

アルゴリズム・論理的思考学習の実践と成果分析

- 単元前後の生徒アンケート結果を元に -

中園 長新*1

Email: zono@slis.tsukuba.ac.jp

*1: 筑波大学大学院 図書館情報メディア研究科

抄録

本研究では、筆者が高等学校教科「情報」で行ったアルゴリズム・論理的思考学習の実践を通して生徒がどのように変化したかについて、授業アンケートの分析を通して明らかにした。実践ではプログラミングのツールとして「アルゴリズム」と「Scratch」を用いた。実践前の生徒はアルゴリズムやプログラミングに対して慣れ親しんでいなかったが、実践の結果、アルゴリズムや論理的思考の重要性や、プログラミングがどのようなものであるかを理解した生徒が多く見られた。しかし、概念理解に関しては一様ではなく、「アルゴリズム」については実践後にある程度理解が定着したのに対して、「論理的思考」については十分な理解が得られていない可能性が示唆された。また、ツールを使ったプログラミング実習は生徒に楽しいものとして受け入れられた反面、プログラミング特有の難しさを感じた生徒も多く見受けられた。本研究により、高等学校教科「情報」にプログラミングを取り入れることで、アルゴリズム・論理的思考学習が実践されること、および、その際に予想される生徒の変化について整理することができた。今後は本研究の知見を元に、様々な学校・生徒を対象に実践を積み重ね、より効果的な実践方法やツールの活用を研究していく必要があると考えられる。

◎Key Words 教科「情報」、アルゴリズム、論理的思考、アルゴリズム、Scratch

1. はじめに

1.1 研究の背景

コンピュータを活用した問題解決学習の必要性は以前から言及されており、1996年にはすでに大岩による提言¹⁾がみられる。この論文で大岩はプログラミング教育を取り上げ、その理念と内容に関する提案を行っている。さらに近年では、経済協力開発機構(OECD)による生徒の学習到達度調査(PISA)¹⁾からの影響も大きい。OECD PISAは、読解力、数学的リテラシー、科学的リテラシーの3分野で生徒の到達度を調査するものであり、3年おきに調査が実施されている。調査結果はニュース等でも大きく報道され、我が国における教育のあり方について考えるきっかけとなっている。中でも、2009年調査でメインテーマとして設定された読解力すなわち「自らの目標を達成し、自らの知識と可能性を発達させ、効果的に社会に参加するために、書かれたテキストを理解し、利用し、熟考し、これに取り組む能力」²⁾については、我が国の子どもに不足している能力であることが調査結果から示唆されており、2008～2009年に告示されたいわゆる新学習指導要領においても、読解力の育成が大きなテーマのひとつとして設定された。読解力はただ単にテキストを読み取るだけでなく、読み取ったことを元に理解したり考えたりすることを含む幅広い概念である。これは、着眼点は異なっているものの、問題解決学習と同じような問題意識や目標を設定していると捉えることもできるだろう。

このような流れの中で、高等学校の普通教科²⁾「情報」においても、コンピュータを活用した問題解決を扱うことが学習指導要領において示されている。これは、1999(平成11)年告示の現行学習指導要領では主に「情報B」の「問題のモデル化とコンピュータを活用した解決」において、2009(平成21)年告示の新学習指導要領では主に「情報の科学」の「問題解決とコンピュータの活用」で扱うこととされている³⁾⁴⁾。

学習指導要領解説によると、コンピュータを活用した問題解決では「アルゴリズム」や「処理手順の自動実行」等の概念を扱うこととされている。また、これらと関連性が深い概念として「論理的思考」がある。本稿では、これらの習得を目指す学習を総称して「アルゴリズム・論理的思考学習」と呼ぶことにする。これらを直接的に体験できる実習としてプログラミングがあり、多くの高等学校において実践に取り入れられている。しかしながら、プログラミングに本格的に取り組むためには、前述のアルゴリズムや処理の自動実行といった概念の理解にはじまり、利用するプログラミング言語の文法の理解やデバッグ等、様々な要素について学習する必要がある。これらを必修2単位の高等学校教科「情報」で扱うためには、教師の知識・技術や授業時間の面において大きな負担を強いる可能性がある。また、必修科目として全生徒が受講するという点を踏まえると、プログラミングの専門的な部

¹⁾ OECD PISA. <http://www.pisa.oecd.org/> (2012-02-01 確認)

²⁾ 1999(平成11)年告示の現行学習指導要領までは「普通教育に関する各教科(普通教科)」であったが、2009(平成21)年告示の新学習指導要領では「各学科に共通する各教科(共通教科)」となった。本稿ではこれらを区別せず用いる。

分をどの程度指導することが望ましいか、その裁量は簡単ではない。

それでは、プログラミングを適切に活用して問題解決の学習をするためにはどのような実践が考えられるだろうか。本研究ではこの間に対する回答のひとつとして、教員・生徒双方に負担の少ないプログラミング学習を設計し、その実践を通してアルゴリズム・論理的思考学習を行うとともに、その成果を評価した。

