

小学生からのプログラミング教育

大岩 元^{†1}

概要： 筆者は、プログラミング教育は中学生から始めても遅くないと考えてきた。しかし、幼稚園児も含む小学生クラスに、イスラエルで小学校高学年向けに開発されたネット教材 `codemonkey` を授業形式で使い、その導入に、英国で開発された3才児向けのプログラミング教材 `Cubetto` を経験させた所、スムーズに `codemonkey` の内容に進んで行くことができた。生徒がグループに分かれて相談しながらプログラムを作成している様子を見て、小学生からのプログラミング教育が可能であり、大きな成果が期待できることを確信した。

`codemonkey` は、「サルがバナナを取りに行く」、という具体的問題設定のもとに、タートルグラフィックスのアルゴリズム構築を `coffee script` という実用言語のコーディングを通じて行なう。作成されたコードの質も評価されるので、単に動けば良いということだけでは先に進めないように構成されていて、200ステップで、引数付関数を使えるようになる。内容が面白いために、大人でも十分に楽しんでプログラミングを学べる内容になっているだけでなく、この教材を使って `active learning` をどのように指導するかというマニュアルまで用意されているので、プログラミングの経験がない小学校教師でも、授業を行なうことができそうである。

キーワード： 幼児・小学生のプログラミング教育, CodeMonkey, Cubetto

Programming Education for Primary Schools

Hajime OHIWA^{†1}

1. はじめに

日本人の49%の仕事がAIに奪われるという予想が出されたこともあって、次期学習指導要領では、教育目標が「21世紀型能力」の育成になった[1]。従来の日本の教育は、社会に役立つ正しい知識とスキルを定着させることに目標を置き、工業化社会で大成功を収め、世界第2位の経済大国になった。しかし、このような能力だけでは21世紀の社会を成立させることが出来ない。社会生活を成り立たせる市民の能力が情報化ですっかり変わってしまったのである。

2000年代に入り、米国の情報産業を中心に、「21世紀型スキル」[2]が研究され、それが教育の目標となってきた。これを承けて文部科学省は従来の日本の教育との整合性を考慮して日本人の教育目標として「21世紀型能力」を策定したのである。その実体は、自分で考える力の育成である。

この動きは学校教育ですでに始まっていて、教えることは少なくして、考える教育を行なうことが日本の学校でも模索されてきた。しかし、大学進学者に対してはセンター試験のような大学入試が重視され、それを目指して生徒が学習を進めた結果、試験問題に正しい解答を与えることが至上命令となり、試験の成績は良いが、社会生活に必要な知識とスキルを持つ大学卒業生の数がすっかり減ってし

まった。

「21世紀型能力」を育成するために、知識とスキルを教えこむ教育から、学習者が自ら考え、仲間とともに学習を進め、教師はそれを支援するという新しい教育が始まっている。そこでは、アクティブ・ラーニングや `Project Based Learning(PBL)` など、新しい協同作業や、学習者同志の議論を学習活動の中心とする授業が行なわれるようになってきた。しかし、教育機関では学習者個人を評価する必要があり、現場の教師は協同作業の結果をどのように学習者個人の評価に反映させるのかで苦勞している。

プログラミング教育は、「21世紀型能力」を育成する教材として優れている。プログラムは何かの問題を解決するものとして作られるものであり、そこでは最低限、プログラミング言語を理解してそれでプログラムを書く能力を訓練する必要があるが、それ自体は「21世紀型能力」を形成することにはならない。習得した言語を用いて何らかの問題解決をする活動ができるようになって、初めて「21世紀型能力」が育成されたことになる。

しかし、日本でここ数十年にわたって行なわれてきたプログラミング教育では、言語を理解するレベルにとどまり、それを用いて社会的に意味のあるプログラムが作成できる能力を育成するところまで行なうことはほとんどなかった。それでも日本の社会が情報技術を使って成立しているのは、少数の情報技術者が個人の努力によって問題解決のための

^{†1} 協創型情報空間研究所
Research Institute of Collaborative and Creative Information Space

プログラミングが行なってきたからである。

2. 5歳からのプログラミング教育

先進国では、5歳からのプログラミング教育が始まった。その内容は、日本で従来行ってきた「プログラミング言語」教育とは大きく異なる。フィンランドのプログラマーが書いた「ルビィのぼうけん」の推薦文で、まつもと ゆきひろ氏は「プログラミングの本質は問題認識と問題解決です。」とした上で、「この本は子どもにもわかるやり方で、プログラミングの本質に迫ります。」という推薦理由を述べている。

