

自由エネルギー原理に従う循環的因果性の構成を支援するプログラミング学習環境の構築

The design of programming learning environment guiding the construction of circular causality based on the free energy principle

鷹觜 莉子, 山中 脩也, 長 慎也, 北島 茂樹, 丸山 農

TAKANOHASHI, Riko YAMANAKA, Naoya
CHO, Shinya KITAJIMA, Shigeki MARUYAMA, Minori

要旨

「行動」と「知覚」による循環的因果性を包含するプログラミング学習環境を、自由エネルギー原理に基づいて構成する方法について述べる。自由エネルギー原理は、脳についての諸理論を統一的に説明するために導入された規範的原理であり、それによれば、外部世界の認知は「行動」と「知覚」の循環を通して「思考」と結びついている。本稿ではプログラミング学習の目的を「実行可能なコンピュータプログラムを設計・構築すること」と位置付け、「行動」と「知覚」による循環的因果性の構成と維持を支援する環境を構成した。また、同環境の実践として3ヶ月に渡るワークショップをオンラインで実施し、その結果から循環の構成と維持についての考察を行なった。

1. はじめに

コンピュータプログラムは、コンピュータに対する命令を記述した記号列であり、プログラミングとは、実行可能なコンピュータプログラムを設計・構築するプロセスをいう。よって、プログラミング学習環境を構成する場合、

- (1) 実行可能な記号列となるための文法の伝達 (例：講義)
- (2) 実行可能な記号列を設計・構築するための具体的な実習 (例：演習)

の順序で学習環境を構成することが一般的である。しかし、特定の学習者群においては、

- 文法の伝達もまた記号であり、その意味の伝達が容易でないことがある
- 演習で実行不能となった際に、設計・構成をせず実行だけを目的とすることがある
- 多様な設計・構築法ではなく、ひとつの設計・構築法に固執してしまうことがある

という現象が見られ、その結果、学習環境への連続的な参加が難しくなる場合がある。

この問題に対し本研究グループは、自由エネルギー原理に基づくプログラミング学習環境を提案し、多様な学習環境でその実践を繰り返している。本稿では、同学習環境の詳細と、多様な背景を持つ学習者がオンラインで繋がった環境での実践について述べる。

2. 自由エネルギー原理

近年、行動下における脳活動の計測技術の発展とともに、脳神経科学の多様な機能を説明し、知覚と行動を包含する理論について活発な議論が行われている [1]. そのひとつとして、「(人間を含む)自己組織化されたシステムが環境内で平衡状態であるためには、そのシステムの情報の自由エネルギーを最小化しなければならない」という、脳についての諸理論を統

一的に説明するために導入された規範的原理「自由エネルギー原理」がある [2]. この原理によれば、人間は、外部世界に関する「生成モデル」と「現在の推測」から計算される自由エネルギーを最小化するために、

- <1> 脳活動の「現在の予測」を変える（無意識的推論）と、
- <2> 「行動」によって「感覚入力」（知覚）を変える（能動的推論）

の二つを組み合わせている、と主張する (図1 参照). この原理が特に重要であるのは<2>の「行動」を伴う認識過程に関して説明可能な枠組みを提供した点にあり、その結果、「行動」と「感覚入力」（以下、本稿では知覚と表現する）の関係性が数理的なモデルとして表現されている [1]. つまり、この原理に従えば上記の<1>と<2>により、「行動」と「知覚」の循環的因果性が生まれ、その循環を通じて外部世界の認知につながる [3]. この外部世界の認知こそ「学習」であり、それは、脳活動を司る脳だけではなく、外部世界と相互作用している身体も使いながら、循環的因果性を通じて、外部環境を理解(身体化)しているといえる [4].

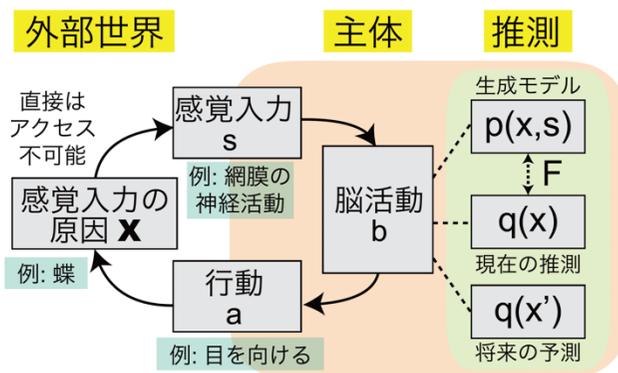


図1：自由エネルギー原理の図式 ([1] より引用)