1.2 研究の目的と意義

本研究では、筆者が高等学校教科「情報」で行ったアルゴリズム・論理的思考学習の実践を通して生徒がどのように変化したかについて、授業アンケートの分析を通して明らかにする。本研究によって、アルゴリズム・論理的思考学習における教材の効果を確認することができ、高等学校教科「情報」において、コンピュータを用いた問題解決の有意義な実践が促進されることに寄与できる。

なお、教科「情報」等の学習用教材としてプログラミングを活用した事例はこれまでも数多く報告されており、実践の具体的な内容を詳細に報告したものも多い。それらに対して、本稿は研究論文としての位置付けであるため、実践の詳細な記述はあえて行わず、その結果として生徒がどのように変化したかを把握するためのデータ分析に主眼を置く。この観点により、本稿は実践の成果を客観的に分析する研究としての立場をとる。

2. 授業実践

2.1 実践と実践校の概要

本研究の実践は、2010年10月から2011年1月にかけて、X県立A高等学校2年生5クラス（学年在籍生徒数198名）を対象として行った。実践校は卒業生の大学等進学割合が多く、地域住民から「進学校」として位置付けられている。教科「情報」については必修科目として「情報C」2単位を設置しており、1年次と2年次で1単位ずつ分割履修している。教員は2名（非常勤講師である本稿筆者と常勤教諭）でティームティーチングを実施しており、この実践では本稿筆者が主として指導を行った。

実践校の情報Cでは、学習指導要領や教科書を元に、生徒の実態に合わせて学習内容を再構成することで単元を設定している。単元ごとに教員がワークシートを準備し、教科書や参考資料、Web上の情報源などを組み合わせて活用する授業を展開している。各単元では生徒が取り組む実習課題を設定し、課題作品を完成させる過程で学習内容の定着を図っている。また、講義と実習の授業形態をシームレスに接続するため、ほとんどの授業をコンピュータ室で実施している。

2.2 情報Cにおけるアルゴリズム・論理的思考学習の意義

本実践は、情報Cの単元「アルゴリズムと論理的思考」として設定した。「情報社会に参画する態度」を主として育成する科目である情報Cの学習指導要領には、

本来このような項目は存在せず、コミュニケーション力の育成などを主とする実践が多い。しかし、人間同士の交流に加えて様々な機械（たとえばコンピュータや携帯電話）とふれあうことが求められる現代社会においては「コンピュータと人間のコミュニケーション」についても社会を生きる力の重要な要素となり得る。さらに、コミュニケーションのためには相手に自分の考えなどを的確に伝達する必要があるが、そのために必要な能力のひとつとして論理的思考力を位置付けることができる。このような理由から、アルゴリズムや論理的思考の育成も広い意味で「情報C」の学習内容の範疇であるとみなし、「情報社会に参画する態度」として重要視されているコミュニケーション能力の育成に関連づけて単元を設定した。

また、これまでのプログラミング学習は、そのほとんどが情報の科学的な理解を目標としたものであった（たとえば、新開・大森 2004³、奥村 2007⁴等）。本研究ではプログラミング教育について、情報科学以外を主眼とした活用を目指しており、この点において先行する多くの実践や研究と一線を画すものである。

2.3 実習に利用したツール

本実践の実習では、第1時に「アルゴロジック」（図1）、第2時以降に「Scratch（スクラッチ）」（図2）という2つのツールを教材として活用した。これらについて簡単に紹介する。

「アルゴロジック」は電子情報技術産業協会（JEITA）が開発した、プログラミングの基本となる論理的思考（アルゴリズム）をゲーム感覚で習得するための課題解決型ゲームソフトである³。Adobe Flash形式で作成されてWeb上に無償公開されているため、Flashに対応したWebブラウザがあればすぐに利用できる。本稿執筆時は初心者問題（Jr.）とチャレンジ問題の2種類から構成されるVer.1.5や、「順次」「繰り返し」「分岐」を実現した「アルゴロジック2」が公開されているが、実践時はVer.1.5のチャレンジ問題にあたる部分のみが公開されていた。ステージごとに課題が設定されており、命令ブロックを組み合わせるとその命令通りにロボットが動き、課題を達成するゲームである。マウス操作だけで直感的に操作できるなど、子どもにも親しみやすいデザインを目指して開発されている。

「Scratch」はマサチューセッツ工科大学メディアラボ等が共同開発した、子どもでも簡単にプログラミングを行うためのプログラミング言語である⁴。アルゴロジックと同様に命令ブロックを組み合わせるプログラムを記述し、そのプログラムに沿って画面上の猫イラストが動く。Windows、Mac OS、Linuxのそれぞれで動くアプリケーションが無償提供されており、コンピュータにインストールすることで利用可能である。