日本でプログラミング教育の本が多数出版されているが、言語の解説が中心で、ゲームを作らせるといった創造活動も含まれるが、それは処方箋がほぼ与えられていて、キャラクターのデザインなどで創造性が発揮されるだけのものが多い。極め付きが「写経型学習」で、言語の理解しか目的とせず、創造的活動については、その方法論の解説が全く含まれていない。

「ルビィのぼうけん」では、プログラミングに関する重要概念の解説の後に、それが現実の問題解決に、どのように生かせるかを、課題を通して体験させている。

3. 3才児用教材 Cubetto

英国では、3才児向けに、Logo を単純化し、タートルの代りになる木製ロボットへの指令という形式でプログラミングを体験させる教材 Cubetto が発売されている。多くの教育機関で活用され、デザイン賞も多数受賞していて、日本にも輸入されている[3]。

Cubetto では、図1のような6×6のマスのカーペットの1マスにおいたロボットに12個までの命令を与えることで、出発のマスから目的のマス目に到達できるプログラミングを体験させている。



図1 ロボットを置いて動かすカーペット

Figure 1 A Carpet for moving a wood robot.

用意されている命令は、Forward (前へ), Right (右へ), Left (左へ) の3つだけであるが、この他に4ステップまでの手続きを呼び出す命令(Function)が用意されている。命令は12個しか書けないので、3命令だけではカーペットの端から端まで、全てのマス目に到達することはできるとは限らない。しかし、Cubetto では4命令までの命令列をFunction (手続き)として定義することができて、これを呼び出すことで、ロボットをカーペット上のどこからどこまででも動かすことが可能になる。



図2 ロボットと命令版

Figure 2 The robot and its direction board

これを児童に使わせる時に、まず自分がロボットになったと思って、命令を実行させてみるのが非常に重要である。命令を自分の体の動作感覚として理解しないと、プログラムはうまく作れない。命令の意味を自分の体を動かす時と同じように捉えられなければならないのである。

体を使うスキルの訓練では、自分が行ないたい運動を頭の中で思い浮かべて、それを実行する状況を思い浮かべるイメージトレーニングが有効である。

プログラミングも、命令列を書く時に、それがどのような効果を生むかを予測する必要がある。プログラムがうまく作れない人は、イメージを作らずに、適当に命令を並べて、それが予想する動作をしないと、命令をいじり出して試行錯誤を繰り返す。

このような試行錯誤は、命令の意味がよく理解できない場合には、理解する行為として意味があるが、アルゴリズム構築の段階で行なうことは望ましくない。

この命令の体感作業において面白い現象が観測された。「前へ」はすぐに理解されて予想通り動くのであるが、「右へ」、「左へ」の命令を体感させると、右や左に向くだけでなく、必ずその後直ぐに1歩進んでしまうのである。これは、生徒だけでなく、プログラミングに慣れた人でも、意識せずに言われるままに体を動かそうとすると、そのようになってしまう。英語で行った場合にも、同じ現象がおきるかどうかは興味深い問題である。また、この問題を解明することは、認知言語学の面白い研究課題である。

幼児が1通りロボットを思い通り動かせるようになったら、どこからどこまで行かせるかという問題自体を考え

させることができる。その際カーペットのマスに描かれた絵を使って道筋の物語を作らせるという作文教育もできる。

このように、幼児用の教材であっても、プログラミングの本質部分が含まれ、それを問題作成とその言語化にまで広げることで、active learning を行なうことができる。

4. 小学生用教材 CodoMonkey

CodeMonkey は5歳から18歳までのプログラミング義務教育が実施されているイスラエルで、スタートアップ企業 CodeMonkey Studios 社 により開発され、同国内の子ども達のプログラミング学習に広く利用されている[4]。現地で3か月間にわたり開催された CodeMonkey プログラミングコンテストには、1700校もの子どもたちが参加し腕を競った実績もあり、全世界の利用者数は130万人を超えている。

