

連想検索に基づく C プログラミング演習コース設計システムの実現

森田 航[†] 立岩 佑一郎[‡] 高橋 直久[‡]

[†]名古屋工業大学大学院 〒466-8555 愛知県名古屋市昭和区御器所町

[‡]名古屋工業大学 〒466-8555 愛知県名古屋市昭和区御器所町

E-mail: [†] morita@moss.elcom.nitech.ac.jp, [‡] {tateiwa, naohisa}@nitech.ac.jp

あらまし プログラミング演習を行う場合、指導者は単元ごとに学習する文法と関数を含む演習問題を作成し、演習コースを設計する必要がある。本稿では、以下の特徴をもつプログラミング演習コース設計システムを提案し、その実現法について述べる。(1) 初学者用の教科書や参考書に掲載された例題などからなるプログラムデータベース(ProgDB)を構築し、指定された文法と関数を含むプログラムを ProgDB から求める連想検索機能を実現する。(2) インストラクタが各単元での学習と ProgDB のプログラム例を結び付けて学習項目を定めると、各単元に対し、プログラム例に含まれる構文と関数から既習項目を考慮して学習範囲(新出範囲、復習範囲、既習範囲など)を定める学習範囲導出機能を実現する。(3) 各単元に対して、学習範囲に基づいて演習問題の正解例となるプログラムの集合を ProgDB から検索して推薦する演習問題推薦機能を実現する。

キーワード e-Learning, ソースコード検索, プログラム解析, プログラミング演習

1. はじめに

我々はプログラミング演習システム CAPES[1]を開発し、初学者向けのプログラミング学習に適用している。この演習では、受講者は、C 言語の文法と関数の使用法を習得するため、与えられた課題に対する C プログラムを作成する。演習は単元に沿って進められる。まず指導者は、繰り返し文や分岐文など、単元において学習させたい内容を決める。そして C 言語の教科書などをもとに、各単元で学習させたい内容に即した文法、関数の項目を学習項目として定め、学習項目の使い方や、実際にそれらの項目を使用した例題プログラムが含まれた講義資料を作成する。そして、それらの項目を含むようなプログラムを作成する演習問題を作成することで、演習コースを設計する。

演習コースを設計する場合、指導者は学習項目を含むようなプログラムを正解例として用意し、それを用いて演習問題を作成する。本稿では、演習問題の作成を容易にするため、過去の演習課題、初学者用の教科書や参考書に掲載されたプログラムからなるデータベースを構築し、データベースからプログラムのソースコードを検索できるシステムを実現する。指導者は、単元で学習する学習項目を含むようなソースコードをデータベースから検索して正解例に用いることで、容易に演習問題を作成することができるようになる。しかし、学習範囲をもとに検索を行う場合、次のような問題点がある。

問題点 1 キーワードによる検索では、キーワードが正しく定められていないと、学習項目を含むソースコードが見つけれられない場合がある。

問題点 2 文法や関数は種類が多いので、指導者が

検索に用いる学習項目を指定するのは、手間がかかる。

問題点 3 検索結果として多数のソースコードが該当した場合、指導者が演習問題に用いるソースコードを選ぶのに時間と手間がかかる。

本稿では、これらの問題点を解決するために、各単元で学習する項目が含まれている講義資料の例題をもとに検索に用いる学習項目を自動で定める機能と、求められた学習項目をもとにソースコードの検索及び推薦を行う機能を考える。

指導者はこれらの機能を用いることで演習問題の作成の手間を省くことができる。

2. 関連研究

プログラミング演習における問題作成支援のための研究、データベースからソースコードを検索する研究は、種々行われている。

文献[2]は、初学者向け C プログラミング演習に用いられる記入式 Web 試験 DrillS-F における類題生成を支援する Web オーサリングツールについて記述している。DrillS-F は、初学者の C 言語の理解支援のため、ソースコードの一部を受講者に提示し、出力値などを回答させるソースコード読解問題の出題を行うことができるシステムである。文献[2]では、DrillS-F における指導者の類題作成を支援するため、1 つのソースコードから異なる類題を自動生成するための Web オーサリングツールの開発を行っている。これに対し本稿では、受講者がソースコードを作成する問題を出題する場合において、指導者の問題作成を支援するため、受講者の作成するソースコードに対する正解例を指導者に推薦することで問題作成を支援できるシステムを提案している。

