

プログラミング初心者向けの Web 学習支援システム

Web-based learning support system for beginner programmers

藤坂 直輝[†]山田 光穂[†]石井 英里子[‡]星野 祐子[†]

Naoki Fujisaka

Mitsuho Yamada

Eriko Ishii

Yuko Hoshino

1. はじめに

プログラミング初心者、なかでも 1 言語以上の学習経験がある初心者はプログラミングを学習する際に、表示されるコンパイルエラーのメッセージがどのような間違いなのか推測できないことが多い。特に、複数のプログラミング言語の文法を混在させたときに、記述文が適切と思い込んでしまう場合や目視しづらい全角括弧や全角空白を入れた場合に、修正が必要な個所を見つけるまでに相当の時間を要することがある。これはある程度の熟練者でも陥りがちな問題であり、初心者にとってプログラミングは難しいという心証を与える要因の一つと考えられる。そのため、初心者の学習に悪影響を及ぼすことや、挫折することが考えられる。すなわち、プログラミング初心者が理解しにくいコンパイルエラーを特定し、記述誤りと正規の文法をわかりやすく提示することで学習効率が上がる可能性があるということである。そこで、我々はコンパイラによって明示的に出力されないエラー内容を初心者が理解しやすく提示し、正規の文法を提示することにより学習効率を向上させることを目的とした Web システムの開発を行っている。東海大学情報通信学部では 1 年次の春学期に Python を学習し、続いて秋学期に C 言語を用いる授業を履修する学生が多いため、本研究では Python の基本的な知識を持つ初心者向け C 言語学習支援システムの試作を行った。

2. 関連研究

プログラミング学習を支援する研究には、初心者がコンパイルエラーメッセージを理解できるように和訳や分類をする研究や、コンパイルエラーから修正案を提案する研究が行われている。西村ら[1]はコンパイルエラーメッセージの内容が英語である点や、初心者が構文知識の不足から対処方法がわからない点、適切な修正方法を選択できない点の支援を行う Web システムを提案した。該当システムは、エラーメッセージの和訳データベースから検索を行い、日本語化されたエラーメッセージを表示する機能や、エラーメッセージから解決策を提示する機能がある。後藤ら[2]はコンパイルエラーメッセージに対応した補助メッセージを追加し、エラーの原因をわかりやすく表示した。プログラムの綴りミスにレーベンシュタイン距離を使用し、正しい綴りを提案するデバック支援の研究を行った。近藤ら[3]は初心者がひとつの構文ミスにより複数のコンパイルエラーメッセージが生じることが学習を妨げる原因であると述べ、ひとつの構文ミスから生じる複数のコンパイルエラーメッセージを分類するシステムを開発した。いずれの研究も、コンパイルエラーメッセージを理解するための支援であるが、初心者が複数のプログラミング言語を混在させた記述

誤りでは、適切であると思いつく可能性があるため初心者がコンパイルエラーメッセージを理解しても修正にかなりの時間を要する場合がある。

複数のプログラミング言語を混在させた記述誤りの研究には、蜂巢ら[4]の複数のプログラミング言語の文法知識に起因する制御文の誤りに対する自動修正方法の提案がある。提案手法では、プログラミング言語の共通モデルを作成し、ある言語を記述中に他の言語を記述した際、記述中の言語に対応する文法に修正が可能である。この自動修正ツールは Visual Studio Code にプラグインとして導入することで動作する。本研究では誤りを提示し、初心者と考えさせることで、何が間違いなのかを理解することが学習において重要と考え、自動修正の機能は実装しないこととした。また、特別なソフトのインストールなしに使用できる Web 上で動作するシステムにした。