CodeMonkey はプログラミング学習のためのゲームで、サル「モンタ」がゴリラにバナナを奪われてしまうところから冒険が始まり、友だちのカメやヤギと協力して道中様々な障害物に遭遇しながらバナナを獲得していく物語の形式をとっている。効率よくバナナを獲得するには、問題点と解決策を見つけ、coffee script の「コード」で解決策を表現しなければならない。Coffee script は JavaScript に変換される、可読性の良い実用言語である

Cubetto の Forward は隣のマスにしか進めなかったが、codeMonkey では進む距離と方向を指定できるようになっている。修得できるプログラミング技術はオブジェクト、変数、配列、FOR ループ、条件式「AND/OR」、UNTIL ループ、IF ELSE 文、関数などである。

最初の第0ステージは、モンタが自分のバナナをゴリラに取られてしまう、という導入が示され、次に "Step 15" と書かれたスクリプトを RUN コマンドで実行することだけである。これでめでたくバナナに到達できる。次の第1ステージでは、"step 10"と書かれたスクリプトの"10"を"15"に書きかえることが求められ、そのように書き直して実行すると、バナナがとれることになる。このように、学習者が行なうべきことが明確に示されることが、この教材の特長である。



図3 距離を測ってバナナを取りに行く

Figure 3 Measuring the distance

次の2ステージでは、モンタとバナナの間を距離を測る

ツールが示され、それを用いて測った距離を与えると、めでたくバナナがとれることになる。次の3ステージでは、モンタの向きを左右に変えるコマンドが導入される。これらのコマンドはキーボードを使わずに下にあるタイルを選択することで行なえる。これで2行のスクリプトを完成できた。ステップ5では、モンタとバナナの間に生け垣があるので、これを迂回してとりに第かなければならない。このように、ステップごとに問題が少しずつ変わり、それに応じてスクリプトも少しずつ複雑化していく。

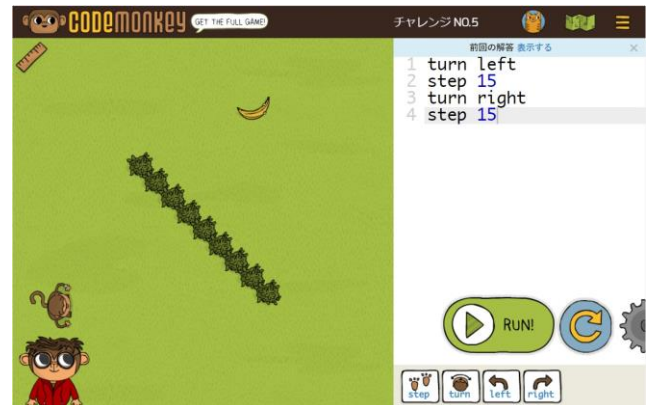


図4 生け垣を迂回してバナナを取りに行く

Figure 4 Round-about a bush for a banana

14ステージでは、オブジェクトとして用意されたかめを利用してバナナをとりに行く。川に浮かぶ亀の上まで進み、その後亀を動かすことで最初のバナナにありつく。次にまたモンタ自身が動いて次のバナナをとる。



図5 川にいる亀に乗ってとりに行く

Figure 5 Taking advantage of a turtle

作られたプログラムが正しくないことも当然起る。これに対して

4行目の直そう

step10 ってなんのこと？

step 10 のことかな？

空白（スペース）を入れてみよう

といったメッセージが出る（図6）。先に進むと、行数が長いからもっと短くするように、という指示が出て、コードを改善すること求められるようになる。正しく動くだけで

は、合格しないのである。



図 6 エラーメッセージ

Figure 6 Error messages.

次に導入される機能は繰り返しである (図 7)。



図 7 繰り返し命令を使う

Figure 7 Repeated commands

利用に関しては、ステージ 30 までがメールアドレスの登録だけで無料で使えるような販売政策がとられている。

以後、変数、配列、FOR ループ、条件式 (AND/OR)、UNTIL ループ、IF ELSE 文、(引数付) 関数までが 200 個のステージで学べるようになっていく。粒度の小さな文法事項ごとに、具体的なゲームの問題を設定して、それを問題解決に使う体験をさせていることが、効果をあげている。

5. 幼児と大学生

幼児にタートルグラフィックスを教えれば、自由に作品が作れるようになることが知られている。下記はハンガリーにおける例 (図 8) であるが、日本でも同様の報告 (図 9) が行なわれている[5]。

