

初学者によるプログラムトレーシングにおける誤答に関する分析

関谷 貴之†, 山口 和紀†, 山本 三雄†

概要

プログラミングに関する基本的能力の一つとして、与えられたプログラムの動作を追う能力、すなわちトレーシングが重要である。本研究では、初心者が犯しやすい誤った解釈に基づいて、与えられたプログラムから誤った出力を得る仕組みを作成した。この仕組みを用いて、学生に与えたごく簡単なトレーシング課題の解答から、プログラミング初学者の誤答のパターンを見つけた。さらに、トレーシング能力を問う試験と期末試験の結果を分析することで、ある誤答パターンをする学生は成績が良くないとの知見が得られた。

Error Analysis of Novices' Program Tracing

Takayuki Sekiya†, Kazunori Yamaguchi†, Mitsuo Yamamoto†

Abstract

Tracing skill is considered one of the important fundamental skills to make a program. In this research, we developed a tool to simulate a novice programmer's incorrect tracing process. Using the tool, we identified error patterns from the novices' responses to tracing quizzes. We found out some relationships between the error patterns and the final examination scores.

1 はじめに

プログラミングの初学者に対する教育に関して、プログラミングに関わる能力に注目した様々な研究 [1] [2] が行われている。例えば、Lopez らはプログラムを読むスキル(リーディング*)、変数の値などの変化を追うスキル(トレーシング)、書くスキル(ライティング)などの関係に注目して、期末試験の学生の解答を統計的に分析することで、スキル間に階層関係があることを示し、トレーシングがライティングに強く関わっている事などを発見している [3]。また、Vainio ら [4] は初学者にはトレーシングが困難であるとして、その原因を分析している。

このようにトレーシングがプログラミングの能力で重要なスキルの一つであるならば、そのスキルを身に付けたか否かを把握することは、その後のプログラミング教育の上で役立つと期待される。そのため、例えば繰返しトレーシング問題を解かせる方法が考えられる。しかし変数の値を答えるだけの解答では、トレーシングスキルが身に付いていないことが原因の誤答なのか、単に解答時の不注意だったのかを判断するのは容易でない。

これまで我々は初学者のプログラミング能力について分析することで、for ループや if 文などプログラムを構成する概念による難易度の違いがあることを見出してきた [5]。本研究では、トレーシングにおいて初学者が陥りやすい誤った解釈に基づいて、与えられたプログラムから誤った出力を得る仕組みを作成した。この仕組みを用いて、初学者の誤りのパターンを見つけた上で、更にその誤答パターンと期末試験の結果を分析した。

本論文の構成を以下に示す。まず2節ではトレーシングに関わる関連研究を紹介する。次に3節では、本研究におけるトレーシング能力の分析手法について述べる。4節では、学生に対して実施したトレーシング試験の分析結果から明らかになったことを報告する。最後に5節で全体をまとめる。

2 関連研究

Vainio らは、プログラミングの初学者のトレーシングスキルが劣っている原因として、Single Value Tracing, Confusing Function and Structure, Inability to Use External Representations, Inability to Raise Abstraction Level の4つを特定したとしている [4]。この Vainio らの研究は、トレーシングにおける思考過程を明らかにする試みとして興味深い。原因の特定に当たっては、集中講義に参加した6名の学生を被験者として、トレーシングの問題を解かせながら、どのように考えたかを問う1時間程度のインタビューを多数回行ったとのことである。し

† 東京大学
The University of Tokyo

* 原論文では、読んだコードを説明できるスキルとして、“Explain”と書かれている。

かし、週に1, 2回程度しか開講されない一般教養科目等でのプログラミング教育の講義で、トレーシングの能力を知るためだけに、インタビューなど多大な時間をかけることは容易でない。また、トレーシングのスキルと成績などとの関係も明らかにしていない。

Gobilらは思考過程に沿った一種の構文解析木を作成するなど、異なるタイプの設問に対して、それぞれの方法で誤答のパターンを見つけるを試みている[6]。しかし、新たな設問に対して自動的に誤答のパターンを生成することは行っていない。

