

論理的思考能力の育成を目的としたプログラム対戦型ゲーム

佐々木 整 竹谷 誠
拓殖大学工学部

<抄録> プログラムのデバッグ作業で必要とされるなど、論理的思考能力は大変重要である。そこで、この能力の育成を目的とし学習者が作成したプログラム同士を対戦させる、ゲーム型学習システムの開発を行っている。本稿では、このゲーム型学習システムの紹介を行うと共に、高校生を対象にした学習者のプログラム推定能力と論理適性との関係について述べる。

A Program Opposing-Type Game System to Cultivate the Ability to Logical Thinking

Hitoshi SASAKI and Makoto TAKEYA
Faculty of Engineering, Takushoku University

Abstract: It is extremely important to be able to think logically when debugging programs. We are presently developing a learning system that makes use of a game to foster this ability in learners by having them compete against each other. In this paper, we will introduce the educational game and discuss how it relates to the ability of high school students to figure out problems and their aptitude for logic.

1. はじめに

プログラム開発の場面では、既存のプログラムの保守や新規作成したプログラムの単体テストや既存のプログラムとの連携のテストなど、いわゆるデバッグのための作業に多くの時間やコストを要しており、プログラムの作成能力だけではなく、バグの追跡、発見、解決などいわゆるデバッグのための能力が必要とされている。これらの能力の育成は、大学などではプログラム演習などにより実現されているが、一般的なプログラム演

習では、課題が構造的に与えられ、課題の解決が方法論として体系化されており、その方法論の習得を中心とした内容であることが多い。一方、プログラム演習でのデバッグ作業は課題実現の過程で付随的に行われているため、デバッグの能力を向上させるような論理的思考能力の育成がなされていないのが現状である。すなわち、バグの原因を論理的に追求して問題を解決し、正常なプログラムに改良するための、論理的思考能力の育成が十分行われていない。さらに、デバッグ作業は、その性質上一般に学習者のやる気を減退させ

る場合が多く、このデバッグ能力を効果的に育成することは非常に困難であった。

そこで、本稿では学習者がゲームを通して楽しみながら、さまざまな発見や試行錯誤的学習をすることで論理的思考能力の育成を目指す、ゲーム型学習システム[1,2]の提案を行う。

2. システムの概念とシステム構成

2. 1 ゲーム型学習

本稿で述べるゲーム型学習システムは、学習者が対戦ゲームを通して論理的な思考に基づきプログラムを作成するとともに、対戦相手のプログラムを推定する能力を育成することを目的とするものである。ここでの対戦ゲームとは、格闘技を例にとれば、2組の学習者がそれぞれ自らの選手の動きをあらかじめプログラムという形で表現し、他の学習者が作成したプログラムと戦わせることでその勝敗を競うものである。なお、この対戦ゲームでは対戦相手のプログラムを閲覧することを禁じている。そこで、学習者は対戦の過程から敗因を調査したり、対戦相手のプログラムを推定し自らのプログラムを改良していく。このプロセスを繰り返すことにより、解決しなければならぬ問題を発見し解決法を見出す能力を育成するものである。つまり、ある時点で不明な相手のプログラムのロジック（ブラックボックス）に対して、論理的思考に基づき、その不明なロジックの特徴を調査できる適切な入力を選択し、そこで得られた出力との関係から問題となる箇所を特定するとともに、その解決法を決定することを実践で繰り返し学ぶことで、能力を育成するのである。例えば、あるブラックボックスにAという入力を与えるとCという出力がえられたとする。この時点ではブラックボックスの中身の判断は困難であるので、続いてCという入力を与えることにする。もし、この入力に対する出力がEであったら、このブラックボックスは入力した文字を2文字シフトして表示するものと推定できる。

この推定が正しいかを、他の入力で検証することになるだろう。一方、入力Cに対しAという出力が得られた場合、ブラックボックスはAとCの変換を行っていると考えられる。他の入力ではどのような変換が行われるか、例えばBを入力するなどして検証する必要があるだろう。本システムはこのような作業を対戦相手のプログラムをブラックボックスとして行うのである。