文献[3]は、ソースコードの構造比較によって類似するソースコードを検索する手法を提案している．具体的には、Java ソースコードを解析し、使用されたメソッドと制御文からソースコードの構造を求め、ソースコード同士の構造を比較することで類似するソースコードを検索する手法について述べている．

これに対し本稿では、C プログラムのソースコードを解析し、関数や制御文に加え、データ型の種類など、初学者向け演習で学習する項目の有無を求め、検索に利用している．ソースコードが使用している文法や関数を特性として取り出して検索に利用している．

文献[4]は、ソースコードの選択の基準となるような特性を求める手法を提案している．具体的には、異なるバージョンのソフトウェアの差分などから、信頼性や機能性を抽出する手法について述べている．これに対して本稿では、指導者は演習で使用したい文法や関数を持つソースコードを必要とするという点から、ソースコードが使用している文法や関数を特性として取り出して検索に利用している．

また、ソースコードを検索するためのシステムとして、開発者のためのソースコード検索システムである Krugle[5]がある．Krugle は、オープンソースのプログラムを対象とし、その中からユーザに指定されたキーワードを持つプログラムを検索できる．しかし、演習に使用するプログラムを検索したい場合、ポインタなどを持つプログラムを、キーワードを用いて検索するのは難しい．そのため本研究では、演習に使用するプログラムを検索するために、文法や関数を条件として用いた検索を行うためのシステムであるという点で、Krugle と異なる．

3. 学習項目と学習範囲

3.1. 学習項目

本稿では、C 言語で用いられる表 1 の 18 種類の文法と、表 2 のヘッダファイルで宣言されている 464 種類のライブラリ関数を学習項目と呼ぶ．これらの 482 種類の学習項目は表 3 の学習項目定義表で定められる．各学習項目は記号 v_i ($i=1, \dots, 482$)によって示す．また学習項目定義表では、文法、関数のどちらを示す学習項目か表される．学習範囲の記号の全体集合を V_0 とし、 $V_0=\{v_i\}(i=1, \dots, 482)$ となる．

表 1 学習項目となる文法

if 文	else-if 文	else 文
while 文	for 文	自作した関数
switch 文	構造体	関数形式マクロ
do-while 文	再帰関数	配列
ポインタ	文字列	多次元配列
共用体	整数型	浮動小数点型

表 2 用いるライブラリ関数が含まれるヘッダファイル

assert.h	locale.h	stdlib.h
complex.h	math.h	string.h
ctype.h	setjmp.h	time.h
fenv.h	signal.h	wchar.h
inttypes.h	stdio.h	wctype.h

表 3 学習項目定義表

記号	学習項目	種類
v_1	if	Syntax
v_2	else-if	Syntax
...		
v_{482}	wctrans	Library

3.2. 学習項目の分類

学習項目を、各単元で学習する内容に基づき、分類を行う．学習項目は、学習された時期や各単元の学習内容によって、以下の 4 つの集合に分類できる．

新出範囲 単元で新しく学ぶ学習項目の集合．単元 n における新出範囲を V_{new}^n とおく．

復習範囲 過去の単元で学習済みであり、現在の単元で再度学習を行う学習項目の集合．単元 n における復習範囲を V_{rev}^n とおく．

既習範囲 過去の単元で学習済みであり、現在の単元では学習を行わない学習項目の集合．単元 n における既習範囲は V_{std}^n とおく．

未習範囲 単元 n の時点でまだ受講者が未学習である学習項目の集合．単元 n における未習範囲を V_{not}^n とおく．

これら 4 つの範囲は排反となる．また、新出範囲と復習範囲の和集合が単元で学習するすべての学習項目の集合となり、これを**学習中範囲**とし、単元 n における学習中範囲を V_{cur}^n とおく．

学習項目は、単元で学習中の項目がより単元において重要な項目となり、またその中でも新しく学ぶ項目のほうがより重要となる．よって、各範囲の集合内に含まれる学習項目の重要度を、図 1 のように表す．

