流れ図テストによるプログラミング学習でのつまずきの分析

中尾茂子

Ⅰ．はじめに

近年のハードウェアおよびソフトウェア技術の急速な進歩により，一般情報処理教育が従来のプログラミング志向からアプリケーション志向へと流れていく傾向にある中で，大学や短期大学においてプログラミング教育のあり方についてさまざまな調査研究が行われている。情報処理学会での調査研究[(1)]では，一般情報処理教育におけるプログラミング教育を，コンピュータサイエンスの基本的な概念を理解するための技能，と位置づけ非情報系の学生においても必要であるとしている。また，芦葉[(2)]は，プログラミングを問題を構成し解法を作り出してアルゴリズムを実行するための表現方法を工夫する過程として捉え，一般情報処理教育でのプログラミング教育の重要性を述べている。

本学の情報科学コースでは，情報処理技術者育成のために情報科学の知識と技術を習得させる基礎教育を行なっている。その中で，2年次にプログラミング基礎論（通年）とプログラミング基礎演習（後期）を必修科目として設定している。これらのプログラミング教育は情報処理を専門とする学生を対象にしているが，上述のような一般情報処理教育におけるプログラミング教育の捉え方を根底におき，プログラム開発の基礎となる問題分析能力とプログラム設計能力を習得させることを目標にしている。

Ⅱ. 研究の目的

学生は，前期にプログラミング基礎論の講義を受け，後期に講義と並行して演習を行っている。プログラミング基礎演習はコンピュータの動作原理を理解し，アルゴリズムを見出す学習の過程を通して論理的な思考を育成することに重点目標をおき，プログラミング基礎論での学習内容をもとに与えられた問題の解決にあたる。

学生は情報処理を専門としているが，演習開始前に「プログラミングはむずかしい」という感想を強く持っている。初心者の場合，命令の機能や記述など言語の文法を学ぶと同時に問題解決のための手順や考え方も学ばなければならない。したがって，それぞれの学習段階で学生のつまずきが発生する。安達ら[3]は，情報を専門としない学生を対象にプログラミング学習における学生のつまずき個所を分析した結果，抽象化プロセス段階でつまずく学生はプログラミングに直接関わる内容よりは問題解決の手順や方法などの理解不足が見られた。そこで，本研究では，抽象化プロセス段階でのつまずきに関わる要因をさらに分析するために，プログラミング基礎演習の授業開始時に流れ図テストを行い，その結果をもとにつまずきに関わる要因を明らかにすることを目的とする。

Ⅲ. プログラミング学習でのつまずきに関する先行研究

プログラミング教育を実際に行っていく場合，教育内容や教育方法を明確化するためには，学習者のつまずき箇所の分析，プログラミング学習に必要な知識の理解度の把握，学習者の状況に応じた学習支援の検討，学習効果を高めるための要因の分析を行なう必要がある。ここでは，プログラミング学習におけるつまずきに関わる先行研究について調べる。

I. Sommerville[4]は知識情報処理の視点から，初心者にプログラミング教育を行なう場合，言語の文法的知識だけではなく，モデルに埋め込まれた意味的概念を同時に学びとらなければならないが，両者の問題の判別や知識の統合を図っていくことが難しい，と指摘している。

Gerald M. Weinberg[5]はプログラミングを個人の活動として捉えたとき，プログラミング学習を阻害する要因として，新しいものに対する恐怖や失敗への予感などの直接的な要因の他に，間接的な要因として学習内容や方法についての不適切な認識と例題の与え方の２つが考えられると述べている。

高本ら[6]は，プログラミング学習過程から，誤りの分析および学習支援の検討をおこなっている。初心者がプログラミング言語を理解し，自分の考えに基づいて問題解決が行えるようになるまでを模範プログラムの模倣段階，自分なりにプログラミング言語を理解する段階，問題のアルゴリズムを理解しプログラム化できる段階，の３つの段階に学習過程を分けて捉え，また，プログラミングの知識をアルゴリズムの知識とプログラミング言語の文法知識に分けて考え，それぞれの段階で起こす誤りおよび学習者の知識および理解の程度を調べて分析している。その結果，各段階において学習者の理解の程度や知識獲得状況に応じたプログラミング学習支援を検討する必要がある，と述べている。