プログラミングに限らず、学生の誤答をいち早く発見して学生への指導に活かすシステムとして、ITS (Intelligent Tutoring System) に関する研究があるが、特に本研究で扱うトレーシング能力に限定すると、例えばChouらのProTracer2.0がある[7]。ProTracerは、プログラム実行時の1ステップ毎の変数の値を学生に入力させ、それが正しい値と異なっていれば、自動的或いは半自動的に誤りを指摘したりヒントを与えたりするシステムである。教員の学生指導を補助するシステムとしての機能は高いが、学生が入力すべき内容が多く、学生への負担は大きい。

高岡ら[8]は、e-Learningを主としたプログラミング講義における理解度の低い学習者を見つける手法の提案と実践を行っている。本研究は、初学者の誤答のパターンを見つけることに目的を限定しており、実験を行った講義はいずれも対面型の講義であるため、e-Learningに比べると学生に余分な負担を強いることなく得られるデータは比較的少なく、現時点では高岡らと同様の手法を適用することは容易でないと考えられる。

3 誤答の分析

プログラミング教育を実施する教員にとって、プログラム中の変数の値の変化を追う問題、つまりトレーシング能力を問う問題は、非常に簡単だと考えがちである。しかし、実際に学生に問題を解かせると、必ずしも簡単ではないことが分かる。例えば、図1のRubyのコードでa=3 (正答は9) 及びa=4 (正答は16) を与えた時の出力を問う問題を初学者に与えたところ、正答率が30%未満であった[†]。

我々はforループとif-thenを様々な組み合わせたコードを用いた複数の設問からなるトレーシングの試験を実施した。その学生の解答を比較すると、似た設問で同じような誤りが見られる。つまり、学生は適当に解答しているのでは無く、構成要素を正確には理解していないものの、彼らなりの何らかの規則性に基づいて解答していると考えられる。例えば、図1でa=3及びa=4とした時の出力として、3及び4とする学生が約42%[‡]おり、これはforループによる反復を解せず、ループ内の式が1回のみ実行されるかのような誤った解釈をしていると推測される。このように、同様の構造を持つコードで共通し

て見られる誤った解釈を本研究では誤答パターンと呼ぶ。

```
def a3(a)
  ans = 0
  for i in 1..a
    ans = ans + a
  end
  p ans
end
```

図1: a3.rb

このような誤答パターンを再現するには、初心者の思考過程と同様にコードを解釈する一種のインタプリタを構築する方法が考えられる。しかし、図1に示したような比較的短いコードであれば、想定する誤答パターンに基づいてコードを変換することでインタプリタの代りとなると考えた。

そこで本研究では、以下の手順でトレーシング能力を問う試験の解答結果を分析する。

1. Rubyのコードからruby-parser[§]を用いて構文木を生成。
2. 想定する誤答パターンに基づいて、1で生成した構文木を変換し、変換後の構文木からRubyのコードを生成。
3. 2で得たコードに適当な入力を与えて実行した際の出力を保持。
4. 3と同じ入力を与えて、1のコードを実行した際の出力を問う試験を学生に対して実施。
5. 3の出力と4の解答を比較して、学生がどの誤答パターンに当てはまるかを分析。

以下では、次の点について考察する。

1. パターンの抽出 (4.2節): どのような誤答パターンが考えられるか。
2. 説明可能性 (4.3節): 誤答パターンでどの程度誤答を説明可能か。
3. 成績との関係 (4.4節): トレーシングの能力或いは誤答パターンと成績とは関係があるのか。
4. 妥当性 (4.5節): 誤答パターンは本当に誤答の理由なのか。

4 実験

4.1 実験の概要

本研究で用いたトレーシング能力を問う試験は10個のRubyのコードについて、それぞれ2種類の入

[†]4.1節で述べる実験 G2011S の被験者。

[‡]先の例と同様、4.1節で述べる実験 G2011S の被験者。

[§]<http://rubyforge.org/projects/parsetree/>

力を与えた際の出力を問うもので、全部で20問の設問で構成される。それぞれ異なる講義と異なる学生に対して合計4回実施した。最初の3回は筆記試験として、Rubyのコードをコンピュータ上で実行せずに解答することを被験者に求めた。残りの1回は4.5節で述べるツールを用いたオンラインテストとして実施した。講義の内容や被験者の人数などの基本的な条件を以下にまとめる。