これまでに、対戦ゲームを用いた共同学習システムとして「あるごありーな」[3]が提案されているが、これらはアルゴリズム的な考え方を育成することを主目的としており、学習者による問題の発見を積極的に支援するものではない。また、教室などの狭い範囲での使用を前提としているため、対戦の相手が限定されるという制約がある。そのため、対戦によって生じる問題もある一定の方向へ収束しやすく、問題発生のための多種多様な状況を構築することは困難である。本システムは、学習者が様々な角度から状況を分析し、問題の発見、解決法の検討を実現できるよう支援を行う。また、インターネット上でシステムを実現することで、世界中の多くの学習者と対戦することができ、様々な問題発生のための多種多様な状況を学習者に提供することができる。このように、本システムとは目的、実現方法ともに異なっている。

2. 2 システムの構成

本システムは、学習者がより多くの発見や試行錯誤をする状況を提供できるよう、インターネットを用いて世界中の多くの学習者との対戦を前提としている。そのため、システムは全てJavaとORBの1つであるHORBを用いてシステムを構築している。ここで、システム概念図を図1に示す。

本システムは、インターネット上の本システムのサーバと学習者に提供されるクライアントの2つで構成されている。サーバは、学習者の対戦プログラムや学習過程、対戦過程や履歴などを保

存し、学習が円滑に行えるよう支援する。本システムでは、インターネットに接続され Java が動作するコンピュータさえあれば、学習を始めることができる。またサーバ側は、学習者と同様の環境と HORB システムのみで実現可能である。

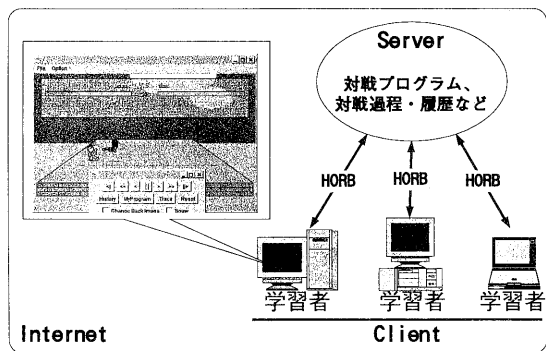


図1 システムの概念

2.3 ルール

学習者は、2.2で述べた環境下でプログラムを作成し、インターネットを介して他の学習者と対戦を行う。この対戦は、次のルールに基づき勝敗が決まる。

(1) 体力

選手はそれぞれ100ポイントの体力が与えられる。この体力は、敵から攻撃を受けると減少する。減少の度合いは、敵の攻撃の方法と攻撃を受けた時の選手の状態によって変化する。例えば、正面から攻撃を受けた時よりも、背後から攻撃を受けた時のほうが体力の減少幅が大きい。

(2) 対戦時間

対戦は100ターンで必ず終了する（時間切れ）。1ターンは、前進など選手の動き1つに相当する。

(3) 勝敗

相手の体力ポイントを0にするか、相手を場外に押し出すと勝利となる。ただし、同時に体力ポイントが0になったり、場外になった場合は引き分けとする。また、時間切れの場合はその時点で体力ポイントが多い方を勝ちとし、等しい場合は引き分けとする。

2.4 対戦プログラム

学習者が作成するプログラムの例を図2に示す。このプログラムの言語仕様の詳細は紙面の都合上割愛するが、C言語に類似した独自の言語を使用している。選手の動作の表現は、「キック」を行いたい場合には「Kick(NORMAL)」と記述し、「前進」を行いたい場合は「Go()」と表記する。相手までの距離や選手の位置、残りの体力などステータス取得の命令は「Distance()」などのように全て関数で表現する。

```
main() {
    while(true) {
        if(Distance(>2) {
            Go();
        }
        if(Distance(<=2) {
            Punch(NORMAL);
        }
        if(Distance(==3) {
            Kick(NORMAL);
        }
    }
}
```

図2 プログラムの例

図2に示したプログラムは、対戦相手との距離が2以上の場合は前進、2以下の場合はパンチ、距離が3の場合はキックという動作を、対戦が終了する（勝敗が決まる）まで繰り返すというものである。

3. 学習の流れ

本システムの利用画面を図3に示す。このように、「プログラミングエディタ」、「対戦車選択機能」、「アニメーション機能」、「プログラム解析機能」、「トレース実行機能」、「プログラム表示機能」の6つの特徴的機能で構成されている。以下では、実際の学習過程に沿って、それぞれの機能を説明する。