³ アルゴロジックは以下のサイトに公開されている。
<http://home.jeita.or.jp/is/highschool/algo/>（2012-02-01 確認）

⁴ 本実践ではScratch 1.4を利用した。Scratchは以下のサイトから入手できる。
<http://scratch.mit.edu/>（2012-02-01 確認）

これらのツールはいずれも、子どもにも親しみやすいイラスト等を使ったインタフェースを採用しており、ほとんどの操作がマウスを使って実施できる。そのため、一般的なプログラミング言語にある「文法を覚える」というステップを省略することができると同時に、プログラミングは難しいという先入観を克服することができると考え、実践で利用することにした。なお、実践校の生徒用コンピュータはアルゴジックを実行できる Web ブラウザが搭載されており、Scratch はコンピュータ室の管理者に依頼しインストールしてもらうことで学習環境を整えた。

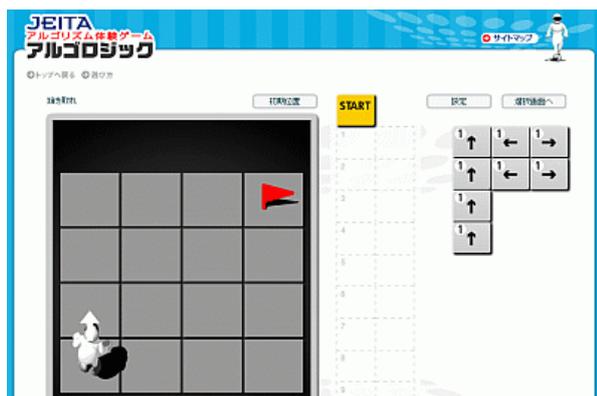


図1 アルゴジック



図2 Scratch

2.4 単元の学習目標と基本方針

本単元の学習目標は、学習指導要領の記述や実践校生徒の実態に基づき、「ものごとを進めるには順序や論理的な思考、アルゴリズムなどが必要であることを、プログラミング実習を通して体験的に理解する」とした。すなわち、本単元においてプログラミング実習は目的ではなく手段である。生徒がプログラミング学習に終始するのではなく、それを手段とした論理的思考に重点を置くような単元構成・教材設計が重要であり、実践においてもこの点を特に留意した。

2.5 単元の概要

本実践の単元「アルゴリズムと論理的思考」は50分授業6回(週1回、6週間)で構成される。各時の学

習内容は表1の通りである。

なお、本単元では実践の成果分析のため、単元開始時(初回)と終了時(最終)にそれぞれアンケートを実施している。これらについては3章で詳説する。

表1 単元の学習内容

時数	内容
第1時	アルゴジックでアルゴリズムの基礎を学ぶ
第2時	Scratch の紹介とプログラミング練習
第3時	プログラミングの準備
第4時	プログラミング実習1
第5時	プログラミング実習2
第6時	課題作品の提出

(1) 第1時: アルゴジックでアルゴリズムの基礎を学ぶ

第1時では単元前の初回アンケートを実施した後、適切な手順を踏まなければものが実行できないこと、同じことを実行する場合でも効率的な方法があることについて具体例を交えて説明した。これらを元に「アルゴリズム」の定義を教え、その実習として「アルゴジック」を体験させた。

アルゴジックを利用して、指示された課題(旗を全て取る、指示された経路をトレースする)を達成するように、命令ブロックを配置し、画面上のロボットを動かすゲーム感覚の実習を行った。実習に取り組むスピードは個人によって異なるため、生徒一人一人が自分のペースで課題に取り組むようにした。課題はアルゴジック内に用意されたものを、各自で順を追って取り組むよう指導した。

(2) 第2時: Scratch の紹介とプログラミング練習

第2時では、前時の復習を元に「論理的思考」の定義を教え、プログラミングだけでなく日常生活においてもこの概念が重要であることを確認した。そして「Scratch」を紹介し、実際に操作しながら基本的なプログラミング方法を学習させた。また、ツールの利用方法とあわせて、プログラミングの3つの制御構造(プログラミングの基本的な概念)といわれる「順次処理」「繰り返し処理」「分岐処理」や、変数の使い方についての解説も行った。

Scratch の言語自体は複雑なプログラムを作成することもできるが、本実践では猫イラストを動かして絵

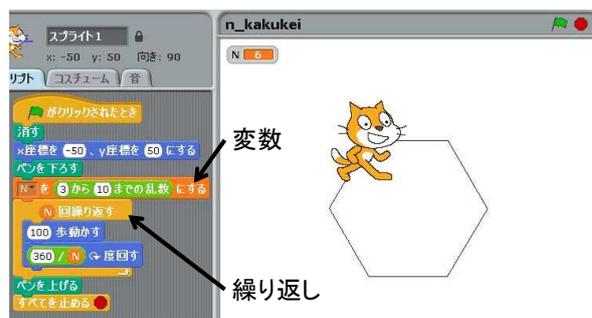


図3 正N角形を描くプログラムと実行例

を描くという基本的な機能のみを利用した。練習用課題として作成した、Scratch で正 N 角形を描くプログラムおよび実行例 (N=6 の場合) を図 3 に示す。この課題では主に「順次処理」「繰り返し処理」や変数についての理解を深めることを目指した。