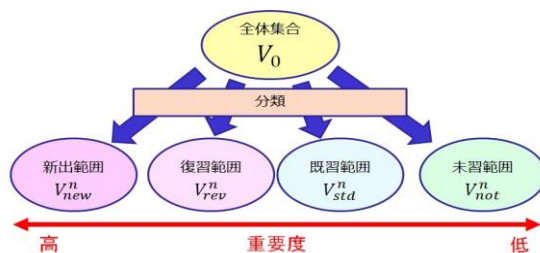


図 1 範囲と重要度

4. 提案システム

4.1. 演習コース設計の方針

本システムでは、節 3.2 で定めた学習項目の分類と

重要度をもとに、以下の3つの方針に即して演習コースが設計できるようになる機能を提案する。

方針1 新出範囲に含まれる学習項目は重要度が高い項目なので、各演習問題に新出範囲の学習項目のいずれかが必ず使用されるようにする。

方針2 新出範囲と復習範囲の学習項目は、単元で学習すべき項目であるので演習コースにおいて網羅されるようにする。

方針3 受講者がまだ学習していない未習範囲の学習項目は、演習問題では使用しないようにする。

4.2. 提案システムの構成

4.2.1. 使用するデータベース

当システムで使用するデータベースとして、ProgDBとExamDBの2つを用意する。

ProgDBには、演習問題の作成を容易にするため、過去の演習課題、初学者用の教科書や参考書に掲載されたプログラムなどのソースコードが格納されている。また各ソースコードには、通し番号としてIDがつけられている。

ExamDBには、演習の講義資料で使用された例題が格納されている。また、演習の各単元における既習範囲、復習範囲、新出範囲、未習範囲が格納されている。

4.2.2. システムの特徴

特徴1 ソースコードの連想検索

ソースコードに含まれる学習項目を解析によって求めて検索条件に用いることで、指導者の指定した学習項目を持つソースコードを連想的に検索する。

解析を用いることで、キーワードによる検索よりも確実に学習項目を持つソースコードを見つけられるようになる。

特徴2 例題をもとにした範囲の導出

講義資料の例題を解析して、含まれている文法と関数を求め、それをもとに各単元における学習項目の分類を自動で行う。これによって、指導者が学習項目を手作業で節3.2のように分類する手間を省くことが可能になる。また、分類した範囲を特徴1の指定に用いることで、方針1,3に即したソースコードを容易に見つけられるようになる。

特徴3 設計方針に即したソースコードの推薦

特徴1で得られた検索結果のソースコードの集合に対し、節3.2で得られた学習範囲の分類と重要度を用いて、重要度の高い学習項目を含むようなソースコードの順位が高くなるようにソースコードに順位付けを行い、上位のソースコードを指導者に推薦する。また、複数のソースコードを検索する場合、指導者が既に選択したソースコードに従って、重要度の高い項目の中でもまだ選択したソースコードにおいて使用されていない項目を持つようなソースコードの順位が高くなる

ように順位付けを変更し、指導者に再度推薦を行う。

これによって指導者は、方針2に即すようにソースコードを選ぶことが容易になる。

4.2.3. システム構成図

システムの構成図を図2に示す。システムは、範囲導出機能、検索機能、演習問題推薦機能の3つの機能と、ProgDB、ExamDBの2つのデータベースからなる。

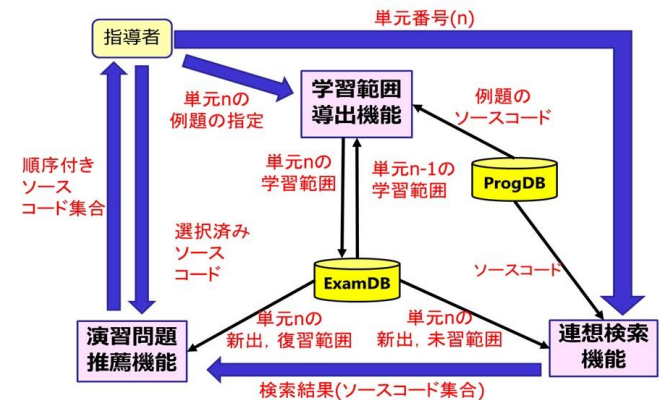


図2 システム構成図

範囲導出機能は、指導者から単元nの講義資料の例題として用いられたソースコードを指定されると、ProgDBから指定されたソースコードを抽出する。また、ExamDBに格納されている単元n-1の学習範囲(新出範囲、既習範囲、復習範囲、未習範囲)を抽出する。抽出したソースコードと新出範囲 V_{new}^{n-1} 、復習範囲 V_{rev}^{n-1} 、既習範囲 V_{std}^{n-1} 、未習範囲 V_{not}^{n-1} をもとに、単元nの学習範囲(既習範囲 V_{std}^n 、復習範囲 V_{rev}^n 、新出範囲 V_{new}^n 、未習範囲 V_{not}^n)を導出し、ExamDBへ格納する。