3. 提案環境

3. 1. 自由エネルギー原理に基づくプログラミング学習環境

プログラミング学習環境を整備する主たる目的は、実行可能なコンピュータプログラムの作成である。しかしコンピュータプログラムの作成には、プログラミング言語の文法を把握するだけでなく、エラーメッセージに対応するコンピュータプログラムの修正や、他者が作成したコンピュータプログラムの内容理解など、「行動」と「知覚」を伴うコンピュータプログラムの設計や構築のプロセスを含むことから、プログラミング言語の文法を伝達するだけでは不十分で、「行動」と「知覚」を含む学習環境を構成する必要がある場合が多い。

本章では、前章で述べた自由エネルギー原理に従う、「行動」と「知覚」の循環的因果性の構成と維持を支援するプログラミング学習環境について述べる。

3. 1. 1. プログラミング学習環境の構造

プログラミング学習における「外部世界」とは「コンピュータ」（または「コンピュータの挙動」）であり、具体的に述べれば「プログラミング言語処理系」（または「プログラミング

言語処理系の挙動)である。言語処理系自身も何らかの言語で構成されていることが多いが、それ自体を読破することは困難であるため、一般的にはプログラミングの文法を何らかの方法で把握し、その文法に従いコンピュータプログラムの表出を行う。しかし、文法もまた記号であり、その意味の伝達が容易ではないことも少なくないことから、学習の継続が難しくなる場合が散見される。ここでは、このような場合に、文法の意味の伝達に基づく把握ではなく、「行動」と「知覚」の循環的因果性に基づく形で、外部世界（言語処理系または言語処理系の挙動）を認知していく方法を、自由エネルギー原理を基に構成する。

はじめに提案環境の前提条件を記載する。提案環境には「言語処理系」と「主体以外の Agent」が必須となる（図2を参照）。「言語処理系」については、主体がその言語処理系に対し「行動」を起こすことができ、また、言語処理系からの結果を「感覚入力」（知覚）できる環境でなくてはならない。同様に「主体以外の Agent」についても、主体がその Agent に対し「書出行動」を起こすことができ、また、主体以外の Agent の書出を「書出入口」（知覚）できる環境でなくてはならない。

多くの言語処理系は対話的なインターフェースを保持し、主体による「行動」の多くにおいて、主体は言語処理系からの実行結果をリアルタイムに「感覚入力」（知覚）できる。同様に、「主体以外の Agent」との対話的なやりとりにおいても、主体による「書出行動」の結果、主体はリアルタイムに「主体以外の Agent」の書出を「書出入口」（知覚）できる必要がある。ただし、主体がリアルタイムに知覚できる必要がある対象は“書出”であり、「主体以外の Agent」が主体に対してリアルタイムに“対応”する必要はない。例えば、書出の共有場所を事前に作成しておき、主体が「書出行動」する前は、主体は「主体以外の Agent」による書出を知覚できない環境を情報システムによって構成することなどが選択肢となる。具体的な情報システムによる構成例は、次項で述べる。