Web 上で動作するプログラミング学習支援システムの例として、松本ら[5]によって開発された Hello C がある。このシステムは、教授者が用意した課題以外の学習者が自ら作成したプログラムに誤りがある場合の支援が難しい。石嶋ら[6]は制御構文のブロック構造の可視化し、実行軌跡と現在の変数の値を表示することができるプログラミング学習支援環境である、AZUR を使用した Web システムを開発した。しかし、該当システムは制御構文のブロックを可視化することで、構文の理解を支援するものであるため、構文ミスの修正支援ができない。その他、Web ブラウザ上でプログラムの実行が可能な環境として ideone[7]、codepad[8]、paiza.IO[9]などが存在する。これらのサービスでは Web ブラウザ上で、C 言語、C++、Python など複数のプログラミング言語のコーディングと実行が可能である。ideone と codepad は表示が英語のみであるため、英語が苦手な人は使いにくい可能性がある。なお、これらのシステムは他の言語の文法が含まれるプログラムの誤りを検知する機能や、正規の文法を提示することなどの学習支援機能がない。

```
#include <stdio.h>

int main(void){
    for(i in range(5)){
        printf("a");
    }
    //全角空白
}
```

図 1 記述誤りがある C 言語のプログラム

3. 誤りを明示できないコンパイラ

初心者はコンパイルエラーメッセージを見たとき、エラーが直接的に指摘されていなければ修正や学習を行うことが難しい。複数のプログラミング言語を混在させた場合や

[†] 東海大学大学院情報通信学研究科 Graduate School of Information and Telecommunication Engineering

[‡] 鹿児島県立短期大学 Kagoshima Prefectural College

```

example.c: In function 'main':
example.c:4:6: error: unknown type name 'i'
   4 | for(i in range(5)){
     |         ^
example.c:4:11: error: expected '=', ',', ';', 'asm' or '__attribute__' before 'range'
   4 | for(i in range(5)){
     |             ^~~~~~
example.c:7:2: error: stray '\343' in program
   7 | . ^
example.c:7:3: error: stray '\200' in program
   7 | 續 ^
example.c:7:4: error: stray '\200' in program
   7 |  .1;31mm
     |     ^
example.c:8:1: error: expected expression before '}' token
   8 | }
     | ^
example.c:8:1: error: expected expression before '}' token
example.c:8:1: error: expected expression before '}' token

```

図 2 出力されるコンパイルエラーメッセージ

全角空白を入れたプログラムを実行した場合に、出現するコンパイルエラーは誤りを直接的に指摘できない。例えば、Python を学習後に、C 言語を学習すると、C 言語のプログラムの一部を Python の文法で記述してしまうことがある。図 1 は C 言語のプログラム中に Python の for 文を記述し、全角括弧と全角空白を入れてしまった誤りの例である。図 1 のプログラムを実行すると、図 2 のコンパイルエラーメッセージが出力される。出力されるコンパイルエラーメッセージはコンパイラによって多少の違いはあるが、通常のコンパイラはエラーの箇所と種類をメッセージとして出力する。学習者はこのコンパイルエラーメッセージを見て、プログラム中に誤りがある箇所を修正し、エラーを解消する。C 言語のプログラム中に Python の構文を混在させた場合は図 2 の「error: expected~」と表示され、次に予期する記号を示す。全角空白を入れた場合は、「error: stray\343」と表示され、文字コードでエラーがあると示される。これらのメッセージは複数のプログラミング言語を混在させたことや全角空白が入ってしまったことを直接的に指摘していない。熟練者であれば、このエラーから記述誤りを推測することができるが、構文知識が曖昧な初心者にとって、このエラーメッセージから原因を特定することは難しい。また、複数のプログラミング言語を混在させた場合は、プログラムの作成者が記述したプログラムを適切であると思ひ込み、誤りに気づかないことがある。

4. 本システムの概要

このシステムは Web 上で動作し、プログラムを入力して、記述誤りを確認するものである。すなわち、修正と確認を繰り返しながら学習することを想定しており、演習をしたり教科書のような解説を載せたりする学習用コンテンツは持たない。