検索機能は、指導者から検索を行う単元の番号nを入力されると、ExamDBから単元nの新出範囲 V_{new}^n と未習範囲 V_{not}^n を抽出する。そして V_{new}^n と V_{not}^n をもとに演習問題の正解例のソースコードに必要な項目を求め、それを含むようなソースコードの集合をProgDBから検索し、ソースコードの集合を検索結果として出力する。

演習問題推薦機能は、検索機能から入力された検索結果のソースコードの集合が入力されると、単元nにおける復習範囲 V_{rev}^n 、新出範囲 V_{new}^n をExamDBから抽出する。そして、検索結果のソースコードに、 V_{rev}^n 、 V_{new}^n を用いて優先順位をつけ、検索結果のソースコードに優先順位を付与した集合を指導者に提示する。また、ソースコード集合の中から演習問題に使用するソースコードが指導者に選択され、演習問題推薦機能へ入力されると、演習問題推薦機能は選択済みのソースコードで使用された項目を元に、優先順位をつける際の条件を変更し、それをもとに再度順序を付け、指導者に提示する。

5. システムの実現法

5.1. データベースの実装

5.1.1. ProgDB

ProgDB には、プログラムのソースコード本文である program, ソースコードの通し番号となる ProgID, プログラムのタイトルを表す progtytle を表 4 のようなソースコード表として格納する.

表 4 ソースコード表

progID	progtytle	program
1	文字の出力	#include<stdio.h> int...
2	for 文の練習	#include<stdio.h> int...
3	while 文の練習	#include<stdio.h> int...

5.1.2. ExamDB

ExamDB には、演習の単位において使われた例題の情報を格納する. ExamDB に格納されている表を表 5 に記述する. まず unitID には、その例題が使われた単位の番号が記述されている. booksymbol には、その単元で使われていた教科書を表す名前が記述され、chapterID にはその中で、単元の例題に使用された例題が booksymbol で示した教科書の何章で使用されていたものであるかが記されている. progIDlist には、booksymbol,chapterID で示されている教科書の章における例題の中で、演習の例題で使用した例題の progID が含まれている. 学習範囲を導出するために、単元 n で使用された例題を求める場合、unitID が n である項を見て、progIDlist に含まれている progID に示されているソースコードを progDB から抽出して行う.

表 5 例題定義表

unitID	booksymbol	chapterID	progIDlist
1	book1	1	1,2,3,
1	book2	1	4,5,
2	book2	2	10,11,

また ExamDB には、各単元における学習範囲(既習範囲 V_{std}^n , 復習範囲 V_{rev}^n , 新出範囲 V_{new}^n , 未習範囲 V_{not}^n)を定義した表も格納されている. 各範囲の定義表を表 6 に記述する. unitID には単元の番号, new,cur, std,not にはそれぞれ V_{new}^n , V_{rev}^n , V_{std}^n , V_{not}^n ごとに含まれている項目の記号が記述されている.

表 6 範囲定義表

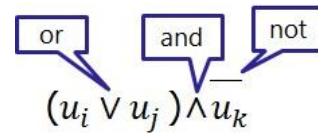
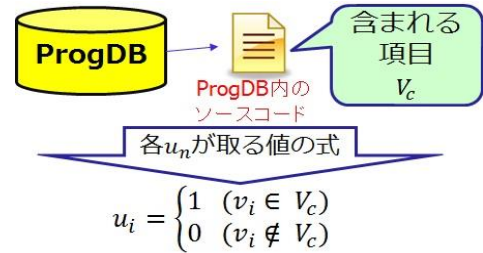
unitID	new	rev	std	not
1	$V_1, V_2,$			$V_3, V_4, \dots$
2	$V_3, V_4,$	V_2	V_1	$V_5, V_6, \dots$

5.2. 検索機能

5.2.1. 検索式

検索機能において、プログラムが持つ学習項目を条件とした検索を行うために、ソースコードが持つべき学習項目の条件を表した論理式である検索式を用いる.

まず論理変数 u_i ($i=1, \dots, 482$) を定義する. この u_i はそれぞれ i の値が等しい学習項目 v_i に対応している. この u_i が取る値はソースコードごとに異なる. 論理変数 u_i の値の定義の方法を図 3 に示す. あるソースコードを C とおき, C において使用されている学習項目を V_c とすると, ソースコード C に対する論理変数 u_i の値は $u_i=1$ (V_c が v_i を含む場合)もしくは $u_i=0$ (V_c が v_i を含まない場合)と定義される. 図 4 のように、これらの論理変数 u_i を論理記号で結合した式を検索式として用いる.