1990 年代前半の、インターネットが普及する前の慶応大学 SFC では、タートルグラフィックスを駆使して、すばらしい絵を描く学生がいたが、欲しいソフトウェアが自由に購入できるようになると、プログラミングへの興味が薄れ、ソフトウェアの利用だけで満足する学生ばかりになって、プログラミング教育への関心が薄れてしまった。

授業としてのプログラミングが続くなかで、受講学生に、幼児が描くタートルグラフィックスを用いた絵 (図 8) を見せると、自分ではとても作れそうもないことを幼児が行

っていることに仰天する (図 8)。

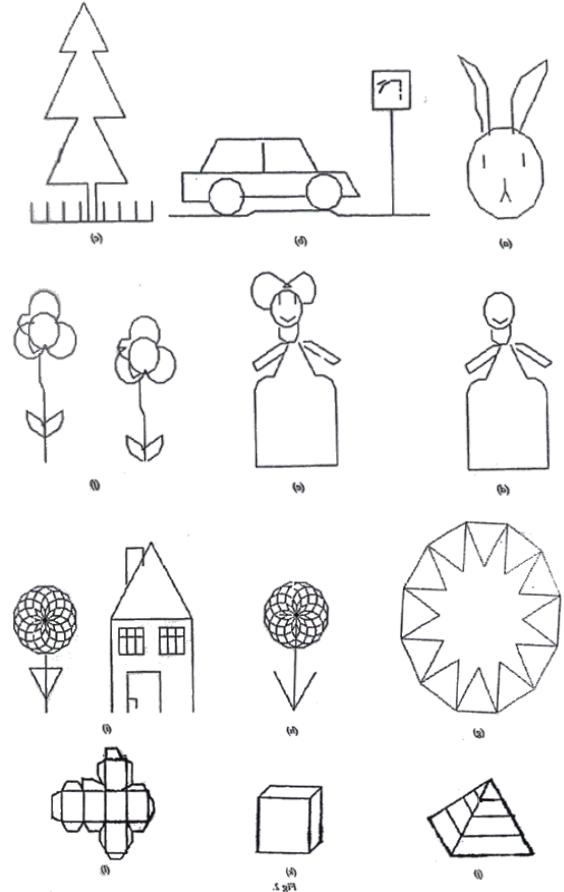


図 8 ハンガリーの幼児 (5, 6 歳) の描いた図

Figure 8 Drawings by Hungarian kids

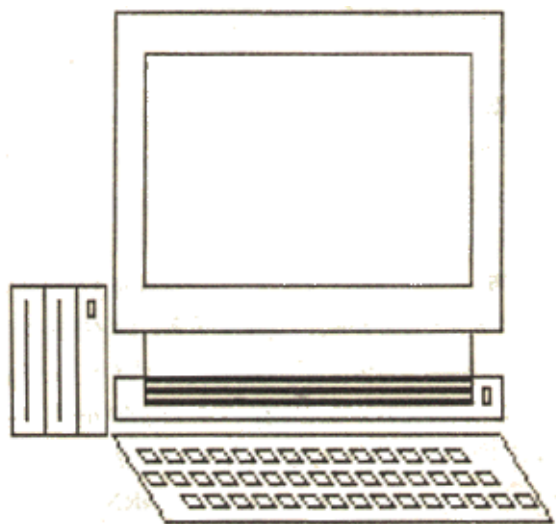


図 13 ひらがな LOGO で描いたコンピュータのシステム図 (この絵を描く「コンピュータ」という手続きの内容は、「付録 ひらがな LOGO のプログラム・リスト」を参照のこと)

図 9 日本人 (6 歳) の描いた図

Figure 9 A drawing of a Japanese Boy

幼児でも教えられるプログラミングを、18歳の大学生に教えてもうまく行かない理由を考えてみよう。

論理的な思考を行なう機会が日本の教育体系の中で殆ど無くなってしまったことが考えられる。大学入試の苛烈化とともに、考えていたのでは間に合わなくなり、特殊な受験技術を用いないと良い成績がとれなくなった。この影響が特に顕著なのが、偏差値の高い大学の受験生である。