藤井ら[7]は，プログラミング言語の制御構造の理解調査をもとに学習者の起こす誤りを分類し，類似の誤りをまとめ，それぞれについての学習支援を検討している。

新開ら[8]は，プログラミングの難しさは言語の文法ではなく，アルゴリズムの作成にあると考え，プログラミングにおける問題解決プロセスを抽象化プロセス（プログラミング言語に無関係）と構造化プロセス（プログラム作成）に分け，抽象化プロセスを重視したプログラミング教育を実践している。その結果，プログラミング言語にとらわれることなく，問題を解決するスタイルを身につけさせることができ，問題解決能力が向上し，教育効果を高めることができた，と述べている。

安達ら[3]は，プログラミング学習過程でのつまずき箇所を，抽象化プロセス段階と構造化プロセス段階に大別し，それぞれの段階でのつまずきの要因を分析し，学習支援の方針を見出している。抽象化プロセスでつまずく学生は，プログラミングに直接関わる内容よりはそれ以外の要因で理解不足が見られ，問題を抽象的に捉えることができない，構造化プロセスでつまずく学生は，プログラミング言語の文法的知識でのつまずきが見られる，というつまずきの特徴があると述べている。

筆者ら[9][10]は，プログラマー適性テスト，プログラミングに対するイメージ，コンピュータ親和度の関連性を分析しているが，情意的特性もプログラミング学習に影響を与えるという結果を報告している。

Ⅳ. 研究の方法

問題解決の手順を明らかにしアルゴリズムを構成する際には，モデルに埋め込まれた意味的概念を学びとる力が必要である。したがって，抽象化プロセス段階でのつまずきを誘発する原因としてこの概念の理解力不足が考えられる。そこで，本研究では，プログラミング基礎演習の授業開始時に，抽象化プロセス段階でのつまずきに関する要因を分析するために，一般的な事象の流れ図の理解度テストを行った。また，つまずき箇所と情意的特性との関連を調べるために，プログラミングに対する意識についての質問紙調査を同時に行った。なお，このテストおよび調査の対象は1997年度2年次生情報科学コース103名である。

(1) 流れ図テスト

A.W. Munzert[11]によって作成されたプログラマー適性テストの中から流れ図問題を4つ選択している。このテストは，場面と目的および流れ図が書かれている。流れ図には空白部分があり，その空白部分に入るべき適切な処理内容を用意された選択肢の中から選ぶ形式で，全部で25問である。

場面1　レコード店でのレコードの在庫枚数と種別を整理して印刷する（10問）

場面2　工場主が2店に売ったラジオと3種類の付属品の出荷台数を管理する（5問）

場面3　箱の中の木ネジと金属ネジの種類別（それぞれ3種類ある）に集計する（5問）

場面4　問屋の品物の注文受けと在庫を管理する（5問）

(2) プログラミング意識調査

全部で20問用意し，あてはまる度合いを4段階（4：あてはまる，3：や

やあてはまる，2：あまりあてはまらない，1：あてはまらない）で回答させている。質問項目の内容については表1に示している。

Ⅴ．結果と考察

1．流れ図テストについて

全部で25問あり，全体の平均点は22.7点である。得点別度数分布を図1に示す。全体の約27%が全問正解している。15問しか正解しなかった学生は，時間切れで最後の問題に手をつけていない。平均点に達していない学生は全体の約3割である。

