数式解答評価システム STACK を利用した プログラミング演習用問題

伊藤隼人*1・北英彦*1
Email: hitou@ce.elec.mie-u.ac.jp

*1：三重大学大学院 工学研究科 電気電子工学専攻

◎Key Words プログラミング演習，数式解答評価システム STACK，自動採点

1. はじめに

近年, 多くの大学で教師の負担を軽減するために, e ラーニングシステムが導入されている．その中の有用なもののひとつにオンラインテストがある．従来のオンラインテストは，多肢選択か文字列として完全に一致するものしか自動採点することはできず, 解答の表現の自由が高い数式を短答記述式かつ自動採点で自然に扱うことは難しかった．そこで，STACK[(1), (2)]や Maple T.A.[(3)]など, e ラーニングシステムの機能として，学生が入力した数式を解答として受け付け，自動で正誤評価・採点を行うサブシステムがいくつか開発された．

本研究では，プログラムのコードには数式と一致している部分や似ている部分があるので，STACK を用いてプログラムのコードを自動で正誤評価・採点できないかということを検討した．その結果，for 文や while 文などの条件式の部分や代入文の順番に関する穴埋め問題を作成することができたので，それについて報告する．また, STACK を用いることにより教師の負担は軽減されるのかを検証し，その結果を報告する．

2. 数式解答評価システム STACK

2.1 概要

数式解答評価システム STACK（System for Teaching and Assessment using a Computer algebra Kernel）とは, 数式処理システム Maxima[(4)]を用いた，数式解答評価システムであり，e ラーニングシステム Moodle のオンラインテスト（小テストモジュール）で，学生の数式による解答を受け付け，正誤評価・自動採点を可能にするシステムである[(5)]．全てオープンソース・ソフトウェアで構成されている．

例えば，微分方程式を解く問題などでは，任意定数が必要となる．その場合，学生の解答を自動評価するにあたって，以下のような困難が発生する．

- 任意定数に用いる文字は自由であり，あらゆる文字が使われる可能性に対し，もれなく解答を準備することは不可能である．
- 任意定数が抜けているだけの解答に対して部分点を与えたい．

STACK は，これらの問題を解決している．問題の解決にあたって大きな役割を果たしているのは，STACK において最も特徴的な機能といえるポテンシャル・レスポンス・ツリーである．

2.2 ポテンシャル・レスポンス・ツリー

ポテンシャル・レスポンス・ツリー[(5)]とは，互いに関係づけられた任意の数のポテンシャル・レスポンス（想定される学生の解答パターン）で構成される，学生の解答を処理するための機構である．

図 1 は，ポテンシャル・レスポンス・ツリーの概念図であり，No: 0~No: 6 がポテンシャル・レスポンスであり，それぞれに対して評価関数による判定を

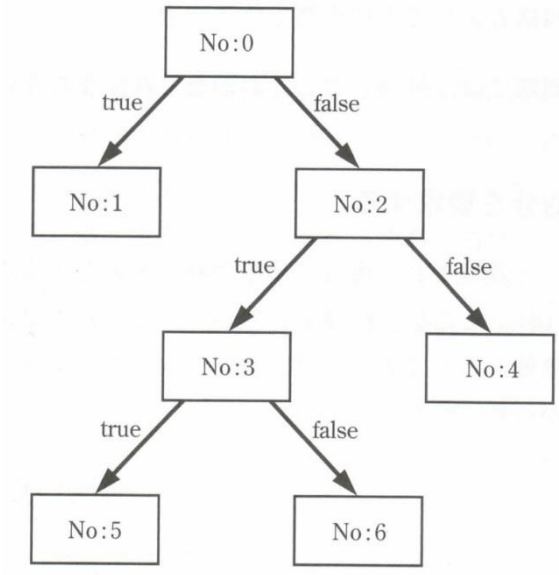


図 1　ポテンシャル・レスポンス・ツリー概念図

行う. 評価関数を満たす(true)か満たさない(false)かによって，次のポテンシャル・レスポンスに推移していきながら，学生の解答に対する採点を行っていく.

これによって，学生の解答が最終的にどこに行き着いたかによって, 与える点数を変えることができ，惜しい解答には部分点を与えることができる.

2.3 評価関数

評価関数[5]とは，ポテンシャル・レスポンスの，評価対象（学生の解答など）と評価基準を引数として各種判定処理を行う関数である．様々な関数が用意されているが，ここでは研究で使用した 2 つの関数のみ紹介する.