国立学術情報研究所で東大に入学できるロボットが可能かを検討する東大ロボットの研究が新井紀子教授によって行なわれたが、そこで用いられた方法は、問題文を読まずに、キーワードの頻度等を手がかりに、最も正解の確率が高い解を選ぶ方法である。情けないことに、ロボットだけでなく受験生も問題文を読む能力が落ちていてロボットと同じような方法を用いて解答しているらしい。このように、大学生が文を正確に読もうとせず、ロボットになる努力をする国は日本しかないであろう。

1960年代までの日本の高校数学には、伝統的なユークリッド幾何学が含まれていて、その中では証明を書く訓練が行なわれ、文を正確に読み、書く訓練が行なわれた。また作図問題は、プログラミングと同じく、結果が正しいかどうか明確に生徒も理解でき、試行錯誤を含む論理的な作業計画の構築が行なわれた。こうした教材が高校数学から無くなって、幾何学は中学校で終らせるようになると、学校教育の中で論理的な訓練を行なうことはほとんどなくなってしまった。

6. 学校教育におけるプログラミング

米英や、電子政府を作ったエストニアでは、プログラマーの数が全く足りないという問題に直面して、その解決手段として小学校でのプログラミング教育を行なおうとしている。

また、オクスフォード大学の研究で分ったように、これからコンピュータとネットワーク技術の進歩で、現在ある仕事の半分は人間の仕事ではなくなる、という予想が真実味をおびてきたことから、コンピュータを使いこなせないと、豊かな社会生活が送れなくなる、という恐怖が生まれ、親が政府にプログラミングは学校教育で行なうべきだと圧力をかけたということもあるらしい。

理由は何であれ、コンピュータとそのネットワークが社会の基盤となったことから、その原理を市民が知っておくことは、健全な社会を構成していくために必要である。そうでないと、コンピュータの専門家が自由にシステムを構築して社会全体を支配するという、恐ろしい社会が出現してしまう恐れがあるからである。

米英では、プログラミング教育の内容として *computational thinking* を育てるという主張が成され、日本でも、これを現場の教員が理解し易いように、「プログラミング的思考」を育てることを学校教育の目的とすることが

決まった。

外国でも日本でも、これらの教育目的が何であるかについて、正確な共通理解は得られていないようである。プログラミング教育を実践しながら、はっきりさせて行くことになると思われるので、ここでは、プログラミング教育には何がなければならないか、また何があってはならないかについて、論じてみたい。

まず、なければならないことは考える能力の育成である。これには、問題を解くことである程度達成されるが、それだけでは十分ではない。問題自体が何であるかを同定すること（問題認識）が、それ以上に重要なことになる。

問題を考えることなど、小学生にできるのかと思われるが、実際には児童は大人が考えている以上によく考えている。Cubetto では、ある場所から別の場所に行く方法を学ぶ必要があるが、それができたらどこからどこへ行くのかという問題自体を考え出すこと、またその道筋の状況から、その道筋の物語をつくり出すことも、言葉を覚えだした幼児はそれなりに行なうことができる。

教師の役割は、その過程がうまく進んでいくように支援していくことになる。それには幼児のしたい事が何であるかを感じとる能力が必要であり、従来のプログラミング教育と違ってプログラミングの能力よりも、幼児教育の専門家としての能力が必要となる。

一方、やっつけはいけないことは、正しい解答を教えることである。従来は正しい知識とスキルを教えこむことに教育の力点が置かれてきた。「教えるはいけない」という行動指針は、日本の教師にはむずかしい。

また、どこまでを教えて、どこから教得てはいけないか、という区別をどう決めるかも問題である。プログラミングの場合、文法の理解までは、教えなければいけないことである。しかし、このレベルでも、なるべく生徒自身が自分で理解できるような環境を用意して考えさせるのが良い、この方法は望ましいが、時間がかかるため、教えて授業を進めざるを得ない時も起こりそうである。

Cubetto レベルの教育では、プログラミング自体は考える能力のある大人であればだれでも指導できるであろう。

問題は、CodeMonkey の教育である。200 ステージでは引数つきの関数があつかわれることになる。この段階になると、指導者はプログラミングに関して系統的な学習をしておかないと、授業を行なうわけには行かないであろう。

おそらく、100 ステージ位までなら、時間さえかければ、理解は可能と思われるが、こうした教材をほとんどあつかったことがない日本の小学校教師にどこまで可能かは、やってみなければ分らない。ただ、外国では、遠からず（10年以内）にこの程度の教材が小学校で使われるようになるだろう。先進国では30年前から、*computer science* と教育学を両方学ぶ、情報教師の育成が計画的に行なわれてきた。