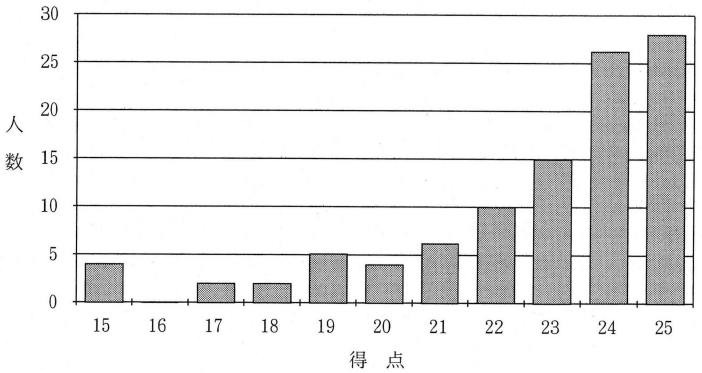

図1　流れ図テストの得点分布

(1) 誤答についての分析

問題別に誤答者数を調べた結果を図2に示す。誤答者が多かった問題は問題2，10，13，14，15，20，22，25である。それらの問題は次のような内容である。

問題 2　　条件判断の理解
問題 10　　問題の要求内容の理解
問題 13　　条件判断の理解
問題 14　　問題の要求内容の理解

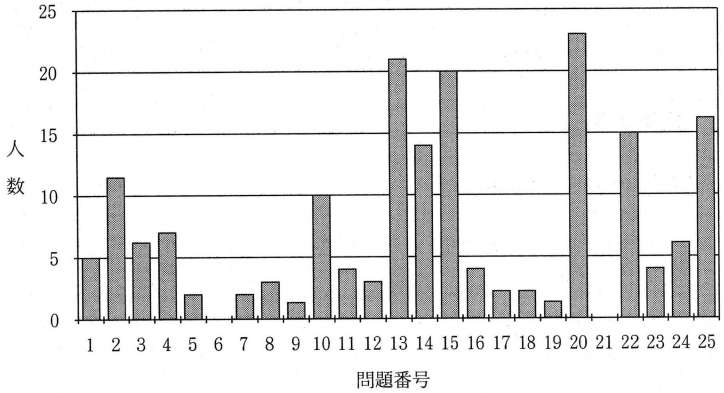

図2　問題別誤答者数

問題 15　　処理の流れの組立て（繰り返し）
問題 20　　処理の流れの組立て（繰り返し）
問題 22　　処理の流れの組立て（問題の要求内容）
問題 25　　処理の流れの組立て（問題の要求内容）

誤答者が多かった問題間での関連性について調べてみた結果，問題の要求内容を理解して処理の流れを組立てる問題 22 と問題 25 を間違った学生がやや多いが，誤答問題の間にはあまり関連性はない。

(2)　テスト得点によるグループ間での差異

テスト得点が平均点以上（23 点以上）の学生を高得点グループ（70 名），平均点に満たない学生を低得点グループ（33 名）とし，両グループ間で誤答問題に違いがあるかを調べた。その結果を図 3 に示しているが，次のような傾向が見られた。

高得点グループ：問題 20 の誤答者がやや多いが，誤答問題はばらついている
低得点グループ：問題 2，13，15，20，22，25 の誤答者が多い

この違いは，低得点グループに属する学生は問題の要求内容を理解していないことから単純な誤りをする，またそのために処理の流れを理解できなくなっている，と思われる。

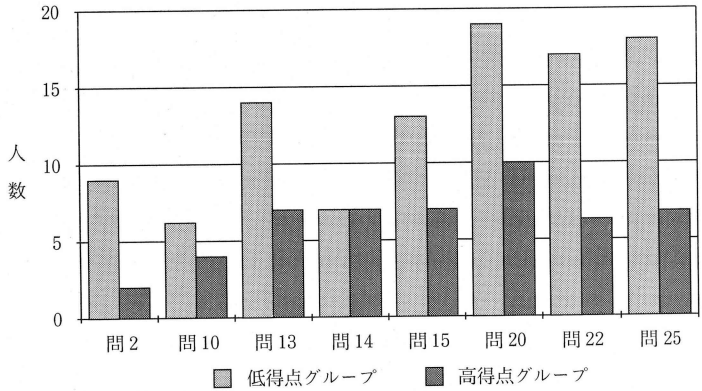

図3　高・低得点グループによる誤答の比較

2. プログラミングに対する意識調査について

プログラミングに対する意識調査の結果は表1の通りである。全体の傾向として，プログラミングはむずかしくあまり好きではないが，プログラムを作り終えたときは充実感を感じられるので，じっくり考えて納得のゆくプログラムを作りたいし，自由に作れるようになりたいと思っている。また，プログラミングに対しては明るいイメージよりはどちらかといえば暗いイメージを持っている。