- 代数等価

  最もよく使われる関数で，評価対象と評価基準が代数的に等しいかどうか，すなわち，（評価対象）－（評価基準）が 0 となるかどうかを判定する.

  これにより，学生の解答が，評価基準と表現が違っても同じ意味であれば, true となる.

- 構文等価

  評価対象と評価基準の書式が等しいかどうかを判定する．代数等価と比べて，等号・不等号をはさんで反転させたものを false とする.

  例えば，評価基準を a=b として，学生が b=a と解答しても false となる（代数等価では true となる）.

3. プログラミング演習への活用

この章では, STACK が実際にプログラミング演習に利用できるのかを検証する．しかし，プログラムのコード全てを自動評価することは不可能である.

そこで，プログラムのコードの中でも，数式として扱うことのできる部分を抜粋し，穴埋め形式にすることで問題を作成できるのではないかと考えた.

STACK での解答の入力書式は，基本的に Maxima に従うが，プログラム中でもよく行う基本演算（四則演算，べき乗，剰余）は，ほとんどが C 言語と同じコマンドである．唯一，剰余だけは，C 言語で使う“%”ではなく，mod(a,b)（a/b の余り）というコマンドを使う[6].

なお，本研究では，以下の 2 つにしぼって STACK で問題を作成した.

- 条件式（不等式のみ）
  - while 文
  - for 文
- 代入文

なぜこの 2 つにしぼったかというと，数式と同じ形で表現されることが多いからである.

以降，C 言語の演習用教材[7]を参考にして実際に作成した問題を紹介する.

3.1 条件式

3.1.1 while 文

図 4 は, while 文の条件式の部分を穴埋めにした問題の解答画面である.

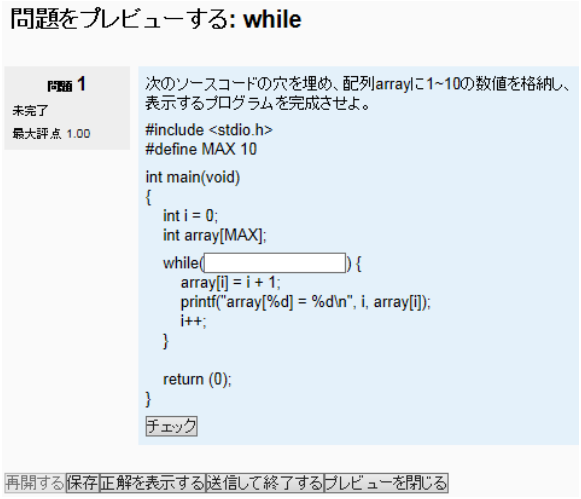


図 4　条件式（while 文）の問題

この問題の正答は,

- i < MAX
- MAX > i
- i <= MAX - 1
- MAX - 1 >= i

の 4 通りである.

この問題において，Moodle では，上記の 4 つ全てを評価基準として用意しておかなければ正解とすることはできない．また，スペースの数まで完全に一致しなければ正解とすることができないため，学生が入れるスペースの数の可能性の分だけ評価基準を用意することは困難である.

しかし，STACK では，i<MAX と i<=MAX-1 の 2 つを評価基準として用意し，評価関数を代数等価にしておけば，上記の 4 つ全てをスペースの数に関係なく正解とすることができる.

3.1.2 for 文

図 5 は, for 文の条件式の部分を穴埋めにした問題の解答画面である.

1 つ目の解答欄では，単一文字の解答欄により 1 文字しか入力できない形式にしてある.

この問題においては，1 つ目の解答欄の解答欄 ID（ans1）を変数として，2 つ目の解答欄の評価基準の式に組み込み,

- i < MAX + ans1
- i <= MAX - 1 + ans1

の 2 つを用意しておけば，1 つ目の解答欄の解答が 0~9 のどれであっても対応できる．

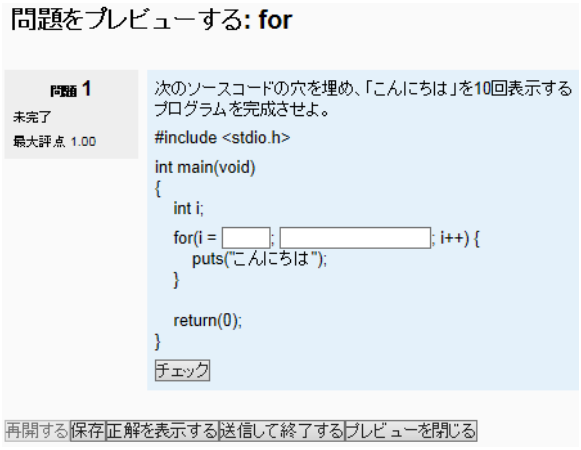


図 5　条件式（for 文）の問題

この問題においても，STACK では，while 文の問題と同様に，表現が違っても同じ意味であれば正解とすることができる．

また，Moodle 単体では，解答欄はそれぞれ独立にしか評価ができないため，上記のように 1 つ目の解答欄の解答によって，2 つ目の解答欄の解答の正誤判定を変えるということはできない．