4.1 システムの仕様

本システムが行う処理の流れを図 3 に示す。プログラミング学習者が Web ブラウザを使用して、C 言語のプログラムを入力すると、システムがそのプログラムの解析を行う。複数のプログラミング言語の構文を混在させた場合や、全角空白、全角括弧などを含ませた場合に、エラーメッセージを出力し、文法に誤りがあれば正規の文法を提示するシステムである。学習者はエラーメッセージを見てプログラ

ムを修正することで、学習を行う。本システムの開発には、Web 画面の作成に HTML と CSS、データベースマネジメントシステムに MySQL、解析を行うプログラムの作成に PHP を使用した。入力されたプログラムに対して字句分割、字句解析、文法確認の順に処理を行い、最後にエラーメッセージを出力する。図 4 は実際に作成した Web ページの初期表示画面である。

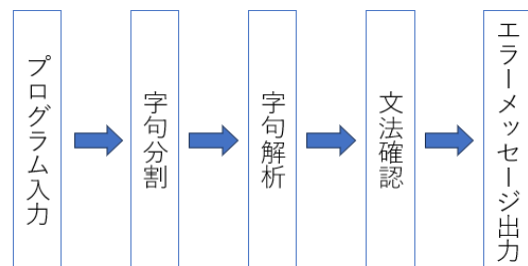


図 3 システムの流れ

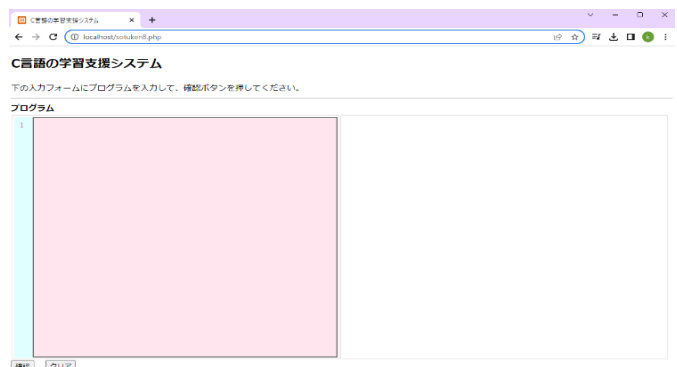


図 4 作成した Web ページの初期表示画面

4.2 字句分割

字句分割は、プログラムを演算子や記号などで区切り、個々の要素である字句に分割する機能である。入力されたプログラムを読み込んで、字句(文字列)を生成する。字句分割の例を図 5 に示す。アルファベット文字、数字、アンダースコア、ドットは連結して一つの文字列とする。空白、エスケープシーケンス、記号は連結せずに文字列の区切り

目とする。図 5 の例では、「==」のように連続する記号による演算子は、記号の数で意味が変化するため、最初の記号「=」をもとに演算子データベースを検索し、後続する記号の判定を行う。連続する記号の数が多い順に検索し、一致する記号の並びが見つければ、それを字句とする。また、ダブルクォーテーションで囲まれた文字列データは、最初のダブルクォーテーションから対応するダブルクォーテーションまでを読み込み、一つの字句とする。同様の手法で、シングルクォーテーションやコメントアウトなども処理する。その後、これらの字句の文字列、プログラム中の位置、改行コードから数えた行番号を字句オブジェクトの配列に保存する。