G2011S 2011年度前期に開講されたRubyを用いた初心者向けのプログラミングの講義で実施した。受講者の大部分は、数学教育を専攻する大学2年生でプログラミング初学者である。同じ内容の講義を2クラスで実施しており、1クラス当りの人数は40-50名である。分岐や反復、配列などの基本的な事項に関する10回の授業を終えた後に、模擬試験の一部としてトレーシング試験を実施している。期末試験は大問10問で構成し、2問がトレーシング試験、他8問はトレーシングと直接関係のない設問である。2クラスの学生のうち、解答時に実験への協力を表明した71名の解答を本研究の分析に用いた。

T2011W 2011年度後期に開講された情報科学の基本概念や思考方法をプログラミングを通して習得する講義で実施した。プログラミング言語としてRubyを用いる。受講者の多くは大学1,2年生。理系が多いが専攻は様々である。プログラミングの経験は受講の前提としておらず、多くが初心者である。トレーシング試験は授業中の小テストとして実施した。なお、期末試験にトレーシングの能力のみを問う設問は含まれない。47名の解答を本研究の分析に用いた。

T2012S 2012年度前期に開講されたグラフ、リスト、集合、木などの基本的なデータモデルを扱う講義で実施した。履修条件としてプログラミングの経験を求めている。受講者の多くは大学2年生である。T2011Wと同様に専攻は様々である。当該試験は授業中の小テストとして実施して、81名の解答を本研究の分析に用いた。

G2012S 2012年度前期の講義で実施した。講義の内容や対象となる受講者の専攻はG2011Sとほぼ同じである。トレーシング試験は76名がオンラインテストに解答し、パソコンの不具合などでオンラインテストを受けられなかった5名が、他の実験と同様に紙の解答用紙を用いた。

4.2 パターンの抽出

まず最初に、表1に示す誤答パターンを設定した。これらのパターンは実験G2011Sのトレーシング試

```
def a3(a)
  ans = 0
  ans = ans + a
  p ans
end
```

図3: 誤答パターン“NFL”に基づく図1のコードの解釈

験の解答結果や著者らの過去のプログラミング教育における経験から推測し、かつ3節で述べたコード変換の仕組みで表現できたものを示している。例えば、NFLと呼ぶパターンに基づく、図1のコードは図3のコードとして解釈される。出題された元のコードがループを含んでいない場合は、NFLパターンで解釈しても、元のコードと違いが無く、当然出力も同じになる。また、パターンの適用によって得られたコードが正しく動作しないこともあり得る。参考のために、誤答パターンに基づくコードの解釈を図2に示す。

表1には、試験で用いた20問のうち、それぞれのパターンの適用によって、元のコードと異なる出力が得られる設問数を示している。

4.3 説明可能性

表2には、実験G2011Sの解答について、どのパターンに一致するかを示したものである。解答数は全員の回答を人数で割ったものである。パターン名の列にある「無解答」とは答案の対応する解答欄が空白であった解答を、「一致無し」とは答案の解答欄に記入された値が、いずれの誤答パターンとも一致しないことを意味する。同じ解答を複数の異なるパターンとして解釈可能なこともあるため、全ての誤答のパターンと一致無しの合計は、「誤答」の解答数や全体割合を越えることに注意されたい。G2011Sでは、NFLやCVILに一致する解答が多く、forループを苦手とした学生が多かったことが推測される。なお、いずれかのパターンに一致する誤答が約64.4%で、誤答パターンによって6割以上の誤答を説明できている。

同様に、実験T2011Wの解答を表3に示す。正答率81.7%はG2011Sの正答率55.9%よりもかなり高いものの、誤答のパターンとしてはG2011Sと同様に、NFLやCVILに一致する解答が多い。なお、パターン全体で誤答の約71.7%を説明できている。

次に、特定の誤答パターンに当てはまる学生がどの程度存在するかを考える。個々の被験者を見た場合、多数の誤答の一部が偶然このパターンの誤答と一致することもあり得る。表4は、NFLのパターンで考えられる8個の設問について、実際にそのパターンに一致する誤答の個数と被験者の人数との関係を示している。半分以上一致する学生を、NFLのパターンに基づいて解答したとみなすならば、実験G2011Sではこのパターンで考えたと思われる学生が40.8%存在する。