3. テスト得点と意識調査との関連

高得点グループと低得点グループの間でプログラミングに対する意識に差異があるかを分析した。その結果を図4に示しているが，次のような違いが見られた。

高得点グループは，

3　段階を追って考えるのがすき

6　数学的考え方が好きである

低得点グループは，

14　時間がかかりすぎてうんざりする

10　例題のプログラムが理解できない

表1　プログラミング意識調査の結果

	調査項目	回答 1	2	3	4	平均	標準偏差
1	プログラミングがすきである	26	47	24	6	2.1	0.84
2	プログラミングは自分で自由に考えることができるからすきである	23	60	18	2	2.0	0.69
3	一つ一つ段階を追って考えることがすきである	9	39	44	11	2.6	0.80
4	プログラミングはコンピュータを使っているという実感がもてる	4	24	42	33	3.0	0.84
5	プログラミングができないと劣等感を感じる	8	17	48	30	3.0	0.88
6	数学的考え方がすきである	21	42	27	13	2.3	0.94
7	プログラムを作り終えるとやり遂げたという充実感がある	3	9	38	53	3.4	0.76
8	課題についてどこから考えていいかわからない	2	19	45	37	3.1	0.78
9	課題の意味が分かればコンピュータの処理の流れ（流れ図）に置き換えられる	17	48	28	10	2.3	0.86
10	例題のプログラムが理解できない	14	53	30	6	2.3	0.77
11	例題のプログラムは理解できるが，課題のプログラムは作れない	5	19	54	25	3.0	0.79
12	プログラミング中に発生するエラーの意味が分かる	17	50	25	11	2.3	0.87
13	プログラミング中にコンピュータの操作がわからない	12	37	41	13	2.5	0.86
14	プログラミングは時間がかかりすぎてうんざりする	9	37	45	12	2.6	0.81
15	プログラミングは暗いイメージがある	34	31	33	5	2.1	0.91
16	プログラムを作成するためにはもっと演習時間がほしい	4	16	40	43	3.2	0.83
17	プログラムを自由に作れるようになりたい	4	4	17	78	3.6	0.74
18	じっくり考えて納得のいくプログラムを作りたい	2	14	32	55	3.4	0.79
19	プログラミング演習はむずかしい	0	5	32	66	3.6	0.58
20	プログラミング演習は必要である	1	6	40	56	3.5	0.65

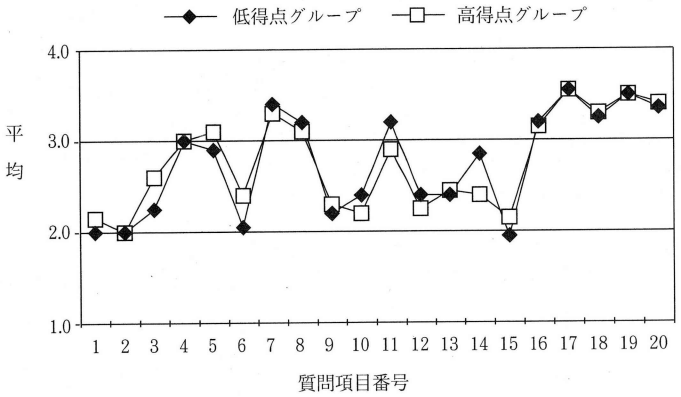

図4 高・低得点グループによるプログラミング意識の比較

11 例題が理解できても，課題のプログラムは作れない

が高くなっていて，項目6および項目14については5%水準で有意差が認められた。

上記以外の項目については，どちらのグループも値にほとんど差がない。

VI. 結　論

本研究では，プログラミング基礎演習の授業開始時に流れ図テストとプログラミング意識調査を行ない，それらをもとに抽象化プロセス段階でのつまずきに関わる要因を分析した。

その結果，つまずく箇所として，

① 与えられた問題の要求内容を理解できない

② 問題の解決手順を論理的に構成していくことができない

が抽出され，その要因として問題理解力，論理的思考力，応用力の不足が得られた。抽象化プロセスでつまずいた学生は，それらの力不足のために段階的に考えることや数学的考え方に対して嫌悪感を持っており，問題への取組み意欲も減少し，問題の要求内容を理解できないことにつながっている，と考えられる。