(3) 第 3 時：プログラミングの準備

第 3 時はプログラミングの事前準備として、Scratch で描画する図形のラフスケッチと、その図形を描画するための簡単な手順を書かせた。

この実習はコンピュータを使わず、ワークシート上に手書きさせた。ラフスケッチは具体的に描画したい図形を描かせ、手順についてはフローチャート風の形式を紹介して手続き順序がわかるような記述をするように指導した。プログラミング実習前に手書きでラフスケッチと手順を書かせることで、自分が描きたい図形がどのようなものであり、その描画を実現するにはどのような命令をどのような順序で実行すればよいのか、生徒自身がアルゴリズムを論理的に思考することを目指したものである。生徒が記入したワークシートの例 (一部) を図 4 に示す。この生徒は順次処理を基本として、一部処理に繰り返しを用いるなどの工夫をしている。

(4) 第 4~6 時：プログラミング実習

第 4 時と第 5 時は、前時で作成したラフスケッチと手順書を元に、実際に Scratch でプログラムを作成させた。最後の第 6 時で完成した課題作品を提出させ、単元後の最終アンケートを実施した。

Scratch による実習は、第 2 時で指導した基本的なプログラミング方法を利用して実現可能なものを想定したが、生徒の要望によって発展的な機能を使うことも可とした。機能の確認等には公式サイトのリファレンスガイド、教員によるアドバイス、そして生徒同士の

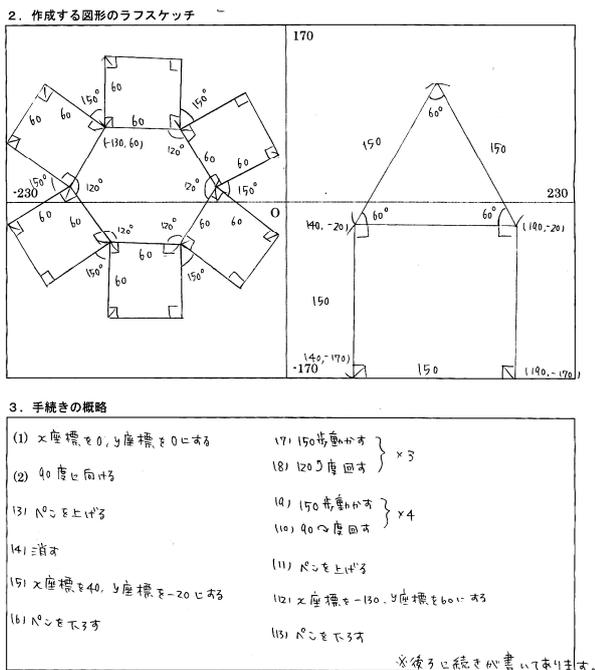


図 4 ワークシートの例 (一部)

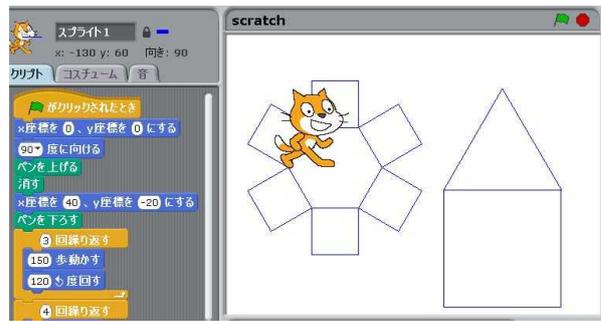


図 5 Scratch を用いた生徒作品例

教え合いや学び合いを活用した。生徒 (図 4 と同一) の作品例を図 5 に示す。ワークシートで考えた手順が Scratch 上でプログラミングとして設計され、ラフスケッチ通りの結果が得られている。

このプログラミング実習では、第 3 時に手書きで作成した手順が、ラフスケッチの通りに実行できるかどうかは鍵となる。頭で考えた手順が Scratch 上で意図したとおりに動作するか、動作しない場合はどの部分に問題があったかを考え、試行錯誤を繰り返すことで、自分のアルゴリズムが論理的かどうか、生徒自身に思考させることを目的とした。

