

# 学習者が設定したBNFに対応するコンパイラの コンパイル処理過程を可視化するシステムの提案

## A Proposal for a System to Visualize the Compilation Process of a Compiler Corresponding to a BNF Set by a Learner

平西 宏彰<sup>\*1</sup>, 香川 考司<sup>\*2</sup>  
Hiroaki HIRANISHI<sup>\*1</sup>, Koji KAGAWA<sup>\*2</sup>

<sup>\*1</sup>香川大学大学院工学研究科

<sup>\*1</sup>Graduate School of Engineering, Kagawa University

<sup>\*2</sup>香川大学創造工学部

<sup>\*2</sup>Faculty of Engineering and Design, Kagawa University

Email: s21g471@kagawa-u.ac.jp

**あらまし:** 本稿では、プログラミング言語およびコンパイラに対する理解の促進を目的としたシステムを提案する。本システムは、コンパイル処理を字句解析、構文解析、意味解析、最適化、目的コード生成の過程で区分し、それぞれの過程で行われる処理を Web ブラウザ上で可視化する。学習者の計算機科学やプログラミングに対する理解度に応じた学習を提供するため、学習者が設定したBNFに対応するプログラミング言語をコンパイル対象とする。

**キーワード:** コンパイラ, 可視化, プログラミング教育

### 1. はじめに

言語処理系はプログラムが計算機上で動作するために重要な役割を果たしている。また、言語処理系で発案されたアルゴリズムは、計算機科学の中でも重要なものがある。このように言語処理系は、計算機科学を学ぶ上で非常に役立つものである。そこで、コンパイラの学習支援のために、Web ベースで動作する可視化システムが開発されてきた。Web ベースのコンパイラ可視化システムは、視覚的に学習可能、特別なソフトウェアのインストールが不要などの利点があり、学習者が効率的にコンパイラの理解をする上で重要なものになっている。しかし、既存のシステムではコンパイラの構文解析までの過程に限定しているものが多い。コード生成まで行なっているものでも、機械語もしくはアセンブリ言語に関する知識が浅いものにとっては理解が困難である。そのため、学習者の理解度に応じた教材を提供できないなどの欠点がある。しかし、本来は機械語やアセンブリ言語の知識が浅い者に対して理解を助ける環境を提供すべきであると考えられる。

本項では、コンパイラ内部処理を可視化するシステムである Visual Compiler<sup>(1)</sup> の改良案を提案する。特にアセンブリ言語に関して理解が乏しい学習者が、コンパイラのコード生成における動作の理解を促進することを目標としている。

### 2. Visual Compiler

Visual Compiler は Web ベースで動作するコンパイラ可視化システムである。このシステムの主な仕様と特徴は以下の通りとなる。

- ・入力されたソースコードに対して、字句解析、構文解析、コード生成を動的に行い、その過程を可視

化する。

- ・ステップ実行が可能であり、任意の箇所まで停止して動作の確認が可能である。

- ・入力されたソースコードにバグがあればエラー箇所を示す。

- ・CPU の内部処理を可視化するシステムである Visual CPU Simulator<sup>(2)</sup> で実行可能なアセンブリコードを生成する。生成されたコードは、Visual CPU Simulator によって動的に実行され視覚的に確認可能である。