Ⅶ. おわりに

プログラミング学習における抽象化プロセスでのつまずきに関わる要因として抽出された問題理解力，論理的思考力，応用力，集中力はプログラミング学習に限定したものではなく，一般的な問題解決においても必要な能力である。特に，「考える力をつける」ことは情報社会においては重要である。自分の考え方や価値観が明確になっていないと，情報の選択，判断がむずかしくなり，したがって情報を発信することもむずかしくなる。

抽象化プロセス段階につまずきがある学生も「プログラミングはむずかしい」と思いながらも，「プログラムを完成させたときの充実感を感じ」，「プログラムを自由に作れるようになりたい」と思っている。これらの学生に対しては，問題解決の方法や手順などを示しながら，段階的に問題理解の過程をくり返すことによって，問題解決能力を高めていくことが必要である。

〈参考文献〉

（1）（社）情報処理学会（1993）：大学等における一般情報処理教育の在り方に関する調査研究（文部省委託調査研究）

（2）芦葉浪久（1994）：情報教育の論点，教育情報研究，10(1)，pp. 5-18

（3）安達一寿，他（1994）：プログラミング学習における学生のつまずき箇所の分析，教育情報研究，10(4), pp. 11-20

（4）I. Sommerville (1993): Software engineering 4th ed., Addison-Wesley Publishing Company

（5）Gerald M. Weinberg (1988): The Psychology of Computer Programming, Van Nostrand Reinhold A Division of Wadsworth, Inc.

（6）高本明美，他（1994）：誤り分析にもとづくプログラミング学習の支援，電子情報通信学会技術研究報告，ET94-32, pp. 23-30

（7）藤井美和子，他（1995）：プログラミング学習支援を目的とした誤りのクラスター分析，日本教育工学会第11回大会講演論文集，pp. 477-478

（8）新開純子，他（1994）：問題解決のプロセスを重視したプログラミング教育の実践，電子情報通信学会技術報告，ET94-55, pp. 77-84

（9）中尾茂子，他（1994）：プログラミング学習に対するイメージとコンピュータ親和度との関連，教育情報研究，10(4), pp. 3-10

（10）中尾茂子，他（1995）：プログラミング学習による学生のイメージ変化と学習効果の分析，教育情報研究，11(1), pp. 3-11

（11）Alfred W. Munzert (1993): Test Your Computer I.Q., Hemisphere

Publications, Inc.
(12) 永野和男 (1995)：これからの情報教育，高陵社書店
(13) 谷口るり子，他 (1997)：Visual Basic によるプログラミング教育の実践，平成9年度第11回私情協大会資料，pp. 151-152
(14) 宮田 仁，他 (1996)：プログラミングの指導方法と問題解決能力育成との関連 (2)，日本教育工学会第12回全国大会講演論文集，pp. 21-22
(15) 市川忠男，他 (1994)：かわりゆくプログラミング，共立出版
(16) 市川伸一 (1996)：学習と教育の心理学，岩波書店
(17) 中村喜宏 (1997)：C言語教材における演習問題の難易度評価，情報処理学会第55回全国大会講演論文集 (4)，pp. 4-533
(18) 大岩 元 (1996)：高校における「情報」としてのプログラミング教育，情報処理学会研究報告，96-CE-40，pp. 53-60
(19) 野口靖夫 (1997)：考える技術，創元社

(1998年9月16日受理)

An Analysis of Tripping in Programming Learning by Flow-chart Tests

Shigeko Nakao

Abstract

The purpose of this paper is to discuss the factors which cause tripping in Programming learning. In order to analyze the factors the author examined the students in my class on their ability of understanding by a flow chart and investigated how they feel about programming by the questionnaires.

The result shows that the students who trip in the process of programming could not understand what is demanded in the problem and not construct a logical procedure of solving the problem. The author points out that these students do not have enough general abilities of understanding, logical thinking and application. In order to improve the ability of solving problems, these students should repeat the process of understanding the problems step by step and we should support them in the process.