日本ではビジネスとしての小学生へのプログラミング教

育が都会では行なわれるようになったが、CodeMonkey はあまり使われていない。その理由はおそらく、教師に力量が要求されるからではないかと思われる。実際、他の教材の場合は、深く理解する必要がなくても何とかなる。小学生が学ぶ程度の学習事項であれば、大人ならすぐに学べると考えたのであろう。

しかし、実際には幼児に教えればできるものであっても、大学生ではそれを学習することが困難なのである。おそらく、世界中で5歳児からのプログラミング教育が始まった理由の一つは、この点にあるのではなかろうか。

識字教育は大人になってからでは困難であるが、社会にとって市民全体で共有されるべき知識と能力だから、世界中で行なわれているのである。Cubetto で行なわれるプログラミング教育は、加減算より幼児に理解し易いものであり、こうした教育が普及すると算数・数学教育は大きな影響を受けることが予想される。

考える力を養うということが、小学生教育におけるプログラミング教育の重要な役割である。「考える」とはどのようなことをプログラミングという文脈で考えると、(複雑な)事態を論理的な表現を用いて記述し、それを役立つものにするのである。こうしたことは日常の行動でもしばしば現われ、例えば、「計画を立てる」という活動の実態である。

従来、計画自体を立てる訓練より、実績のある計画を実行することに教育の重点は置かれていたように思う。そこでは、正しいことを覚えることが重要で、それを間違いなく実行することが求められてきた。新しく自分で考えることは、信頼性の点で問題があることから、市民が自分で行なうことではないと考えられてきた。

しかし、プログラムとして記述される仕事は、コンピュータで実行が可能な時代が来ている。このために人の仕事が無くなってきているのである。残された人間の仕事は、問題を発見し、それを自分自身で解決する能力である。

7. おわりに

Cubetto と CodeMonkey という具体的な教材の紹介を行った上で、それが学校教育としてどのような意味を持つかについて議論した。Computer science を知らない小学校教師でも、言語教育としてプログラミング教育が行なえることを示した。

筆者は10年前に、「情報教育学の確立に向けて」という論考を発表した[6]。そこで行った主張は、「情報教育学は単なる情報技術のみを対象とするのではなく、人文・社会科学的内容をとり入れる必要があり、学問分野の融合と、現場の教師が情報教育の確立に主体的に参加できるものでなければならない。このためには明晰性、革新性、発展性を評価軸として、現場教師の共感を呼ぶことのできる研究成果を論文とすべきである。」と主張した。

また、「研究は社会に役立つ情報を新たに作り出すこと」とした上で、成果に科学的エビデンスとして客観性を求めることに疑問を呈した。客観性追求のために、有用性の少ない研究が増えただけでなく、本来母集団を確定し、その中からランダム・サンプリングを行ってはじめて統計解析が行なうことが可能になるのであるが、このような前提を示さずに、統計解析だけを形式的に行ってそれで科学的エビデンスである、とする誤解が蔓延している。人間の心が影響する教育活動は、よほどの大規模予算がなければ、科学的な教育研究は不可能である。従って、実践的な教育研究は、教師にとって有用な事実の客観的な記述を教育学の立場で分析することが最も重要である。「幼児の作った作品を見て、大学生が仰天した。」という事業の記述は、一般には知られていない重要な情報であり、それを指摘することは重要な研究活動である。

参考文献

- [1] 勝野頼彦(研究代表者):社会の変化に対応する資質や能力を育成する教育課程編成の基本原則(平成24年度 プロジェクト研究調査研究報告書), 国立教育政策研究所, (2013), <https://www.nier.go.jp/kaihatsu/pdf/Houkokusho-5.pdf>
- [2] 三宅なほみ:「21世紀型スキル」は世界標準の力, (2012), http://www.disc.co.jp/uploads/2012/03/2012.1.10miyakeshi_jinzai.pdf
- [3] プリモトイズキューベット. <https://www.primotoys.jp/>
- [4] Codemonkey <https://codemonkey.jp/>
- [5] 子安増生:幼児にもわかるコンピュータ教育 LOGO プログラムの学習, 福村出版, (1987).
- [6] 大岩 元:情報教育学の確立に向けて、情報処理学会研究報告 2006-CE-80, (2006).