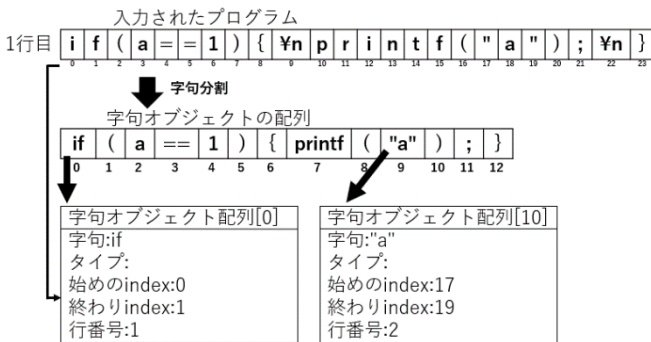


図 5 字句分割の例

4.3 字句解析

字句解析は分割された字句を解析し、タイプを設定する機能である。字句は演算子、記号、予約語、変数名や関数名、C 言語にない字句に分けられる。もし字句がアルファベットである場合、事前に番号が設定された C 言語の予約語データベースから予約語に対応する番号を抽出し、その番号をタイプとして設定する。一方、もし字句が記号である場合は演算子データベースからタイプを抽出し、表 1 のようにアルファベット 1 文字でタイプを設定する。

表 1 演算子のタイプ表

演算子名(タイプ)	演算子記号
比較演算子(C)	<,<=,>,>=,!=
ビット演算子(B)	&,&^,&&,>>,<<,&~
計算演算子(A)	+,-,*,/,%
論理演算子(L)	&& ,!
代入演算子(E)	=
代入演算子(P)	+=,-=,*=,/=,%=<<=,>>=,&=,^=, =
カウンタ演算子(I)	++,--
コメントアウト(O)	//content\n,/*content*/
括弧	(,),[,],{,}
その他の記号	?,:,:;,()

演算子のタイプは比較演算子(C)、ビット演算子(B)、計算演算子(A)、論理演算子(L)、代入演算子(E)、複合代入演算子(P)、カウンタ演算子(I)とする。括弧やセミコロン、コンマなどの記号はその記号をタイプとして設定する。コメントアウトされた字句のタイプはコメントアウト(O)とする。数値のタイプは数値データ(N)とし、ダブルクォーテーションで囲まれた字句を文字列データ(S)とする。その他の字句は、他の言語の予約語やプログラマが設定した変数名や関数名と考えられるため、事前に登録した他の言語の

予約語データベースを検索し、存在しなければタイプを(ID)と設定する。字句解析のタイプ設定例を図 6 に示す。

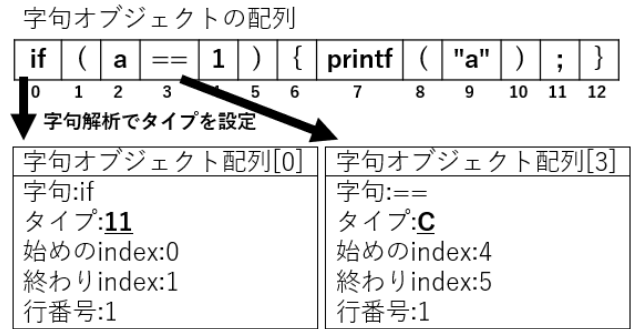


図 6 字句解析のタイプ設定例

続いて、括弧の中身をまとめ、字句の塊を作成する。これは、次に行う文法確認で括弧の中身を参照しないためである。そのため、字句オブジェクトとは別に括弧内オブジェクトを作成する。字句オブジェクト及び括弧内オブジェクト配列への保存例を図 7 に示す。括弧内オブジェクトは小括弧(), 中括弧{}, 大括弧[]の中身をそれぞれタイプ T, G, K とする。そして、括弧のタイプ、プログラム中の場所、括弧の中身を保存する。字句オブジェクトと括弧内オブジェクトを組み合わせることで、文法の確認を容易にした。また、括弧の中身は配列として保存することで再帰処理を可能にした。