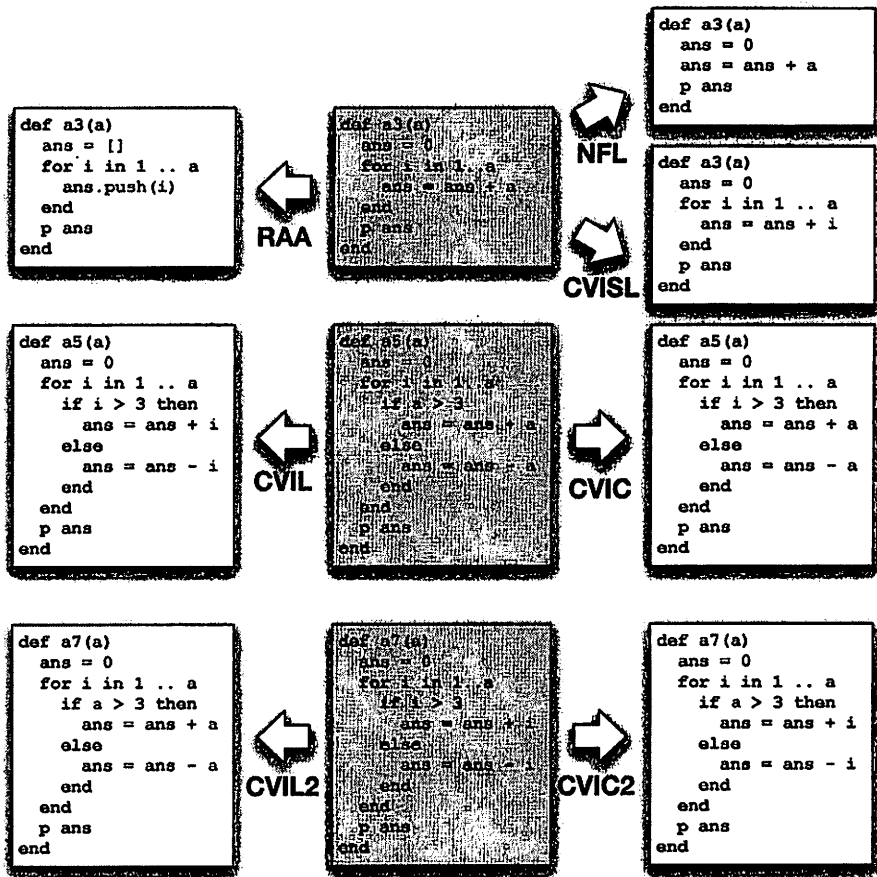


図 2: 誤答パターンに基づくコードの解釈。中央が元のプログラム、それぞれの誤答パターンによって変化した箇所を赤字で示している。

パターン名	説明	設問数
CVIC	for ループ内の if 文の条件式のなかに現れる for 文の終端の変数を, for 文の制御変数 がそこにあると解釈する。	3
CVIC2	CVIC とは反対に, for ループ内の if 文の条件式のなかに現れる for 文の制御変数を, for 文の終端の変数 がそこにあると解釈する。	3
CVIL	for ループ内に現れる for 文の終端の変数を, for 文の制御変数と解釈する。	8
CVIL2	CVIL とは反対に, for ループ内に現れる for 文の制御変数を, for 文の終端の変数と解釈する。	6
CVISL	CVIL と同様だが, for ループの中が代入式 1 個の場合のみ, for 文の終端の変数を for 文の制御変数と解釈する。	4
NFL	for ループを解釈しない。	8
RAA	答えとなる変数を配列と解釈。	13

他の誤答パターンについても、同一の学生が当該パターンでの解釈可能な設問のうち、半分以上の設問でそのパターンの解釈結果と一致する場合を算出したのが、表 5 である。学生全体の中で、66.2% がいずれかのパターンに一致している。実験 G2011S と同様に、実験 T2011W において、パターンに一致する個々の学生の数を示したのが表 6 である。学生の正答率が 81.7% と高いこともあり、いずれか

の誤答パターンに一致する学生は 34.0% と少ないが、NFL のパターンに一致する学生が 21.3% と最も多い。両実験の結果から、誤答の中では NFL のパターンで考える学生が多いことが分かる。