しかし，STACK においては，これが可能である．すなわち，STACK においては，同じ問題中の複数の解答欄の解答を組み合わせて評価・採点することができる．

2.2　代入文

図 6 は，代入文を 3 つ使用して 2 つの変数 a と b の値を入れ替える問題の解答画面である．

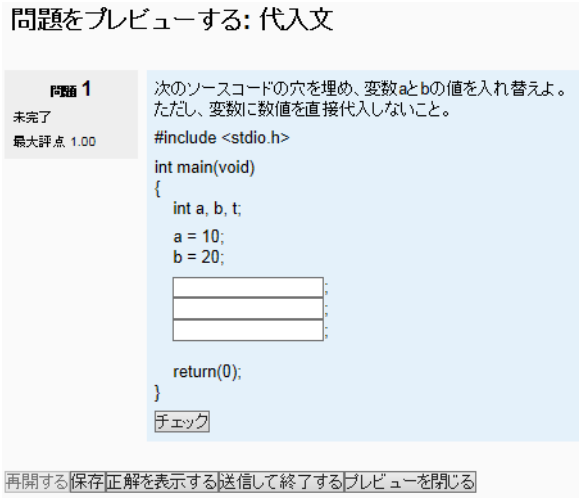


図 6　代入文の問題

この問題の正答は，

- 一番上：t = a
  真ん中：a = b
  一番下：b = t
- 一番上：t = b
  真ん中：b = a
  一番下：a = t

の 2 通りである．

この問題においては，条件式の問題のように評価関数を代数等価にしてしまうと，等号をはさんで変数の順番を入れ替えても，STACK ではあくまで等式・方程式として扱われているため正解としてしまう．

したがって，この問題においては，解答の評価関数として，代数等価ではなく構文等価を用いることにより，等号をはさんで反転したものを不正解とすることができる．

4.　Moodle での解答との比較

Moodle と STACK を組み合わせた時の短答記述式小テストが，Moodle 単体のそれと比べて，自動評価・採点の際にどのような点が優れているかを表 1 にまとめた．

表 1　Moodle と STACK の比較

Moodle	Moodle+STACK
用意された正答と，文字列として完全に一致しなければ正解にできない．	数式として同じ意味であれば正解にできる．スペースの数に関係なく正解にできる．
解答欄はそれぞれ独立にしか評価ができない．	同じ問題中の複数の解答欄の解答を組み合わせて評価・採点することができる．

5.　実験

この研究における最大の目的は，プログラミング演習において教師の負担を軽減することである．しかし，本当に STACK を用いることにより教師の負担を軽減することができるのかを検証するために，以下のような実験を行った．

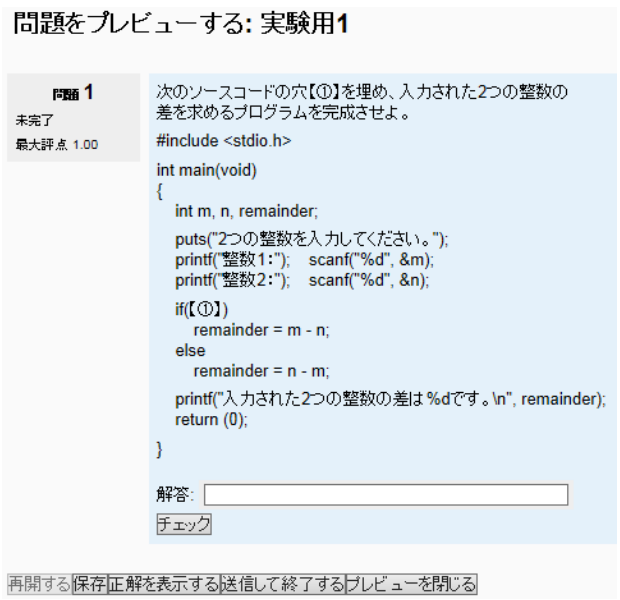
図 7 に示すような問題を，Moodle 単体の短答記述式で作った時と，STACK で作った時とで問題作成に要した時間をそれぞれ計測し，その結果を表 2 に示す．なお，Moodle において用意した正答パターンは，不等号と変数の間のスペースの数が，1 または 0 の場合のみ考慮した．