2.6 実習における生徒の様子

本節では、教員として実践を行った筆者の立場から、実習中の生徒の様子について概説する。

実践校の情報Cでは、普段からコンピュータを用いた実習を多く取り入れている。そのため、アルゴリズムや Scratch を使う実習に対して、生徒は抵抗なく自然に取り組んでいた。いずれのツールも見た目やキャラクターに工夫が施されており、親しみを持って楽しみながら利用している生徒がほとんどであった。一部の生徒においては教員の説明中にも実習を続けるといった行動も見受けられた。こうした態度は授業態度として褒められるものではないものの、ツールが生徒に受け入れられた証とみなすことができるだろう。

一方で、ツールの使いこなしに関しては一部に個人差が見られた。アルゴリズムについては提示された少数のブロックを組み合わせるといった単純さのため、ほとんどの生徒が使い方を適切に把握できていた。しかし Scratch は様々な機能が搭載されたツールであるため、使いこなしに戸惑う生徒も見受けられた。こうした個人差を吸収するため、実習時は教員が机間巡視しながら適宜アドバイスするとともに、Scratch 公式サイトで公開されているリファレンスガイドを印刷して生徒が自主的に参照できる環境を整えた。生徒はこうした支援を活用することに加えて、生徒同士による教え合いや学び合いをしながら実習に取り組んでいた。

3. 授業アンケートの実施と分析

3.1 アンケートの実施方法

生徒対象のアンケートは、第 1 時 (初回授業) の冒頭と第 6 時 (最終授業) の最後に実施した。これらをそれぞれ初回アンケート、最終アンケートと呼

ぶ。アンケートはX県が設置した教育情報ネットワークシステムのアンケート機能を使ってフォームを作成し、コンピュータ画面上で生徒に回答させた⁵。調査対象は本単元を学習した生徒全員であるが、欠席等で回答できなかった生徒もいた。本研究は実践による生徒の変化を知ることが目的としているため、初回アンケートと最終アンケートの両方を提出した177名（学年在籍生徒数の89.4%）のデータを有効回答とみなし、分析を行うこととした。

3.2 授業前の生徒の実態

まず、初回アンケートにおいて学習前の生徒の実態を調査した。具体的な質問項目としては、アルゴリズムという語を知っているか、プログラム経験の有無等である。

アルゴリズムという語については、「意味も知っている」が1名、「言葉は知っている」が116名、「知らない」が60名であった。また、プログラミングの経験について「ある」と回答したのは11名で、「ない」と回答したのが166名であった。これらの結果から、ほとんどの生徒はアルゴリズムやプログラミングに触れておらず、言葉を聞いたことがあっても意味を知っているわけではないという実態が明らかになった。

3.3 実習ツールに対する生徒の意識

次に、最終アンケートにおいて、実習に使ったツールに対する意識を質問した結果を分析する。ここでは、アルゴロジックとScratchの楽しさ、難しさ、操作のわかりやすさについてそれぞれ6件法で質問した。それぞれの回答を集計したものを図6に示す。

ツールの楽しさについては、両ツールとも半数以上の生徒が「とても楽しかった」または「楽しかった」（アルゴロジック107名、Scratch89名）と回答しており、親しみを持ってツールを活用したことがうかがえる。一方で、ツールの難しさについては6割以上の生徒が「とても難しかった」または「難しかった」（アルゴロジック110名、Scratch140名）と回答しており、普段親しんでいないプログラミング関連の実習は生徒にとって困難を伴うものであったことがわかる。ツールのわかりやすさについては双方ともに6割以上（アルゴロジック120名、Scratch107名）の生徒が肯定的（「とてもわかりやすかった」「わかりやすかった」「ややわかりやすかった」の合計）であることから、実習においては「ツール自体はわかりやすいが、それを用いて行う内容が難しかった」と考えられる。

アルゴロジックとScratchのツール間比較を行うと、いずれの質問項目に関してもScratchのほうがネガティブな評価になっていることがわかる。また、授業中の生徒の様子では、アルゴロジック実習ではほとんど聞かれなかった「自分は（課題として）何をすればい