表 2: 実験 G2011S の解答全体

パターン名	解答数	全体 割合 (%)	誤答中の 割合 (%)
CVIC	0.01	0.1	0.2
CVIC2	0.87	4.4	10.7
CVIL	1.38	6.9	17.0
CVIL2	0.17	0.8	2.1
CVISL	0.73	3.7	9.0
NFL	3.07	15.4	37.7
RAA	0.20	1.0	2.4
いずれかに一致	5.24	26.2	64.4
一致無し	2.90	14.5	35.6
誤答	8.14	40.7	100.0
無解答	0.68	3.4	—
正答	11.18	55.9	—

表 3: 実験 T2011W の解答全体

パターン名	解答数	全体 割合 (%)	誤答中の 割合 (%)
CVIC	0.04	0.2	1.2
CVIC2	0.28	1.4	7.8
CVIL	0.79	3.9	22.3
CVIL2	0.02	0.1	0.6
CVISL	0.34	1.7	9.6
NFL	1.66	8.3	47.0
RAA	0.00	0.0	0.0
いずれかに一致	2.53	12.7	71.7
一致無し	1.00	5.0	28.3
誤答	3.53	17.7	100.0
無解答	0.13	0.6	—
正答	16.34	81.7	—

表 4: 実験 G2011S において NFL と一致する個数と被験者数

一致数 (個)	該当者 (名)	該当者 (割合)
0	31	43.7%
1	4	5.6%
2	4	5.6%
3	3	4.2%
4	1	1.4%
5	4	5.6%
6	8	11.3%
7	3	4.2%
8	13	18.3%

4.4 成績との関係

次に、誤答パターンと成績との関係について分析する。4.3 節の後半で示したように、個々の学生について、特定のパターンに基づく誤答の数が当該パターンの誤答の候補と半分以上一致する場合、そのパターンと同じ誤答を行っていると考え、そこで、実験 G2011S と T2011W について、あるパターン

表 5: 実験 G2011S においてパターンに一致する学生

パターン名	学生数	全体 割合 (%)
CVIC	0	0.0
CVIC2	22	31.0
CVIL	7	9.9
CVIL2	1	1.4
CVISL	17	23.9
NFL	29	40.8
RAA	0	0.0
いずれかに一致	47	66.2

表 6: 実験 T2011W においてパターンに一致する学生

パターン名	学生数	全体 割合 (%)
CVIC	0	0.0
CVIC2	4	8.5
CVIL	2	4.3
CVIL2	0	0.0
CVISL	5	10.6
NFL	10	21.3
RAA	0	0.0
いずれかに一致	16	34.0

表 7: 誤答パターンと成績との関係 (実験 G2011S)

パターン名	A	B	C	D
CVIC	0	0	0	0
CVIC2	3	5	5	9
CVIL	2	2	1	2
CVIL2	0	1	0	0
CVISL	4	4	3	5
NFL	1	7	11	10
RAA	0	0	0	0

に基づく解答をした被験者について、期末試験の得点との関係について分析する。

実験 G2011S の学生を、期末試験の成績に基づいて A-D グループに分けている。基本的には四分位だが、最初成績上位 A, B と下位 C, D のグループの人数がほぼ等しくなるように 2 分した上で、それぞれを 2 分した。各グループの人数は 19, 17, 18, 16 名[†]である。トレーニング試験での誤答パターンに一致した学生ごとに、期末試験でどのグループに属するかを示したのが、表 7 である。

同様に、実験 T2011W の学生についても期末試験の得点に基づいて分類した。受験しなかった学生 6 名を除いた上で、実験 G2011S と同様に期末試験の得点によって A-D の 4 つのグループに分けた。各グループの人数は 8, 12, 11, 10 名である。トレーニング試験での誤答パターンに一致した学生ご

[†] トレーニング試験受験者の内、期末試験結果の分析を了承しなかった 1 名を除いたため、合計 70 名である。

表 8: 誤答パターンと成績との関係 (実験 T2011W)

パターン名	A	B	C	D
CVIC	0	0	0	0
CVIC2	0	1	1	0
CVIL	0	1	0	1
CVIL2	0	0	0	0
CVISL	0	2	1	2
NFL	0	0	5	3
RAA	0	0	0	0

表 9: 誤答パターン NFL に一致するグループと一致しないグループ (実験 G2011S)

	最 低 点	平 均 点	最 高 点	分 散
NFL (N = 29)	70.0	132.8	190.0	29.3
Not NFL (N = 41)	50.0	160.5	200.0	38.9