3.1 選手動作のプログラミング

学習者は初めに自分の選手の動作をプログラムで表現する。学習者が常に同じ環境で学習が行

なえるよう、Java でプログラミングエディタを作成した (図3①)。このエディタには、コンピュータに不慣れた学習者の利用を考慮して、ソフトウェアキーボードや予約語登録などのマウスによる簡便なプログラム入力を実現している。

3. 2 プログラムの登録

続いて、作成したプログラムをサーバに転送し、選手としてシステムに登録する。これにより、インターネット上の他の学習者との対戦が可能となる。なお、登録時にプログラムの文法をサーバがチェックするので、文法エラーがあるプログラムは登録できない。登録後は対戦相手を、これまでの勝敗から決定されるレベルを参考に「対戦者一覧」から選択する (図3②)。

3. 3 プログラムの対戦

アニメーション機能 (図3③) を用いて、3.2 で選択した対戦相手との対戦過程を観察する。この対戦過程は現在のターン数や実行コマンドと共にアニメーション表示される。ビデオプレーヤのように、巻き戻しやコマ送りなどの機能も有している。

学習者はこの対戦過程を見ることで、敗因となった行動や相手の動作などを視覚的に把握することが出来る。

3. 4 プログラムの解析

対戦結果から、学習者はその敗因、あるいは対戦に勝利した場合はより強くするための検討を行う。この検討を様々な角度から分析できるよう、対戦過程を追跡できる、プログラム解析機能 (図

