

調査項目の拡張しやすさを考慮した ソースコード解析システムの構築

19T305 小方 亮人（香川研究室）

プログラミングを学び始めたばかりの初学者はコンパイル時のエラー内容を理解することが困難である。本研究では入力されたソースコードに対して解析を行い、間違っている箇所を指摘するシステムを構築した。解析には Haskell を用い、調査項目の拡張性が期待できる。

1 はじめに

学習者が C 言語を学び始めたばかりの時、コンパイルした際に表示されるエラーや警告が何を意味しているのかが分からず、エラーの修正に多くの時間をかけてしまうことが多々ある。初学者によくある間違いに対してより明確なエラーメッセージを出力することで、エラーの修正が的確に行えるようになり学習効率が向上すると考えられる。また、プログラミング学習者がソースコードを記述する際に犯す間違いには様々な種類がある。エラーを指摘するシステムを構築した時点でその全ての間違いを網羅することは不可能であり、システムを運用していくにつれてエラーの調査項目を増やす必要がある。そのため、エラー指摘システムには新しく調査したい調査項目ができた時に容易にシステムに組み込むことができる拡張性が求められる。

2 先行研究

島川の研究 [1] は C 言語を初めて学ぶ学習者によくある間違いに対して理解の容易なエラーメッセージを表示しエラーの修正を補助する機能を持つ。これにより簡易な質問数が減少することで学習効率の向上や指導者の負担軽減を図っている。しかしこのシステムで利用している C-Helper [2] が実装に用いている Java では構文解析結果である構文木を扱う際に Visitor パターンというデザインパターンが用いられている。新しく調査項目を増やそうとする場合、新しいメソッドの追加と各クラスにそのメソッドに応じた実装を追加する

必要があるため、調査できる項目の拡張性に優れていない。この問題点に対して Haskell を用いて解決を試みた研究が木村の研究 [3] である。木村の研究はソースコードのエラー箇所をハイライトすることで教育者の添削を支援するものである。しかし学習者に対して即座にミスのフィードバックを示すことはできず、教育者に対してもエラー指摘の完全な自動化は図られていない。そこで本研究では、実装に Haskell を用いることで容易に調査項目を増やすことができ、学習者に即座にフィードバックを行うことができるシステムを構築することを目的とする。

3 システムの実装

本研究では Haskell とそのライブラリである language-c-quote を用いてシステムの実装を行う。Haskell には参照透過性や静的型付けといった関数型言語に多く採用されている機能に加え、パターンマッチや型推論、モナドなどの特徴的な機能がある。これらの機能を組み合わせることで構文木のようなデータ型を扱う際に命令型言語に比べて Haskell はより簡潔かつ容易に関数を実装できることが期待される。language-c-quote は C 言語の構文解析を行う Haskell のライブラリであり、解析結果が扱いやすいため、新しく調査項目を増やす際に容易にプログラムに追加することができると考えた。また、Web ベースで実装を行うことでシステムのインストールを不要とし、利用者側と開発側が双方ともにシステム導入に関する負担が軽減されることを目指した。

本研究のシステムは、ブラウザ上のテキストエリアの入力フォームに入力されたソースコードを文字列として受け取り language-c-quote によって構文解析を行う。その結果から初学者が間違いを起こしやすい if 文などの情報を抜き出し、間違っていた記述をしていないか調査を行う。本システムで調査できる項目は、インデントのミス、printf や scanf の受け取る変数の数が間違っているミス、if 文の条件式が関係演算子による式ではなく代入式になっているミス、関数名が他の関数と重複しているミス、返却値が int 型の関数に return が記述されていないミスがある。間違っている箇所の行番号とその内容を HTML ファイルに書き込み、解析結果を表示するページに出力する。この出力表示では入力されたソースコードとソースコードの間違っている行番号の色を変更して表示しているため、視覚的にエラー箇所を理解することができる。以下にソースコードの入力画面 (図 1) と解析結果の表示画面 (図 2) を示す。

C言語ソースコード解析システムです。

解析したいソースコードを入力してCHECKボタンを押してください。

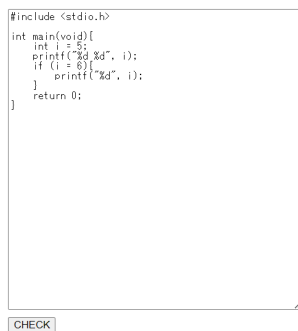


図 1: ソースコード入力画面

4 まとめ

本研究では、プログラミング初学者向けのソースコード解析システムの構築を行った。インデントのミス等の初学者が犯しがちなミスに対して調査を行う機能を Web ベースシステムで実装することに成功した。また、ソースコード構文解析結果から Haskell によって調査したい構文の箇所を

解析結果

```
1  #include <stdio.h> ERROR
2
3  int main(void){
4      int i = 5;
5      printf("%d %d", i);
6      if (i = 6){
7          printf("%d", i);
8      }
9      return 0;
10 }
```

5行目 printfで指定されている引数の数が違います
6行目 if文の条件式は = ではなく == を使用してください

図 2: 解析結果表示画面

抜き出して調査を行っているため、Haskell の知識さえあれば調査項目を容易に拡張することが可能である。

しかし、現状で調査することが可能な項目の数は少なく早急な調査項目の追加が求められる。また Haskell の知識が必要なことから、調査項目の拡張についての評価実験を行うことが困難であったため拡張しやすさに関する評価は筆者の主観的なものでしかない。そのため実際に Haskell を学ぶ講義で追加する調査項目のコードを記述する課題を出したり、知識がなくても調査のイメージをつかめるように分かりやすさを重視したビジュアルプログラミングを利用したりといった方法で解決を図る必要がある。

参考文献

- [1] 島川 大輝, 香川 考司, “C-Helper を用いた Web ベースの C 言語開発環境の構築”, 教育システム情報学会第 40 回全国大会 (JSiSE2015) 講演論文集, A3-1, 2015.
- [2] サイボウズ・ラボユース Ucida Kota, “C-Helper GitHub”, <https://github.com/uchan-nos/c-helper> (閲覧日:2023 年 2 月)
- [3] 木村 光星, 香川 考司, “構文解析を用いた C 言語指導コメント支援システムの構築”, 教育システム情報学会 (JSiSE) 2018 年度第 4 回研究会, 2018