とに、どのグループに属するかを示したのが、表 8 である。

それぞれの実験の結果からは、次のような点を読み取ることができる。

- 実験 G2011S では、誤答パターン CVIL, CVISL に一致する学生については、期末試験の成績にあまり顕著な傾向はない。しかし、CVIC2, NFL パターンに一致する学生については、成績下位の C, D のグループの人数が多い。
- 実験 T2011W では、誤答パターン CVIC2, CVIL, CVISL に一致する学生自体が少なく、あまり顕著な傾向がない。しかし、G2011S と同様に NFL パターンに一致する学生については、成績下位のグループの人数が多い。また、全体で見れば誤答パターンに一致する学生は A グループに属さず、期末試験の成績が低い傾向がある。

なお、実験 G2011S の学生を、NFL パターンに該当する 29 名と該当しない 41 名との 2 つのグループに分けて t 検定を行ったところ、平均値に有意な差があることが分かった (両側検定 $p = 0.001136 < 0.05$)。参考のために 2 つのグループの基本的な統計量を表 9 に示す。

両実験の結果から、トレーシングの段階で誤答パターン NFL に一致する学生は、for 文を正しく理解しておらず、トレーシング以外の能力も問われる試験であっても成績が低くなる傾向があると推測される。従って、その誤った理解を早い段階で正せば、成績の改善に結びつくことが期待される。

なお、実験全体を通じて、NFL 以外の誤答パターンのうち、CVIC, CVIL, CVIL2, RAA にマッチする学生は非常に少なく、典型的な誤答のパターンとは考え難い。G2011S と G2012S の講義では、「1 から 10 までの数の和」や「1 から 10 までの数の積」等

表 10: 実験 T2012S の解答全体

パターン名	解答数	全体 割合 (%)	誤答中の 割合 (%)
CVIC	0.12	0.6	7.8
CVIC2	0.27	1.4	17.1
CVIL	0.43	2.2	27.1
CVIL2	0.00	0.0	0.0
CVISL	0.37	1.9	23.3
NFL	0.17	0.9	10.9
RAA	0.06	0.3	3.9
いずれかに一致	1.04	5.2	65.1
一致無し	0.56	2.8	34.9
誤答	1.59	8.0	100.0
無解答	0.86	4.3	—
正答	17.54	87.7	—

の制御変数を足し合わせたり掛け合わせたりプログラムを for ループの簡単な例として最初に示したため、その印象が残って、CVISL にマッチする学生数が多くなったと考えられる。CVIC2 はトレーシング試験として示された他のコードとの見間違いの可能性がある。いずれも単なる誤解やケアレスミスであるため、CVISL/CVIC2 に該当した学生の成績が特に劣ることが無かったと推測する。

4.5 妥当性

実験 T2011W および G2011S の分析結果から、トレーシングの誤答パターンの存在を推測して、特に NFL パターンに当てはまる学生については、期末試験の成績との間に何らかの関係があることが推測された。しかし、いずれの実験も筆記試験の解答を後日分析しただけで、解答した学生が誤答パターンと同様に考えたかどうかを確認していなかった。

そこで、実験 T2012S において、G2011S や T2011W と同様の筆記試験を行った後、採点した答案を見ながら、問題を解くにあたってどのように考えたかを自己分析するアンケートを実施した。実験 T2012S の解答全体を表 10 に、パターンに一致する学生を表 11 に示す。実験 T2012S の受講者はプログラミング経験者で、トレーシング試験の正答率が 87.7% と非常に高く、NFL パターンに一致する学生が少ないなど、G2011S や T2011W の学生とは性質の異なる可能性がある。

表 12 に実験 T2012S のアンケートの回答の一部を示す。回答からは、CVIC, CVIC2, CVIL/CVISL の誤答パターンに相当すると見られる回答が得られた。但し、自身の誤答に対して説明していない回答もあり、その他のパターンに相当すると見られる回答が無かった。

このように後日学生に尋ねる方法で有効な回答を得るのは、必ずしも容易でないため、トレーシングの試験を実施可能な Web ベースのツールを構築した。図 4 にツールのスクリーンショットを示す。本ツールでは、Web ブラウザを用いて学生がトレーシ

