施
- 実装できていない関数に関する表現
 - 関数を関数の引数にした場合の関数間の矢印
 - 内側の関数から外側の関数の変数を参照した場合の変数間の一部の矢印
- 視覚化表現を行う線引き
関数呼び出しや返り値の表現の表現の仕方
- スケールの問題
コードの量が増えると図が縦長になってしまい見にくくなる問題

参考文献

- [1] H. John Reekie: Visual Haskell: A First Attempt, Technical Report 94.5, University of Technology, Sydney, 1994.
- [2] Philip J Guo. Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education. In Proceeding of the 44th ACM technical symposium on Computer science education, pp.579-584. ACM, 2013.
- [3] JavaScript Tutor - Visualize JavaScript code execution to learn JavaScript online, <http://pythontutor.com/javascript.html>.
- [4] Douglas Crockford. JavaScript: The Good Parts 「良いパーツ」によるベストプラティクス. オライリー・ジャパン, 2008.
- [5] Michael Fogus. JavaScript で学ぶ関数型プログラミング. オライリー・ジャパン, 2014