5.2.2. 検索式の作成

検索機能でははじめに、新出範囲 V_{new}^n と未習範囲 V_{not}^n をもとに、提案手法の方針 1, 方針 3 を満たすソースコードの検索式を作成する.

検索式の作成方法は図 5 のようになる. はじめに、検索する単元 n の新出範囲 V_{new}^n , 未習範囲 V_{not}^n を ExamDB から抽出する. そして、以下の STEP に沿って検索式を作成する.

STEP1 V_{new}^n に属する項目に対応する論理変数 u_n を全て求める.

STEP2 STEP1 で求められた論理変数 u_n を or 演算子でつなげ、検索式を作成する.

STEP3 V_{not}^n に属する項目に対応する論理変数 u_n を全て求める.

STEP4 STEP1 で求められた論理変数 u_n を not にして and 演算子でつなげ、検索式を作成する.

STEP5 STEP2,STEP4 で作成された検索式同士を and 演算子でつなげる.

このように行くと、STEP2 で作成される検索式は方針 1 を満たすソースコードの検索式, STEP4 で作成されるのは方針 3 を満たすソースコードの検索式となる. よって、これらの検索式をつなげた STEP5 で作成される検索式は、方針 1, 方針 3 の双方を満たすソースコ

ードが検索される検索式となるので、これをマッチングに利用する。

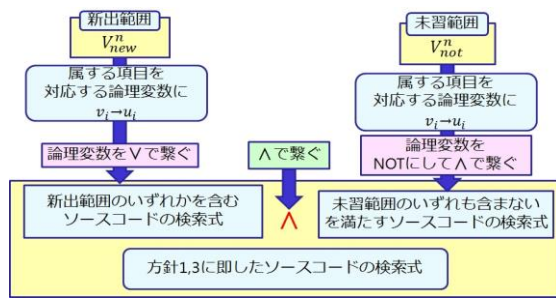


図5 検索式の作成

5.2.3. マッチング

検索式と ProgDB 内のソースコードでマッチングを行う。まずソースコードを解析し、使用されている文法、関数を求める。これにより、各ソースコードにおいて各論理変数 u_i が 1 と 0 のどちらの値を取るかが決まる。各論理変数の値が決まると、その値を検索式の各記号へと代入する。代入した場合の、検索式全体が取る値が 1 となれば、ソースコードは検索式の表す検索条件を満たすことになる。このマッチングを ProgDB 内の全ソースコードに行い、検索条件を満たしたソースコードの集合を検索結果として出力する。

5.3. 学習範囲導出機能

学習範囲導出機能では、例題に用いられたソースコードをもとに、学習範囲の各範囲に含まれる項目を導出する。単元 n における学習範囲を求める場合、指導者に指定された単元 n の講義資料の例題と、ExamDB に格納された単元 $n-1$ までの新出範囲 V_{new}^{n-1} 、復習範囲 V_{rev}^{n-1} 、既習範囲 V_{std}^{n-1} 、未習範囲 V_{not}^{n-1} を導出に用いる。

5.3.1. 学習中範囲の導出

まず、講義資料の例題を用いて単元 n の学習中範囲 V_{cur}^n を導出する。まず単元 n で使用された講義資料の例題を全て ProgDB から抽出する。各例題には、単元 n で学習する項目が含まれているため、各例題を解析し、使用されている項目を求めると、それが各例題で学習する項目と考えられるので、全ての例題で使われた項目の和集合が単元 n で学習中の範囲であると考えられるので、これを学習中範囲 V_{cur}^n とする。

5.3.2. 新出範囲、復習範囲の導出

単元 n の新出範囲 V_{new}^n は、学習中範囲 V_{cur}^n の中で、単元 $n-1$ まででまだ学習していない項目が含まれることになるので、 $V_{new}^n = V_{cur}^n - (V_{new}^{n-1} + V_{std}^{n-1} + V_{rev}^{n-1})$ として求められる。また復習範囲は、学習中範囲の中で、新出範囲に含まれない項目の集合となるので、 $V_{rev}^n = V_{cur}^n - V_{new}^n$ で求められる。