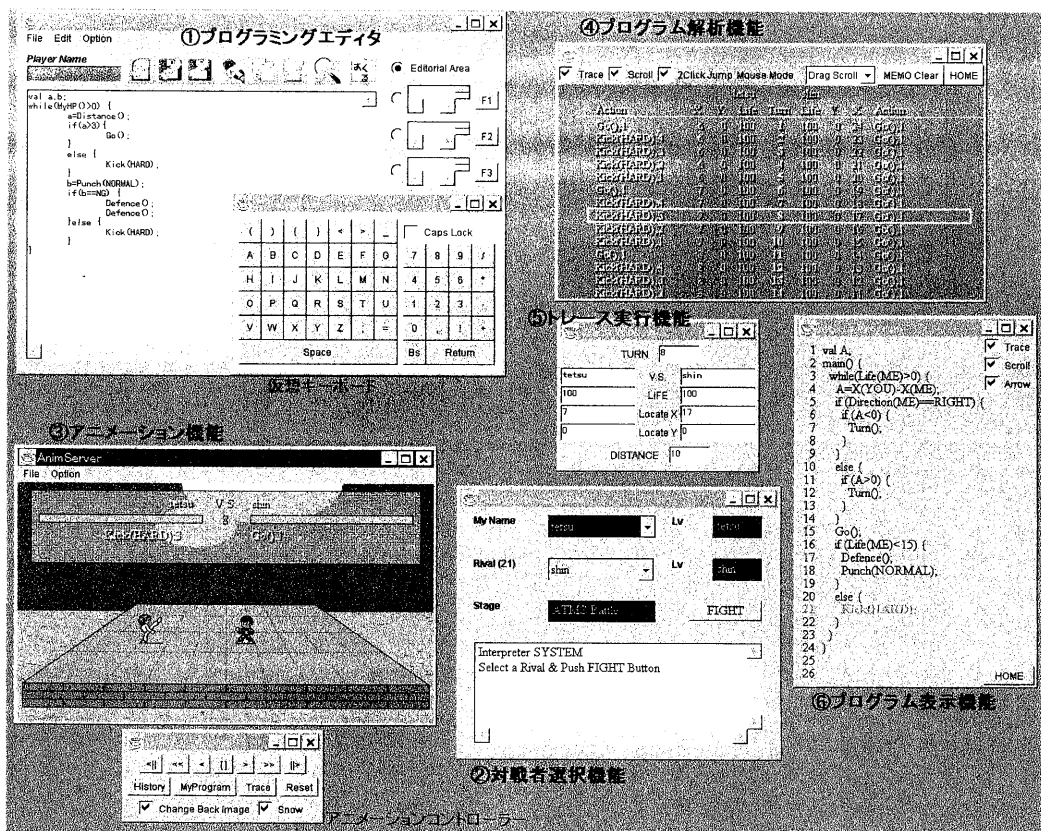


図3 システム全体図

3④、⑤)を用意した。プログラム解析機能は、各ターン毎に選手の座標と体力、実行中のコマンドを対比して一覧表示を行う。この機能によって対戦相手の行動から規則性や因果性を見出すことが容易になる。さらに、学習者は、アニメーション機能で対戦を観察しながら、プログラムリストをトレース表示することで、プログラムのどの部分が実際に実行されているのかを、明確に把握することが出来る(図3⑥)。

このように、システムから得られる相手プログラムに関する情報は、相手が実行した命令(行動)や相手の状態という断片的なものでしか得られない。学習者は、この断片的な情報を効率よく集め、それらを統合化し分析をおこなう。不足している場合は必要な情報を得るために、自分のプログラムの一部を変更し、再び対戦を行うことで変更した動作に対する相手の動作の変化を観察し分析する。この作業を繰り返すことで、どのような変更で相手の動作が変化するかを把握し、プログラムの推定を行う。

4. 本システムを利用した実験と評価

本システムの目的は、すでに述べたように論理的思考能力の育成にあるので、本システムの評価は実際に学習者が本システムを利用することでどの程度論理的思考能力が育成されたかを評価しなければならない。しかし、そのためには長期的視野で実践授業を行わなければならない。本稿ではその基礎実験として、論理適正試験による学習者の論理的思考能力と、本システムで学習するプログラムの推定能力との関係調査を行ったので、それについて報告する。

4.1 実験内容

この実験は、本システムを用いたプログラムの推定能力と、本学入学試験の一部で利用している論理適性検査との相関を調べ、対戦相手のプログラムを推定する能力と、論理的思考能力の関係を

明らかにすることが目的である。被験者は、本学付属高等学校の2年生の生徒35名であり、全員がプログラミングの初心者である。

まず、被験者全員について論理適正検査[4]を実施した。この論理適性検査は、本学入学試験の一部で用いられているものであり、全10問で各問10点の配点である。適性検査の各問題については次節に示す。

続いて、プログラム推定能力の測定実験を行った。まず実験の準備として、被験者に対し本システムのプログラムを推定するための基本的な機能の操作方法や、プログラムの命令、条件文などの記述方法とその動作の説明を2時間行った。さらに、数種の例題プログラムを用いて、プログラムの説明を踏まえながら、実際の推定方法の説明を3時間行っている。

この実験で学習者に推定させるプログラムは、図2で示したプログラムである。被験者は、90分間でプログラムを作成し、推定しなければならないプログラムと対戦させることを繰り返して、プログラムの推定を行う。推定したプログラムはこの講義の担当者が10点満点で採点を行った。このプログラムの推定結果と論理的思考能力との関係を検証するため、論理適性検査の結果との相関について分析を行った。

4.2では論理適性検査の各設問について、簡潔に紹介し、4.3で分析結果に付いて述べることにする。

4.2 論理適性検査の各問題の特性

本システムでのプログラム推定にどのように関係しているかを詳しく調査するため、論理適性検査の各問題について、どのような特性があるか分析を行ったので、これについて述べる。また、問5、問9を除く各問題の内容は、参考文献[4]を参照されたい。

問1：提示された情報から、そこにある条件を見出す問題である。この問題は、あり得ないパターンを効率良く排除し、解答できるかが重要とな

る。

問2：組み合わせの問題である。組み合わせの問題は、問題内の必要な要素を的確に抽出しなければならないが、公式を覚えていると比較的容易に解答できる。また、組み合わせを全て検索することでも正解は得られる問題である。

問3：平面図形の問題である。しかし、平面図形の軌跡問題のように、提示されている平面図形を動かすといった、様々な視点からの情報の整理を必要とせず、試行錯誤的な問題である。