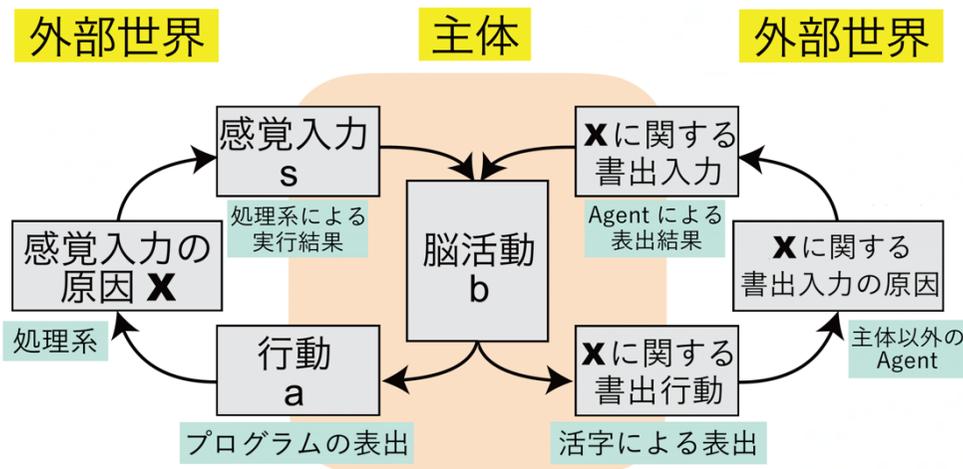


図2：提案するプログラミング学習環境の図式

提案環境の概観を図2に示す。提案環境では、「言語処理系を含む循環的因果性」と、「主体以外の Agent を含む循環的因果性」の二つが主体を介して結合され、それらの循環的因果性を構成・維持できるように、環境は主体に対して提示を与える。ここで、プログラミング学習環境の主たる目的は、「言語処理系」とのやりとりを通じた、実行可能なコンピュータプログラム設計・構築であることから、「主体以外の Agent」を含む循環的因果性の構成は、あくまでも「言語処理系を含む循環的因果性」の構成を目指すものであることに注意する。

続いて、提案環境のプロセスを表1に示す。本プロセスは12工程で形成され、環境から主体への行動がトリガーとなり、主体が「行動」し、その行動により主体に何らかの「知覚」がもたらされるという構造になっている。前半の1-6は「言語処理系を含む循環的因果性」を構成し、後半の7-12は「主体以外の Agent を含む循環的因果性」を構成する。

前半 1-6 は主に、「言語処理系を含む循環的因果性」を通して、図1における主体の脳活動のうち「生成モデル $p(x, s)$ 」の形成に寄与するプロセスである。1-2 と 3-4 のプロセスでは、具体的な「行動」と「知覚」の因果関係を主体が収集し、その情報に基づき、生成モデルを推論する（帰納的推論）。ただし、この帰納的推論は発展的推論であるため、形成したモデルの誤謬可能性を排除できない。そのため、5-6のプロセスにおいて、1-4で形成した生成モデルを利用した「行動」と「知覚」を誘導し、主体が立てた仮説について、「行動」し目的に沿った「知覚」をもたらすことができることを検証する（演繹的推論）。

後半 7-12 は、「主体以外の Agent を含む循環的因果性」を通して、図1における主体の脳活動のうち「現在の推測 $q(x)$ 」の形成に寄与するプロセスである。7-8 と 9-10 のプロセスで

表1：提案環境における循環的因果性を構成・維持するプロセス

No	行動者	行動内容	主体による行動内容が主体にもたらす知覚
1	環境 ^(*1)	行動 a を提示 (1)	-
2	主体	行動 a の表出 (1)	対応する感覚入力 s を知覚 (生成モデル p の推論)
3	環境 ^(*1)	行動 a を提示 (2)	-
4	主体	行動 a の表出 (2)	対応する感覚入力 s を知覚 (生成モデル p の推論)
5	環境 ^(*1)	感覚入力 s を提示	-
6	主体	行動 a を表出	対応する感覚入力 s を知覚 (生成モデル p の検証・再形成)
7	環境 ^(*1)	書出行動の提示	-
8	主体	活字による表出 ^(*2)	関連する書入出力を知覚 (現在の推測 q の推論)
9	環境 ^(*1)	相違点の表出を提示	-
10	主体	相違点の表出 ^(*2)	関連する書入出力を知覚 (現在の推測 q の推論)
11	環境 ^(*1)	問いの表出を提示	-
12	主体	問いの表出 ^(*2)	Agent による問いを知覚 (現在の推測 q の検証・再形成)

(*1)「環境」からの提示ができれば行動者に制限はない。例えば「資料」でも良い。

(*2)「主体以外の Agent」を含む循環内の表出では、「言語処理系」を含む循環の行動 (1-6) を再度行動することを妨げない。この行動により、表出内容を増補する可能性が高まる。

は、「生成モデル」に関する主体と主体以外の Agent の表出により、「現在の推測」を推論する（帰納的推論）。ただし、この帰納的推論もまた発展的推論であるため、「現在の推測」における誤謬可能性を排除できないことから、何かの「問い」を形成することが可能となる。11-12 のプロセスでは、7-10 のプロセスで生じた問いを表出し、「現在の推測」において問いが存在していることを明確にした上で、現在の推測の検証・再形成を行う。

なお、表1の(*2)に記載のあるように、7-12の「主体以外の Agent を含む循環的因果性」のプロセスにおいて、「言語処理系を含む循環的因果性」のプロセスを行動することは妨げない。これは、プログラミング学習環境の主たる目的が、「言語処理系」を通じた実行可能なコンピュータプログラムの設計・構築であるためであり、その行動が「主体以外の Agent を含む循環的因果性」のプロセスにおける表出の内容を増補する可能性を高めるため（換言すれば、情報の自由エネルギーを最小化する可能性があるため）である。