5.3.3. 既習範囲、未習範囲の導出

既習範囲 V_{std}^n は、単元 $n-1$ までに学習済みの項目の

中で、復習範囲に含まれない学習項目の集合となる。よって、 $V_{std}^n = (V_{new}^{n-1} + V_{std}^{n-1} + V_{rev}^{n-1}) - V_{rev}^n$ となる。また未習範囲 V_{not}^n は、単元 $n-1$ の未習範囲のうち、単元 n でも学習されなかった項目となるので、 $V_{not}^n = V_{not}^{n-1} - V_{new}^n$ で求められる。

5.4. 演習問題推薦機能

演習問題推薦機能では、検索機能から求められた検索結果のソースコードの集合に対し、ソースコードに順序を付け、指導者に提示する。重要度の高い学習項目を持つソースコードの推薦を行うよう、復習範囲 V_{rev}^n 、新出範囲 V_{new}^n を用いて順序付けを行う。順序付けの方法は、指導者が演習問題に使用するソースコードを選んでいない場合と選んでいる場合で変わる。順序付けを行う際につける順位は、1,2,...という順につけられ、1 が順位が一番高くなる。

5.4.1. ソースコードが選択されていない場合

重要度の高い項目を持つソースコードを選ぶために、新出範囲、復習範囲を多く持つソースコードを優先して提示する方法を考える。そのため、以下の STEP で順位付けを行う。

STEP1 ExamDB から復習範囲 V_{rev}^n 、新出範囲 V_{new}^n を求める。

STEP2 検索結果のソースコードに対し、ProgID の値が小さいものから順に順位が 1,2,...となるように順位をつける。

STEP3 バブルソートを用いて、順位が 1 つ違いであるソースコード同士を比較し、条件 A,B のいずれかを満たした場合、その 2 つのソースコードの順位を入れ替える。

条件 A ソースコード内で使用されている V_{new}^n の項目の数を比較し、順位が高いソースコードのほうが少ない場合。

条件 B ソースコード内で使用されている V_{new}^n の項目の数が等しければ、ソースコード内で使用されている V_{rev}^n の項目の数を比較し、順位が高いソースコードのほうが少ない場合。

5.4.2. ソースコードが選択されている場合

既に指導者が使用するソースコードをいくつか選択している場合、次に選択するソースコードは、 V_{rev}^n 、 V_{new}^n の中でまだ使用していない項目を持つソースコードを推薦するようにする。そのため、以下の STEP で順位付けを行う。

STEP1 ExamDB から復習範囲 V_{rev}^n 、新出範囲 V_{new}^n を求める。

STEP2 指導者が選んだソースコードを全て解析し、使用されている項目を求める。この項目の集合を V_u とおく。

STEP3 V_{rev}^n の集合から V_u に含まれている項目を除き、未使用の復習範囲の項目の集合を求める。この集合を V'_{rev}^n とおく。

STEP4 V_{new}^n の集合から V_u に含まれている項目を除き、未使用の新出範囲の項目の集合を求める。この集合を V'_{new}^n とおく。

STEP5 検索結果のソースコードの集合から指導者が選択したソースコードを取り除き、ProgID の値が小さいものから順に順位が 1,2,...となるように順位をつける。

STEP6 バブルソートを用いて、順位が 1 つ違いであるソースコード同士を比較し、条件 A,B のいずれかを満たした場合、その 2 つのソースコードの順位を入れ替える。

条件 A ソースコード内で使用されている V'_{new}^n の項目の数を比較し、順位が高いソースコードのほうが少ない場合。

条件 B ソースコード内で使用されている V'_{new}^n の項目の数が等しければ、ソースコード内で使用されている V'_{rev}^n の項目の数を比較し、順位が高いソースコードのほうが少ない場合。

6. 評価実験

1597 個のソースコードを有する ProgDB を構築し、提案システムのプロトタイプを開発した。これを用いて、検索・推薦したソースコードが演習問題の正解例として適するかを実験的に調べた。

実験では、名古屋工業大学の C プログラミング演習における講義資料の例題を用いて、推薦した上位 10 個のソースコードに対して、教員や TA 経験者などの被験者 11 名へ 5 段階評価のアンケートにより演習問題の正解例としてそのソースコードを使用したいか調査した。