実験は，Moodle，STACK における問題作成方法を両方理解している被験者 A と，その両方について

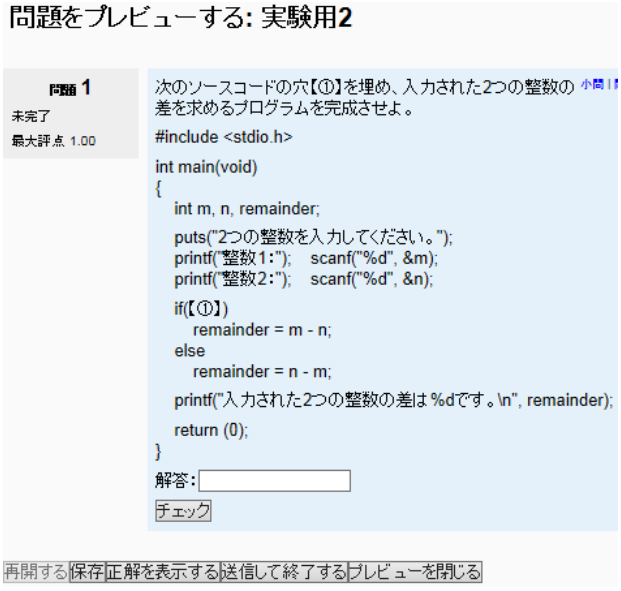
知らない被験者 B，C，D を対象に行った。知らない被験者に対しては，それぞれの問題作成方法を説明し，それにかかった時間も結果に含まれている．

表 2 より，被験者 A, B は，説明時間を含め，STACK の方が短い時間で作成することができ，被験者 C, D は STACK の方が長い時間かかっている．しかし，問題作成のみにかかった時間を見ると，全員が STACK の方がより短い時間で作成することができている．すなわち，作成方法さえわかれば，STACK の方が短い時間で問題を作成できるということがわかった．

とはいえ，それほど差があるとも言えない．しかし，Moodle 単体よりも STACK で作成した問題の方がより自由度の高い（スペースの数に関係なく）解答ができるという点は，大きな利点である．



(a) Moodle 単体（短答記述式）の画面



(b) STACK の画面

図 7 実験用の問題

表 2 問題作成の所要時間

被験者	Moodle		Moodle+STACK	
	説明	作成	説明	作成
A	-	9.4 分	-	8.2 分
B	3.6 分	11.6 分	5.5 分	9.3 分
C	3.6 分	13.3 分	5.5 分	12.7 分
D	3.6 分	11.7 分	5.5 分	10.5 分

6. 注意点

プログラミング演習に STACK を利用するにあたって，以下のような注意点が挙げられる．

- Maxima の書式と異なる数式記号（%，==，!= など）を扱うことができない．
- C 言語でよく使われる変数（tmp など）を STACK では使えないため，学生には使えない変数名をあらかじめ伝える必要がある．

7. まとめ

オンラインテストにおいて，プログラムのコードを自動で評価・採点することは非常に困難である．しかし，本研究で，STACK というシステムを用いることによって，一部ではあるがプログラムのコードの自動で評価･採点ができるということがわかった．

また，短答記述式問題に関しては，Moodle 単体よりも STACK で作成する方が，より短い時間でより自由度の高い問題が作成できることがわかり，教師の負担軽減に繋がると考えられる．

参考文献

(1) STACK, http://stack.bham.ac.uk/
(2) Ja STACK, http://ja-stack.org/
(3) Maple T.A., Maplesoft, http://www.cybernet.co.jp/maple/product/maple_ta/
(4) Maxima, a Computer Algebra System, http://maxima.sourceforge.net/
(5) 中村泰之：“数学 e ラーニング-数式解答評価システム STACK と Moodle による理工系教育”, p.6, p.152, pp.164-166，東京電機大学出版局（2010）
(6) 赤間世紀：“Maxima で学ぶ微分方程式”，p.25，工学社（2013）
(7) 柴田望洋，赤尾浩，肘井信一，高木宏典：“解きながら学ぶ C 言語”，pp.74-83，ソフトバンクパブリッシング株式会社（2004）