問4：立体図形の与えられた情報を整理する問題である。さらに、整理した情報を再構築し、そこで見出した新たな情報を解答とする問題であ

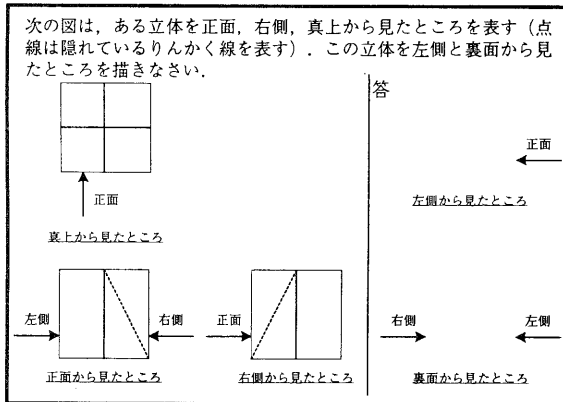


図4 適性検査の第5問目

る。

問5：立体図形の全形を推定する問題である（図4）。この問題は、先述した問4の応用であり、様々な角度から視点を変えて、情報を整理・再構築し、そこから得られた新たな情報をもとに、別の新たな情報を見出す問題である。

問6：断片的な情報をもとに、状態遷移を作成する問題である。この問題は、問1の応用であり、提示された条件を整理し、その条件を満たしている情報を発見するものである。

問7：与えられた条件からその条件に該当するものを見出す問題である。この問題は、順序立てて推論を行わなければならない、数量的な思考を必要とする問題である。

問8：与えられた情報を整理し、連立方程式を立て、その連立方程式を解く問題である。この問題は、物事を数学的に考えられるかを問う問題であり、数量的な知識を必要とする。

問9：他の問題と違い、推論の過程を記述する問題である（図5）。推論の始点と終点が提示されており、その間の一連の過程を推定しなければならない。

問10：集合の問題であり、数量的な問題である。情報が断片的に与えられており、その情報をもとに積集合などの情報を見出していく問題である。

2個のグラスA, Bがあり, Aは4リットル, Bは3リットル入る. これらのグラスに外から水を一杯に注ぐか, 1つのグラスの水をすべて捨てるか, 一方のグラスから他方のグラスへ水を注ぐかの操作をして, Aに2リットル, Bが空の状態を作りたい. もちろんグラスに目盛りはついていない. Aにxリットル, Bにyリットル入っている状態を(x, y)と記す. 移っていく状態の順序を記しなさい. 初期状態はどちらのグラスも空, つまり(0, 0)とする.

参考 (状態遷移の例): 今Aに3リットル入っており, Bが空の状態. Aに水を一杯になるまで注ぐと状態は(3, 0)から(4, 0)に移る.

答 $(0, 0) \rightarrow (,) \rightarrow (,) \rightarrow (,)$
 $\rightarrow (,) \rightarrow (,) \rightarrow (2, 0)$

図5 適性検査の第9問

これら、各問題の分析をもとに、実験を行った。

4.3 実験結果と考察

実験の結果を図6に示す。図6に示す縦軸は、適性検査の点数で満点は100点である。横軸は、本システムを利用してのプログラム推定の点数であり、満点は10点である。また、図中の直線は回帰直線を示している。ただし、同点者が存在しているため、図中の各点が重なって表示されている。