評価結果を図 6 に示す。

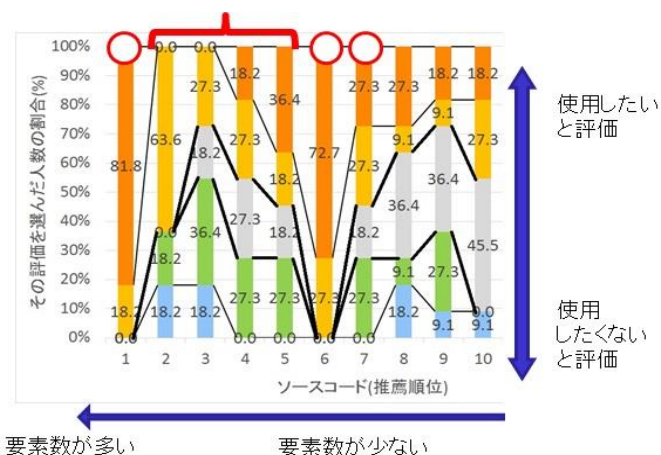


図 6 評価結果の分布図

図 6 では、丸が付いているソースコードが特に被験

者からの評価が高いソースコードとなる。図 6 を見ると、優先順位が高いソースコードでも、評価の低いソースコードが含まれていることがわかる。

次に、上位 10 個のソースコードを、新出範囲の出現回数を考慮し、出現回数が多い順に並べ替えた場合の図を図 7 に示す。

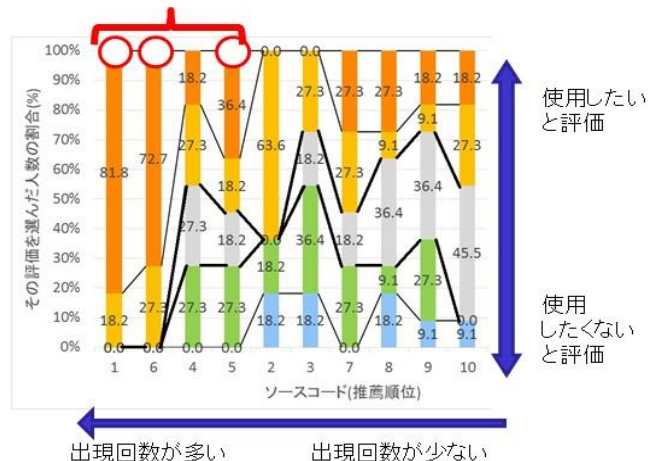


図 7 並べ替えた場合の分布図

図 6 に対し、図 7 を見ると、被験者からの評価が高いソースコードが上位に集まっていることがわかる。

これらのことから、順位付けの方法によって、評価の低いようなソースコードが上位となる場合や、評価の高いソースコードが上位となる場合があることがわかる。これらのことから、正解例として適するようなソースコードを提案システムが推薦できるようにするために、複数の順位付けの方法を実装し、指導者が使用する方法を選べるようにすることが必要であると考えられる。

7. まとめと今後の予定

本稿では、プログラミング演習における演習コース設計を容易にするため、講義資料で用いられた例題をもとに単元で学習する範囲を求める手法を提案し、それをもとに演習問題として使用するソースコードを検索できるシステムの実現法について記述した。今後の予定として、ソースコードの過去の出題履歴や難易度などを推薦における条件に加えた順位付けなどを行う予定である。

参考文献

- [1] 中島秀樹, 高橋直久, 細川宜秀, “プログラミング学習のための QA サイクル受講者の習得度に応じた問題自動提示メカニズム”, 電子情報通信学会論文誌, VOL.J88-D-I, NO.2, 2005
- [2] 国信真吾, 富永 浩之, 林 敏浩: 記入式 Web 試験 DrillIs における C プログラムのソースコード読解問題 一類題生成を支援する Web オーサリングツールの開発ー, 電子情報通信学会技術研究報告, ET, 教育工学 108(470), 217-222, 2009-02-28
- [3] 宇田川佳久: 制御構造とメソッド名に基づくソースコードの類似検索について, 電子情報通信学会

技術研究報告. SWIM, ソフトウェアインタプライズモデリング 112(331), 29-34, 2012-11-23

- [4] 小西裕司, 荻原剛志, “ソースコードのバージョン間比較による情報抽出とソースコード再利用への応用”, 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス 106(427), 7-11, 2006-12-08
- [5] Krugle <http://www.krugle.com/>
- [6] Java <https://java.com>
- [7] MySQL <https://www-jp.mysql.com/>