表 12: 実験 T2012S 及び G2012S でのアンケート回答の一部

パターン名	
CVIC	「4行目で a と i をまちがえた (T2012S)」
CVIC2	「if i > 3 の “i” と “a” を混同して, a=4 のときはすべて ans=ans+1 をすればいいものと考えた. (T2012S)」
CVIL/CVISL	「ans=ans+a の a と i として考えたため 1+2+3+4 = 10 としたが 4+4+4+4 = 16 だった. (T2012S)」 「1+2+3 だと思っていた. (G2012S)」
NFL	「ans+a なので i は関係ないので, 3 だと思った (G2012S)」

表 11: 実験 T2012S においてパターンに一致する学生

パターン名	学生数	全体割合 (%)
CVIC	3	3.7
CVIC2	6	7.4
CVIL	0	0.0
CVIL2	0	0.0
CVISL	10	12.3
NFL	1	1.2
RAA	0	0.0
いずれかに一致	19	23.5

ング試験を解答後, 自分の解答と正答とを比較して, 解答に至った思考過程を問うアンケートに回答可能である. サーバ上では, 解答データと共に学生の解答がどのパターンに当てはまるかを判定した結果を保存する.

実験 G2012S で, 本ツールを用いて各設問の思考過程に関するアンケートを実施したところ, たとえば図 1 のコードで $a = 3$ を与えた際の出力を問う設問のアンケートに対して, 「ans+a なので i は関係ないので, 3 だと思った」と回答しており, NFL パターンに相当する思考過程だと考えられる. 実験 G2012S の解答全体を表 13 に, パターンに一致する学生を表 14 に示す. これらは G2011S と非常に近い結果を示している. また表 12 には, G2012S の回答の一部も示している.

5 終わりに

本研究では, プログラミング初学者のトレーシング能力に注目して, その能力を測る試験から誤答のパターンを見つける方法を開発した. 更にトレーシング能力と成績とを比較・分析することで, トレーシング試験の結果と期末試験の傾向に関係があり, 特に for 文を解釈しない誤答のパターンに一致する学生の成績が良くない傾向があることを明らかにした.

今後は, トレーシング試験の結果を踏まえた練習問題を与えることでの効果の測定や, 授業で利用する LMS(Learning Management System) のログイン回数や宿題の提出状況などの本研究で取り扱っていないデータとの相関を調べるなどして, プログラミングを苦手とする学生の発見やその学生の成績の改

表 13: 実験 G2012S の解答全体

パターン名	解答数	全体割合 (%)	誤答中の割合 (%)
CVIC	0.04	0.2	0.4
CVIC2	0.69	3.5	7.9
CVIL	1.36	6.8	15.5
CVIL2	0.02	0.1	0.3
CVISL	0.68	3.4	7.7
NFL	3.06	15.3	34.9
RAA	0.20	1.0	2.3
いずれかに一致	4.93	24.6	56.1
一致無し	3.85	19.3	43.9
誤答	8.78	43.9	100.0
無解答	0.35	1.7	3.9
正答	10.88	54.4	123.9

表 14: 実験 G2012S においてパターンに一致する学生

パターン名	学生数	全体割合 (%)
CVIC	0	0.0
CVIC2	17	21.0
CVIL	9	11.1
CVIL2	0	0.0
CVISL	16	19.8
NFL	34	42.0
RAA	1	1.2
いずれかに一致	52	64.2

善に活用できないかを検討する. また, トレーシング以外のより高度な能力についても, なるべく簡単に把握する事ができないかを検討する.

参考文献

- [1] Michael McCracken, Vicki Almstrum, Danny Diaz, Mark Guzdial, Dianne Hagan, Yifat Ben-David Kolikant, Cary Laxer, Lynda Thomas, Ian Utting, and Tadeusz Wilusz. A multi-national, multi-institutional study of assessment of programming skills of first-year cs students. In *Working group reports from ITiCSE*