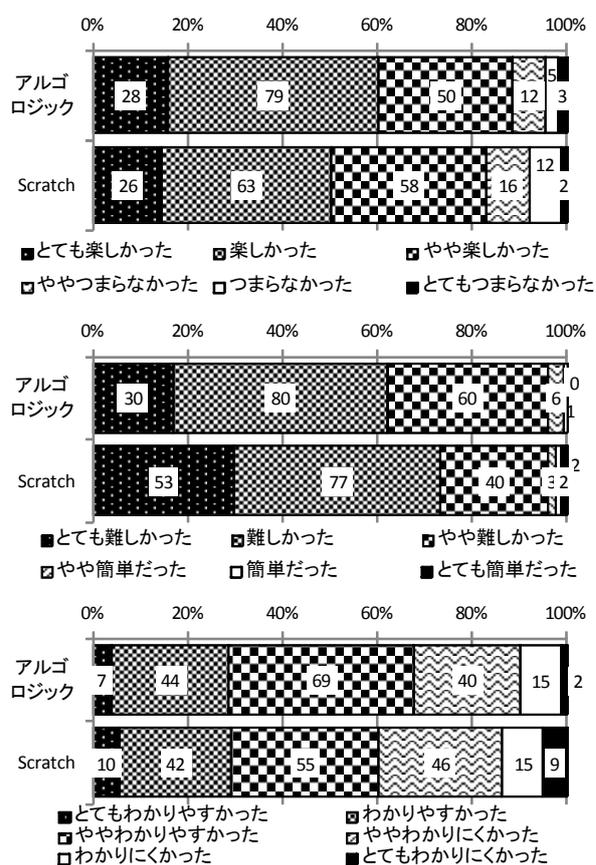


図6 ツールの楽しさ、難しさ、わかりやすさ

いか」という趣旨の質問がScratch実習時に散見された。これは、アルゴロジックが「与えられた課題に対する回答を見つける」ものであるのに対してScratchが「自分が作りたいものを実現するプログラムを作る」ものであり、課題の自由度が生徒に困難さを感じさせ、ネガティブな評価につながったと考えられる。

3.4 アルゴリズムに対する実践前後の生徒の変化

アンケートではさらに、本実践を通して、アルゴリズムに対する生徒の意識がどのように変化したかを調査した。調査は、初回アンケートと最終アンケートの回答傾向を比較することで行った。

初回アンケートと最終アンケートでは、「アルゴリズム」という語について思い浮かべることができることを自由記述によって得た。本調査では、その回答結果をテキストマイニング用ソフトウェアである「KH Coder」⁶に入力し、共起ネットワーク図を作成、比較した。

テキストマイニングの手順は次の通りである。まず、生徒のアンケート回答集計表から、分析を行う設問の回答のみをテキスト形式で抽出する。このデータをKH Coderに入力し、描画する共起関係（edge）数を150として共起ネットワーク図を描画した。共起ネットワーク図中の円は単語を表し、円の大きさはその単語の出現数に比例している。円同士をつなぐ線は、出現パ

⁵ 生徒一人一人に個別のIDが付与されているため、アンケートは記名式となる。回答に際しては、アンケート回答内容と成績は一切関係しないこと、結果を研究で利用する場合は統計処理を行ったりして個人が特定できないようにすることを申し添えた。

⁶ 本研究ではKH Coder Ver. 2.beta.25aを利用した。KH Coderは以下のサイトから入手できる。http://khc.sourceforge.net/ (2012-02-01 確認)

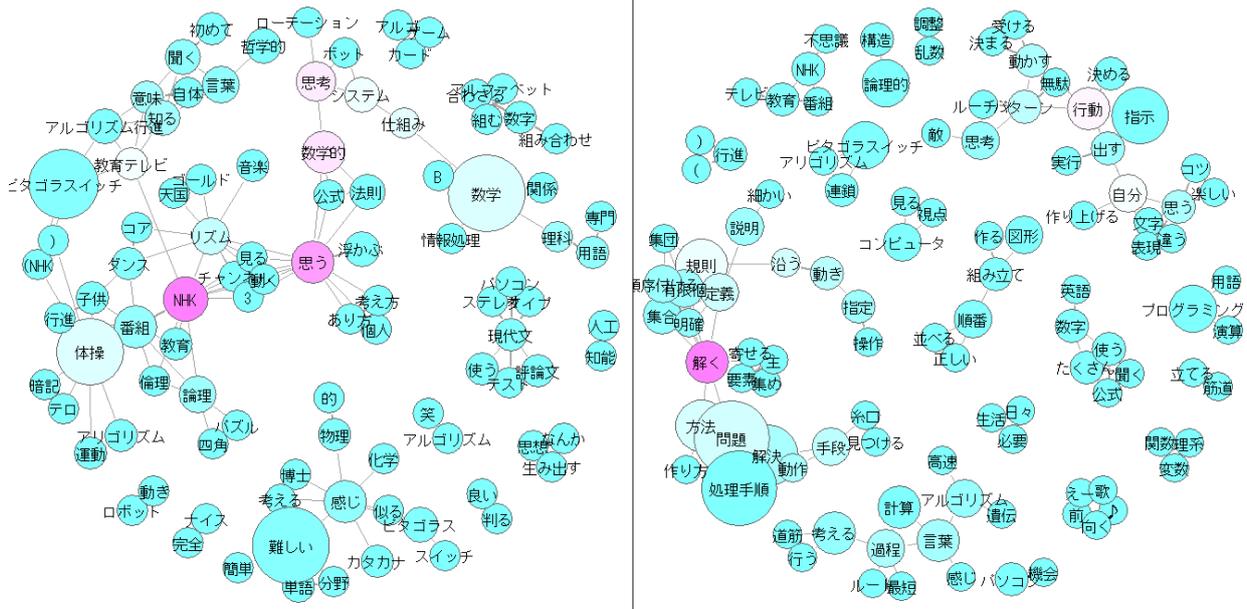


図7 アルゴリズムについての共起ネットワーク (左: 初回, 右: 最終)

