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

Implementation of a Source Code Analysis System Considering the Ease of Expansion of Survey Items.

小方 亮人[*1], 香川 考司[*2]
Ryoto OGATA[*1], Koji KAGAWA[*2]
[*1] 香川大学大学院創発科学研究科
[*1]Graduate School of Science for Creative Emergence, Kagawa University
[*2] 香川大学創造工学部
[*2]Faculty of Engineering and Design, Kagawa University
Email: s23g353@kagawa-u.ac.jp

あらまし：本研究ではブラウザ上の入力フォームに入力された C 言語のソースコードに対して解析を行い、構文解析を通過したプログラムに対して問題点を指摘するシステムを構築した。主に初学者が間違いやすい項目に対して調査を行う。解析には Haskell を用い、調査項目の拡張性が期待できる。
キーワード：Haskell, 構文解析, プログラミング学習支援

1. はじめに

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問題点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点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。本研究では Haskell を用いることで、構文解析を通過した C 言語のソースコードに対して初心者が犯しやすい間違いやコードの複雑化につながる問題点を指摘するシステムを構築する。

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を用いる必要があるため、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。しかし教育者に対しての支援を目的として構築されたシステムであるため、即座にミスのフィードバックを示すことができない等の学習者のプログラミング学習を支援する上での課題点が存在する。そこで本研究では、学習者のソースコードのエラーを即座に特定し、エラー改善の方針を示すことが可能なシステムを構築することを目的とする。また、実装に Haskell を用いることで調査項目の拡張性に優れたシステムを目指す。

3. システムの概要

本研究のシステムは、入力された C 言語のソースコード中の間違っている可能性の高い箇所の指摘を行う。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、Haskell のライブラリである language-c-quote によって構文解析を行う。その解析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、間違っている可能性の高い記述をしていないか調査を行う。問題がある箇所の行番号とその内容やエラーの修正方針を HTML 形式で解析結果を表示するページに出力する。以下に本研究のシステム概要図 (図 1) を示す。

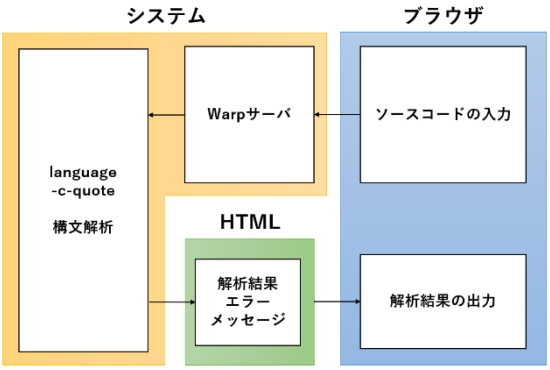

図 1 システム概要図

4. 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、パターンマッチングや型推論、Datatype-generic programming 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。

本システムで調査できる現在の項目は、インデントが規定のルールと異なっているミス、printf と scanf の受け取る引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。主に C 言語を学び始めたばかりの学習者が間違いやすい項目の調査の実装を優先的に行った。解析結果の出力表示では入力されたソースコードと、ソースコードの問題がある行番号の色を変更し、他の行に比べて目立つように表示しているためエラーのある箇所を視覚的に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 2) と解析結果の表示画面 (図 3) を示す。それぞれの画面はクリックひとつで即座にページの遷移を行うことができるため、利用者に速やかにフィードバックを示すことができる。

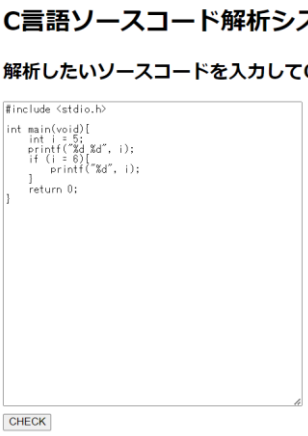

図 2　ソースコード入力画面

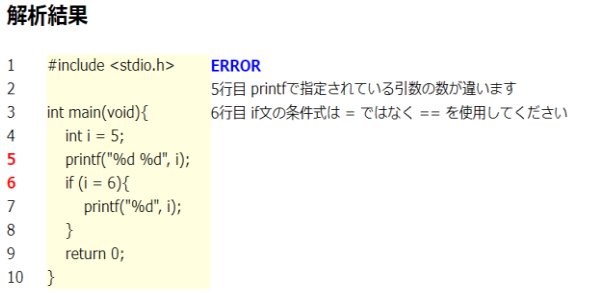

図 3　解析結果表示画面

5.　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や問題点に対して調査を行う機能を Web ベースシステムで実装した。また、ソースコードの構文解析結果から Haskell によって調査したい箇所を抜き出すことで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調査対象の追加が求められる。特に、初学者が正しく使用することが難しい for 文や、宣言した変数の型と異なる型の値を代入しているミスといった項目の実装を考えている。他にも無駄な if 文や意味のない代入のような、コンパイラでは検査することができないソースコードの冗長な部分の調査を行うことでソースコードの簡潔化や学習者のプログラミングスキルを上達させる支援ができると考えている。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。また、Web ベースでのシステム実装を行ったが、Web ベースである利点を活かせていない課題点も存在する。例えば掲示板のような学習者間でコミュニケーションを行える機能や、他のシステムで利用しているソースコードを本システムと送受信するような仕組みを実装するといった別の Web ベースシステムとの連携が考えられる。

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, A3-1, 2015.
(2)　サイボウズ・ラボユース, Kota Uchida:“C-Helper GitHub”, https://github.com/uchan-nos/c-helper (閲覧日：2023 年 5 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.