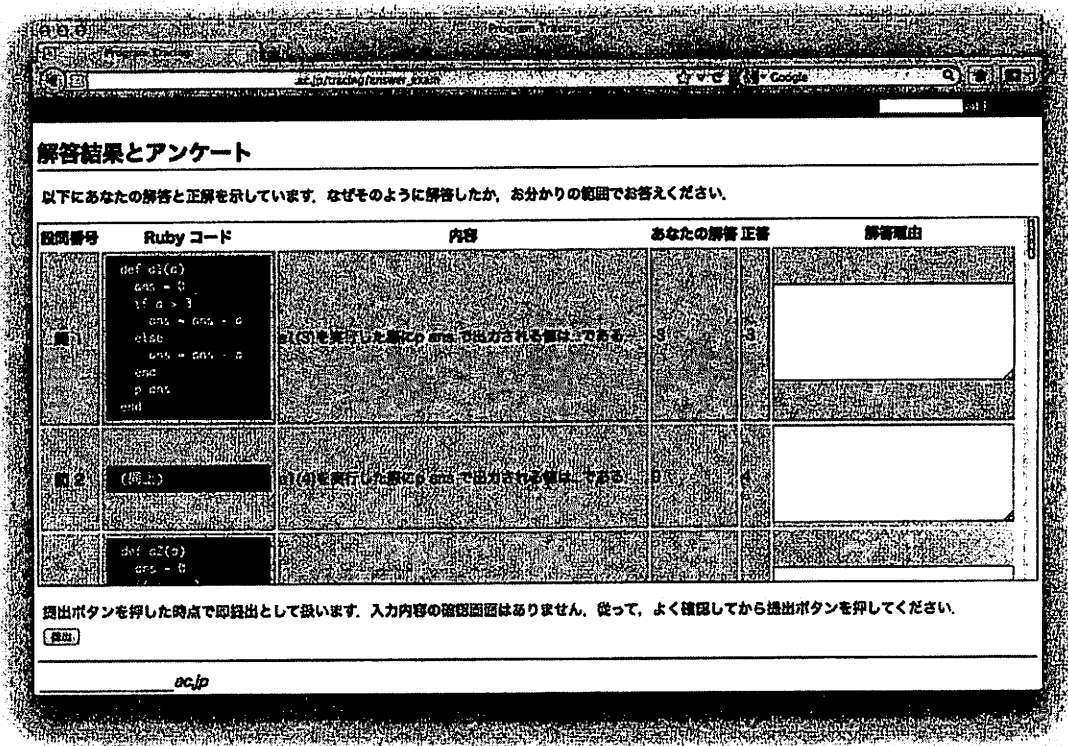


図 4: トレーシング試験用ツールのインタフェース

- on Innovation and technology in computer science education, ITiCSE-WGR '01, pp. 125-180, New York, NY, USA, 2001. ACM.
- [2] Anne Venables, Grace Tan, and Raymond Lister. A closer look at tracing, explaining and code writing skills in the novice programmer. In *Proceedings of the fifth international workshop on Computing education research workshop*, ICER '09, pp. 117-128, New York, NY, USA, 2009. ACM.
- [3] Mike Lopez, Jacqueline Whalley, Phil Robbins, and Raymond Lister. Relationships between reading, tracing and writing skills in introductory programming. In *Proceeding of the Fourth international Workshop on Computing Education Research*, ICER '08, pp. 101-112, New York, NY, USA, 2008. ACM.
- [4] Vesa Vainio and Jorma Sajaniemi. Factors in novice programmers' poor tracing skills. *SIGCSE Bull.*, Vol. 39, No. 3, pp. 236-240, June 2007.
- [5] M. Yamamoto, T. Sekiya, and K. Yamaguchi. Relationship between programming concepts underlying programming skills. In *ITHET 2011, 10th International Conference on Information Technology Based Higher Education and Training*, pp. 1-7, Aug. 2011.
- [6] A. Gobil, Z. Shukor, and I.A. Mohtar. Novice difficulties in selection structure. In *Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference on*, Vol. 02, pp. 351-356, Aug. 2009.
- [7] Chih-Yueh Chou, Bau-Hung Huang, and Chi-Jen Lin. Complementary machine intelligence and human intelligence in virtual teaching assistant for tutoring program tracing. *Computers & Education*, Vol. 57, No. 4, pp. 2303 - 2312, 2011.
- [8] 高岡詠子, 大澤佑至, 吉田淳一. e-learning 学習履歴を用いたドロップアウト兆候者早期抽出手法の提案, 検証および今後の可能性. *情報処理学会論文誌*, Vol. 52, No. 12, pp. 3080-3095, 2011.