ターンの似通った (共起関係にある) 単語を結んでいるが、紙面での見栄えを考慮して太さは一定とした。このようにして描画した「アルゴリズム」に関するイメージの共起ネットワーク図を図7に示す。

まず、「アルゴリズム」に関する初回アンケート結果 (図7左) を分析する。初回アンケート結果では、共起ネットワーク図の左側に NHK の子ども向け番組コンテンツである「アルゴリズムたいそう」や「アルゴリズムこうしん」⁷に関連した語句が多く見られる。図の右上には「数学」という単語を中心としたクラスターが見られる。さらに、図の中央付近および下側には、「思う」「難しい」等の感覚的な単語が多く見られる。これらの結果から、単元前の生徒は「アルゴリズム」という語について、テレビで見た知識や曖昧なイメージを中心に捉えていることが確認できた。

一方、最終アンケート結果 (図7右) では、図の左下に「処理手順」「問題解決」「手段」など、アルゴリズムの本質に関わる単語によるクラスターが観察された。図の右上には「指示」「行動」等の単語も見られる。初回アンケートで大きめのクラスターを形成していたテレビ番組関連の単語については、最終アンケートでも一定数登場しているものの、ネットワーク図全体に占める割合は低くなっている。これらの結果から、単元後の生徒は「アルゴリズム」という語について、語の本質に迫る理解をしていると考えられる。

以上の結果から、生徒は本単元の学習および実習を通して、アルゴリズムに対する理解を深められたことが確認できた。

3.5 論理的思考に対する生徒の意識

アルゴリズムに続き、「論理的思考」という語に対す

る生徒の意識を調査した。質問形式は「アルゴリズム」と同様であるが、「論理的思考」についての意識を問う質問は最終アンケートにのみ設置した。これは、本実践の前年度および前々年度に行った同様の実践⁸において、単元実施前の生徒は「論理的思考」という単語に慣れ親しんでおらず、初回アンケートでは十分なデータが収集できない可能性が確認されていたためである。「論理的思考」に関するイメージの共起ネットワーク図を図8に示す。分析方法は前節と同様である。

論理的思考についての共起ネットワーク図で特徴的なのは、単語同士のつながりが比較的に少ないことである。左上付近に「相手」「伝える」「思考方法」等の単語によるクラスターが確認できるが、各単語の出現頻度は多くない。単元終了後の最終アンケートでこのような結果が得られたということは、生徒の多くが「論理的思考」について十分な理解が得られなかった可能性を示唆している。これらのことから、「アルゴリズム」と比較して「論理的思考」は生徒にとって理解が難しい概念であることが示唆された。

もっとも、論理的思考はアルゴリズムに比べてその概念がわかりにくいことも事実である。これらの理解度の差は、実践の在り方によるものか、概念の難易度そのものによるものなのかについては、今後更なる分析が必要である。

3.6 単元終了後の生徒の意識

最後に、単元終了後の生徒の意識について最終アンケート結果を元に調査した。調査項目は授業内容全体に関して「アルゴリズムや論理的思考について理解できたか」および「今後プログラミングをやりたい

⁷ いずれもNHK教育 (現・Eテレ) の子ども向けテレビ番組「ピタゴラスイッチ」のコンテンツである。
<http://www.nhk.or.jp/kids/program/pitagora.html> (2012-02-01 確認)

⁸ 本実践以前の成果については、2009年8月に開催された第2回全国高等学校情報教育研究大会において「プログラミング実習で論理的思考力とコミュニケーション力を高める: 情報Cの教材としてScratchを活用する試み」と題したポスター発表を行っている。

を取り入れることが望ましい場合もあり、その場合はプログラミングについて文法を含め丁寧に学習することも必要であろう。しかし、今回の実践校のようにプログラミングを強調しない場合であっても、ツールを適切に選択することで学習効果を上げることができることが確認された。プログラミングに関するツールは他にも様々なものが開発されているため、実態に即したツール選択が今後重要になると考えられる。

4. おわりに

初回アンケートの結果から、本実践で扱ったアルゴリズム・論理的思考学習は、単元前の生徒にとって比較的親しみがない概念であったと考えられる。一方で、第1章で言及したように、これからの時代を生き抜くためにはアルゴリズムや論理的思考の力が不可欠である。本実践ではこの困難な課題を授業で取扱い、生徒にどのような成果がもたらされたかを研究した。