また、実際に Visual compiler を利用した学習者からは、良い点として、

- ・視覚的に確認できるため、コンパイラ全体の概要を掴みやすい。

- ・コンパイラがプログラミングの過程でどのような役割を果たしているのか理解しやすい。

という意見を得ることができた。一方、悪い点として、

- ・アセンブリ言語について理解が乏しい場合、コンパイル処理の理解が困難である。

- ・ユーザインターフェースに関する内容で画面が見にくい。

という評価を得ている。つまり、コンパイラ全体の概要を掴むことはできるが、詳細部分、特にコード生成の箇所で理解が困難という結果になっている。

また、Visual Compiler のユーザインターフェースを図1に示す。

### 3. システムの問題点

#### 3.1 コード生成の理解

Visual Compiler のコード生成の理解の促進の効果について、学習者は理解が困難であるという結果

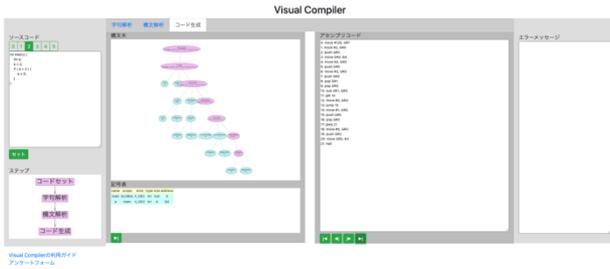


図1 Visual Compiler のユーザインターフェース

を得ていた。実際の評価はアンケートの回答者によって評価が分かれており、これは自由記述による回答から、Visual Compiler の学習者が機械語ないしは、アセンブリ言語のような低級言語への元々の理解度が影響していると考えられる。これらの低級言語について理解が十分な者は、コード生成の過程でどのような変換が行われているか理解することを容易であったのに対し、低級言語について理解が乏しい者は、コード生成についての理解を困難であった。しかし、Visual Compiler はプログラミング初学者等の低級言語について深く理解していない学習者を対象として開発されている。そのため、低級言語について最低限の理解があれば、どのような変換がされているか理解が促進される仕組みが必要であると考ええる。

### 3.2 ユーザインターフェース

ユーザインターフェースは、画面を見にくい点が問題点となっている。具体的には、コード入力画面、構文木等の文字が全体的に小さく、見えにくい点である。文字サイズの大小の変更は、CSS を記述すれば良いだけである点を踏まえて、簡単に実装できると考えられる。しかし、それぞれのオブジェクトの文字サイズを単に大きくするだけでは、コード量が多くなった場合、構文木が大きくなった場合等の対応ができなくなる。そのため、Visual Compiler の学習者が文字サイズを任意に変更可能にすることや、オブジェクトのサイズが変更されるタイミングで文字サイズが変更する機構を用意するなどの対応が必要である。

## 4. 問題点に対する改良案

### 4.1 コード生成の理解

コード生成に関する理解をより促進するための改良案を提案する。低級言語に関する知識の差が、Visual Compiler のコード生成における理解促進の効果を大きく左右していたことから、低級言語の中でも一部の命令についてのみ理解しておけば、生成されたコードの意味がわかるようにすることで解決に近づくと考える。これを実現するために、システムのコンパイラがコンパイル対象とするプログラミング言語の構文を限定することを提案する。従来のシステムでは、システムで設定していた構文にしか対応していなかったが、構文を任意のものに対応さ

せることで、構文の複雑さを可変にすることができる。これにより、コンパイル過程の難易度も同様に变化させることができると考える。構文の変更は、学習者が BNF によって構文を記述して設定すればよく、別の選択肢として指導者があらかじめ設定した構文を用いることができるようにしても良い。任意の構文でコンパイラを生成する必要があるため、解析方法も変更する必要があるが、元々は、下向き構文解析で実装していたが、LR 法を用いることにより解決できる。

### 4.2 ユーザインターフェース

ソースコード入力部分をテキストエリアから埋め込みエディタに変更することで解決可能であると考えられる。これにはブラウザにエディタを埋め込むことができるライブラリである ACE<sup>(3)</sup>を利用することで解決できる。ACE を用いることにより、課題である文字サイズを動的に変更することが可能になる。さらに、副次的な効果としてシンタックスハイライトを付加することができ、学習者がソースコードを記述する環境を整えることにもつながる。

次に、構文木は構文木全体を見たい場合と細かい箇所を見たい場合がある。構文木全体を見たい場合は、構文木の末端に近い箇所を上位の葉に格納し、細かい箇所を見たい場合は、格納した葉を表示することを考える。これにより、普段は構文木全体を視認することができ、必要になった場合に、該当する葉から下位の葉を広げていけば良くなる。

また、LR 法を用いることによる新たな問題点も発生する。構文解析表が作成されるが、この表は大きくなりすぎることがあり、工夫して表示する必要がある。必要箇所だけをハイライト表示し、その他の不必要な箇所を小さく表示することで解決できると考える。

## 5. おわりに

本項では、Visual Compiler を改善することで、低級言語への理解が浅い学習者でもコンパイラについて理解を促進できるシステムを開発する指針を示した。今後は、提案した改良案に基づいて仕様を確定し実装することが課題となる。

### 参考文献

- (1) Shinya Hara, Yoshiro Imai, "Register-Transfer-level CPU Simulator for Computer Architecture Education and Its Quantitative Evaluation", IEEJ Transaction on Electronics, Information and Systems Vol. 138, No.9, pp.1123-1130, (2018)
- (2) Hiroaki Hiranishi, Primož Podržaj, Yoshiro Imai, Tetsuo Hattori, "Development of Web-based Visual Compiler for Computer Literacy" Proceedings of the Sixth International Conference on Electronics and Software Science ICES2020, pp.23-30, (2020)
- (3) Ace, <https://ace.c9.io/>