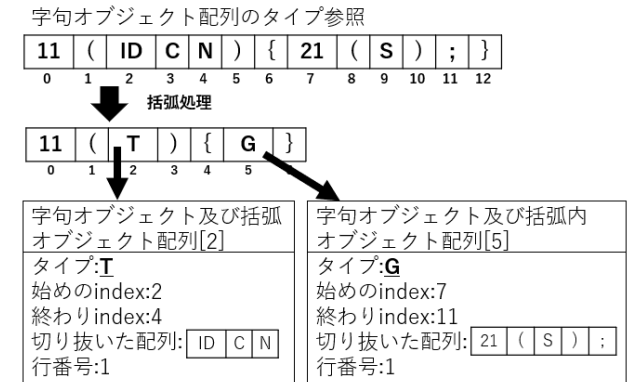


図 7 字句オブジェクト及び括弧内オブジェクト配列への保存例

4.4 文法確認

文法確認では字句の並びと文法の並びを照合して、正誤を判定する。字句解析で得た字句オブジェクト及び括弧内オブジェクト配列から予約語である、数字タイプをキーに文法データベースを検索し、登録されている字句の並び順と一致しているか確認する。括弧内オブジェクトであるタイプ T や G などが出現した場合、その中身の配列についても文法確認を行う。if 文の文法確認の例を図 8 に示す。字句解析で得た数字タイプ 11 番を用いて文法データベースを検索し、結果と字句のタイプの並びを確認する。if 文や for 文などでは続く処理が 1 行しかない場合、中括弧{}が省略できるため、データベースにタイプ D(1 行の処理)と登録し、文法確認の際に、セミコロンまで確認する処理にした。複数の文法が存在する予約語は少なくとも 1 つの文法が合っているならば正しいと判断する。予約語の番号から

文法が正しいと判断された場合、タイプ T(条件など)のアウトタイプにその予約語の番号を設定する。アウトタイプとは小括弧()の中身であるタイプ T がどの予約語と対応しているかを示したものである。文法確認が終了した後、アウトタイプを参照し、括弧の中身がその文の条件に合っているかを確認する。しかし、これは文ごとで特性が異なるため、文 1 つずつに確認をするプログラムの作成が必要である。今回は for 文の小括弧()のみ確認するプログラムを作成した。

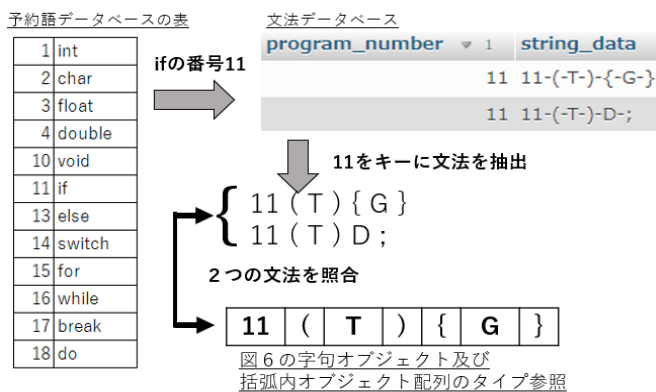


図 8 文法データベースの照合例

4.5 エラーメッセージ

このシステムの各機能で発見したエラーはエラーメッセージ配列にエラー内容と行番号を保存し出力する。字句分割ではクォーテーションの対がない場合、全角空白や全角括弧などが含まれている場合はエラーを追加する。字句解析では、言語にない記号や括弧の数が合わない、または他の言語の予約語があるとエラーを追加する。文法確認ではデータベースから予約語に対する正しい文法を検索し、1つも合致するものがなければ、その予約語の文法を書き方データベースから参照し、エラーメッセージ配列に書き方を追加する。また、タイプ T(条件)の中身と合致する文法構造が存在しなければ同様に文法を書き方を追加する。最後に確認ボタンが押された際にこれらのエラーメッセージを実行画面の右側に出力する。文法誤りのエラーメッセージは、該当する予約語を青文字とし、マウスカーソルを予約語の上に乗せると正規の文法を表示する。また、3節で述べたように、複数のプログラミング言語を混在させた場合や、全角空白を入れた場合に出力されるエラーメッセージは初心者が理解しにくいことを考え、エラーを平易な日本語で説明し、Python の予約語があれば、Python の予約語や演算子であると出力するようにした。図 9 に誤りの内容及び正規の文法を提示した例を示す。