授業における生徒の様子や初回アンケート、最終アンケートの結果等を総合すると、単元前の生徒の多くはアルゴリズムや論理的思考、そしてプログラミングに対してほとんど馴染みがない状態であり、知っている言葉も聞いたことがある程度で、その本質は理解していなかった。しかし、教員による講義とプログラミング実習を通して、アルゴリズムについて概念を理解し、プログラミングに親しむことができた。実習を通して、親しみを持ってアルゴリズム・論理的思考学習を行うことができたことが、本実践の最大の成果であるといえる。

一方で、概念の理解については一様ではなく、「アルゴリズム」については実践後にある程度理解が定着したのに対して、「論理的思考」については十分な理解が得られていない可能性が示唆された。テレビ番組などで聞き覚えのある「アルゴリズム」に対し、聞き覚えのない「論理的思考」に対する親近感がなかったのではないかと考えられる。あるいは、「アルゴリズム」に比べて「論理的思考」が概念として難解であることも影響している可能性がある。この問題点については、問題解決を扱った市販のわかりやすい文献^⑧を活用する等、教育方法を研究する必要がある。

また、プログラミングに対する難しさも今後の検討課題である。本実践ではアルゴリズムと Scratch を活用したことにより、ツールを活用すること自体について生徒は楽しんでいった。しかし、楽しくツールを活用することと、そのツールを使って適切な作業を実施できることは別の問題である。プログラミングの概念（たとえばループ、条件分岐、変数等）を丁寧に教えたり、適切な教材を組み合わせたりする等、今後の改善が求められる。改善案としては様々な方策が考えられるが、たとえば日本語を利用したプログラミング言語（なでしこ^⑨、ドリトル^⑩等）を活用することで、生徒のプログラミングに対する敷居を下げつつ、言語的な学習も取り入れるといった実践が考えられる。また、その他の例としてアンブラグドコンピュータサイエンス^⑪の活用が考えられる。プログラミングやアルゴリズムというどうしてもコンピュータを活用しなけれ

ばならないと考えがちだが、現在は様々な教材開発が進んでおり、コンピュータを使わないアナログの教材でそれらを学習することも可能である。今後の検討課題としたい。

本研究により、高等学校教科「情報」にプログラミングを取り入れることで、アルゴリズム・論理的思考学習が実践されること、および、その際に予想される生徒の変化について整理することができた。今後は本研究の知見を元に、様々な学校・生徒を対象に実践を積み重ね、より効果的な実践方法やツールの活用を研究していく必要があると考えられる。その際、アルゴリズム・論理的思考に関する理解と数学的能力との関連について、あるいは ICT に対する意識との関連についてクロス分析等を用いて、深い分析を行っていくことが必要となるだろう。21世紀を生きる力として重要視されている問題解決力や、それに関連する読解力を育成するためにも、高等学校教科「情報」での実践事例を蓄積し、分析、改善を繰り返すことで今後ますます推進していく必要がある。

謝辞

実践とアンケートにご協力いただいた X 県立 A 高等学校 2 年生（当時）の生徒の皆様、および学校関係者各位に心より感謝いたします。

参考文献

- (1) 大岩元：“高校における教科「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：コンピュータと教育研究会報告，Vol. 96, No. 52, pp. 53-60 (1996).
- (2) 文部科学省：“OECD 生徒の学習到達度調査：2009 年調査国際結果の要約”，文部科学省 (2010).
http://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/fieldfile/2010/12/07/1284443_01.pdf (2012-02-01 確認).
- (3) 文部科学省：“高等学校学習指導要領解説：情報編（平成 12 年 3 月、平成 17 年 5 月 一部補訂）”，開隆堂 (2005).
- (4) 文部科学省：“高等学校学習指導要領解説：情報編（平成 22 年 5 月）”，開隆堂 (2010).
- (5) 新開純子，大森克史：“情報科学を意識したプログラミング教育の実践”，日本教育工学会研究報告集，Vol. 2004, No. 4, pp. 13-18 (2004).
- (6) 奥村晴彦：“情報科学教育への利用（特集：教育用プログラミング言語と授業利用）”，情報処理，Vol. 48, No. 6, pp. 598-601 (2007).
- (7) 石原正雄：“スクラッチアイデアブック：ゼロから学ぶスクラッチプログラミング”，カットシステム (2009).
- (8) 渡辺健介：“世界一やさしい問題解決の授業：自分で考え、行動する力が身につく”，ダイヤモンド社 (2007).
- (9) クジラ飛行機：“日本語プログラム言語なでしこ公式バイブル”，ソシム (2008).
- (10) 兼宗進，久野靖：“ドリトルで学ぶプログラミング：第 2 版”，イーテキスト研究所 (2011).
- (11) T. Bell, I. H. Witten & M. Fellows 著・兼宗進監訳：“コンピュータを使わない情報教育アンブラグドコンピュータサイエンス”，イーテキスト研究所 (2007).

(2011 年 12 月 1 日 受付)

(2012 年 2 月 17 日 採録)