図6に示したとおり、今回の実験において、論理適性検査の合計とシステムを利用してのプログラムの推定結果には相関があり、相関係数は0.412と高い相関を得ている。

また、各問題に対しての関係を調査した結果、最も相関が高かった問題の、問5と問9の関係を図7、図8に示す。縦軸は適性検査の問題の点数

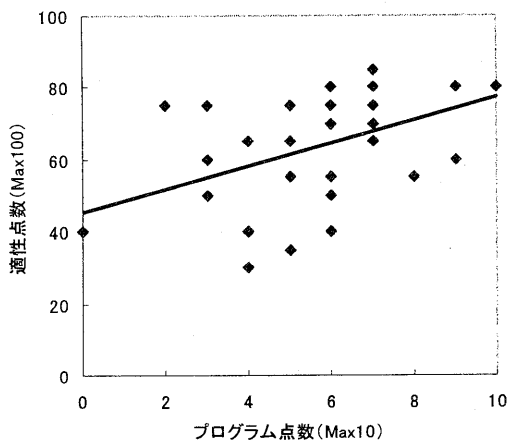


図6 適性検査の合計と推定

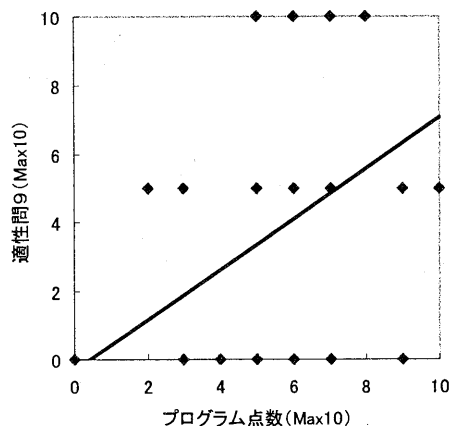


図8 適性検査の問9と推定結果との

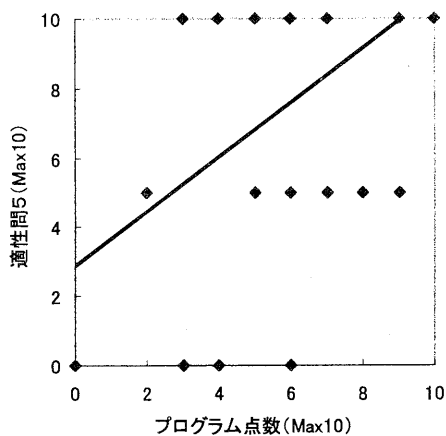


図7 問5と推定結果との相関

を示しており、満点は10点である。また、図4と同様、図内の点は複数の点が重なってプロットされている。図7に示す通り問5の結果と本システムを用いてのプログラム推定結果との相関は相関係数0.451と高い。これは、問5は3次元の図形をイメージするために、情報の統合化を行い、その統合化された情報をもとに分析を行って

いく問題であるため、本システムでのプログラム推定と相関が見られたと考える。

さらに、図8に示した通り、問9の結果と本システムを用いてのプログラム推定結果の間にも相関があり、相関係数は0.385である。問9の問題は、状態遷移数が限定されているため、最適解を求めなければならない。また、提示された情報を導くための新たな情報を見出さなければならない問題であるため、より強い相関が見られたと考える。

さらに、問5、問9の合計点との相関を調べた結果、相関係数0.507という高い相関が得られた。これらの問題の環境や解答方法は、本システムでのプログラムの推定方法と大変類似しているため、大きく関係が現れたと考える。

今回の実験から、論理適性検査と本システムでのプログラム推定との関係を調査した結果、それらの間には相関があることが認められた。よって、本システムを用いてプログラムの推定能力の訓練を行うことにより、論理的な能力の育成が行える可能性があると考えられる。

5. おわりに

本稿では、インターネットを利用した、論理的思考能力の育成を目指した、ゲーム型学習システムの紹介と、本システムを用いた実験学習の報告について述べた。今回の実験では、論理適性検査と本システムでのプログラム推定との関係を検証した結果、本システムを用いてプログラム推定を繰り返し行うことにより、論理的思考能力の育成が行える可能性のあることが実証できた。特に、与えられた情報を様々な角度から見直し、情報を再構築して、新たな情報を得るといった能力や、推定を行う際の順序立てを的確に行うことができる能力を育成できる可能性があると考えられる。

今後、数多くの長期的な実験学習を行い、本システムを用いてプログラムの推定の訓練を行うことにより、論理的な能力の育成が行えるかの検証を行っていかなければならない。

参考文献

- [1] 佐々木整，竹谷誠：情報教育を目的としたゲーム型協調学習システム，信学技報，ET96-58，pp.135-140(1996).
- [2] 森川哲史，佐々木整，竹谷誠：情報教育を目的としたゲーム型協調学習システム(II)－学習者環境による問題発見・解決支援－，信学技報，ET97-16，pp.9-16(1997).
- [3] 加藤浩，井出有紀子：ソフトウェア教育システム あるごありーなの構想(1)－設計思想－，電子情報通信学会春季大会，pp.475-476(1993).
- [4] 平成9年度拓殖大学入学試験問題集，拓殖大学入試課，pp.307-308(1997).