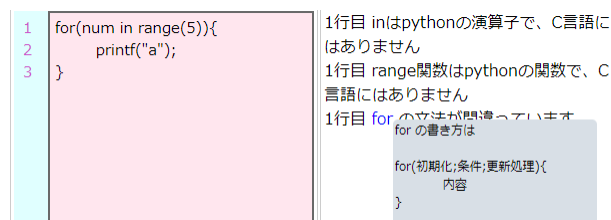


図 9 文法の誤り内容及び正規の文法提示した例

5. 課題及び解決案

このシステムの課題として、予約語の綴り誤りが存在すると変数名等を表すタイプ(ID)とされ、正しく文法を確認できない点や、switch 文の文法における、case や default など、特定の文でしか扱えない字句の文法の正誤を判定できない点がある。さらに、if 文や for 文など、小括弧の中身が予約語に対応したものであるかをひとつひとつ確認するプログラムを作らなければならない点がある。これらの解決策として機械学習を用いることで、字句のタイプの並びから文法モデルを作成して、入力された構文がどのモデルに近いのかを判定することや、過去の入力ミス学習データとし、綴りミス等を検知できる可能性があると考えられる。

6. おわりに

本研究ではコンパイラによって明示的に出力されないエラー内容を初心者が理解しやすく提示し、正規の文法を提示することで学習効率を向上させる Web システムの開発を目指した。その結果、字句解析にて字句を演算子、予約語、記号と分類し、他言語の予約語が含まれていた場合にエラーを出力すること、記述されたプログラムの文法を確認し、誤りがあれば正規の文法を提示することができた。しかし、switch 文の文法における、case や default などの特定の字句はその種類ごとに正誤を判定するプログラムを作る必要がある。今後の展望として、機械学習を用いてユーザがミスしやすい場所を推測することや、初心者と熟練者がコンパイルエラー画面のどこを注視しているのかを調査し、初心者が誤りをより容易に発見できるようなインタフェースの開発に取り組む予定である。また、プログラミング学習者を対象に、利便性や学習効率の定量的評価やアンケート評価を行う予定である。

謝辞

本研究の一部は JSPS 科研費 JP20K12411, JP23K11635 の助成を受けた。

参考文献

- [1] 西村将広, 橋浦弘明, 古宮誠一 “プログラム初学者へのコンパイルエラー修正支援システム—エラーメッセージを用いた修正支援—” 情報システム学会第 6 回全国大会・研究発表大会論文集 2010, pp.1-4
- [2] 後藤孔, 藤中透 “プログラミング教育におけるデバッグ支援” システム制御情報学会論文集 Vol. 32, No. 6(2019), pp.249-255
- [3] 近藤亮太, 名倉正剛 “コンパイルエラーメッセージの分類に基づく初学者へのプログラミング学習支援手法” 第 9 回 実践的 IT 教育シンポジウム rePIT2023 in 函館 p.63-74
- [4] 蜂巣吉成, 東直希, 三上比呂, 長野滉大, 吉田敦, 桑原寛明 “複数のプログラミング言語の文法知識に起因する制御文の誤りの自動修正方法の提案” 日本ソフトウェア科学会 2022 年 39 巻 4 号 p.4_38-4_48
- [5] 松本慎平, 大下昌紀, 買田康介 “C 言語諸学者及びその教授者のためのサーバ・クライアントに基づくプログラミング学習支援システムの開発” 電気学会論文集 C 2020 年 140 巻 9 号 p.1096-1109
- [6] 石嶋慧, 平川豊, 大関和夫 “ウェブブラウザを用いたプログラミング学習支援環境” 報処理学会全国大会講演論文集 Vol.78, pp.349-350 (2016)
- [7] ideone.com <https://ideone.com/> (参照 2023-06-16)
- [8] codepad <http://codepad.org/> (参照 2023-06-16)
- [9] piazza.IO <https://piazza.io/> (参照 2023-06-16)