3. 1. 2. 学習環境の構築を支援する情報システムの一例

前項で述べたプログラミング学習環境の構築では、情報システムを活用することで循環的因果性の構築が簡便になる。本項では、環境の構築を支援する情報システムについて述べる。

前項のプログラミング学習環境では、「言語処理系を含む循環的因果性」と「主体以外の Agent を含む循環的因果性」の二つが主体を介して結合されている。「言語処理系を含む循環的因果性」では、主体が言語処理系に「行動」を行えることと、言語処理系から主体がその行動に対応する「知覚」を行えることに加え、表1で示した工程1・3・5を行う環境が必須となり、「主体以外の Agent を含む循環的因果性」では、Agent が参加できるシステムであり、主体が Agent に「書出行動」を行えることと、Agent から主体がその書出に対応する書入出力を「知覚」できること、そして表1で示した工程7・9・11を行う環境が必須となる。ここでは本研究グループが構築した

- (1) 表1の工程1・3・5・7・9・11を行うシステム
 - (2) 「言語処理系を含む循環的因果性」での行動と知覚を構成するシステム
 - (3) 「主体以外の Agent を含む循環的因果性」での行動と知覚を構成するシステム
- を一例として下記で紹介したい。

(1) 表1の工程1・3・5・7・9・11の提示するシステム

本システムは、表1で示した循環的因果性の構成・維持を全工程に渡って支援するシステムである。本研究グループでは、工程1・3・5を提示することで工程1-6までの循環的因果性の構成を支援するよう、プログラミング学習の資料の体裁を変更し、細分化している。具体的には、多くの内容において、工程1・3・5を1枚の資料（A4サイズのPDF1枚）にして主体に提供した。これにより、主体はその1枚で区切りとなっていることを容易に把握することができるだけで無く、工程1-6までがひとつの流れになっていることを認識できるようになる。

資料の細分化は、工程1-6と同様に、工程7-12においても実施し、「言語処理系を含む循環的因果性」を構成したのちに、「主体以外の Agent を含む循環的因果性」に移行できるよう

に配慮している。具体的には、工程 7-12 においても、工程 7・9・11 を 1 枚の資料 (A4 サイズの PDF 1 枚) にして主体に提供した。

なお、資料の配布方法においても「行動」と「知覚」の循環的因果性を意識させるようなシステムとしている。これは、主体のオンデマンドな「行動」に伴って、システムから資料を主体が「知覚」できる仕組みを構成したもので、本研究グループでは、Slack 内の Apps として、特定のメッセージを送付することにより、資料を受領できるアプリを開発した [5]。加えて、開発したアプリにより学習環境の管理者は、全ユーザ (主体とすべての Agent) が、どの内容の資料を、いつリクエストしたのかという情報を、リアルタイムに把握できる環境が構成されている。

(2) 「言語処理系を含む循環的因果性」での行動と知覚を構成するシステム

本システムは、(1) の資料の提示を踏まえ、主体が工程 2・4・6 を「行動」することを支援するシステムである。処理系への「行動」と実行結果としての「知覚」については、対話性のある多くの言語処理系がその機能を持っており、それを活用することができる。本研究グループでは、ブラウザ上で動作するオンラインアプリケーションとして、遠隔のサーバ内の言語処理系への「行動」と、同処理系からの「知覚」を行うことができるシステムを採用している [6]。この導入により、言語処理系の導入の手間が削減されるだけでなく、Agent 同士でコンピュータ環境が異なっても、言語処理系の環境を揃えることができる。なお、採用したシステムにより学習環境の管理者は、全ユーザが作成したすべてのプログラムに加え、その実行の時間、実行により得られた結果・エラーメッセージという情報を、リアルタイムに把握できる環境が構成されている。

(3) 「主体以外の Agent を含む循環的因果性」での行動と知覚を構成するシステム

本システムは、(2)での循環的因果性の構成を踏まえ、主体が工程 8・10・12 を「行動」することを支援するシステムである。主体の「書出行動」と主体以外の Agent からの「書出入力」については、「書出行動」の結果、「書出入力」ができるように構成する必要がある。主体と Agent が活字のやりとりを行うという意味では学習環境内 SNS など基本のシステムを構成可能であるが、因果関係を含めた構成には工夫が必要である。本研究グループでは、学習環境内 SNS として Slack を採用しており、因果関係を構成するため、Slack の Apps とプライベートチャンネルを組み合わせ、主体が特定のメッセージをアプリに送付することにより、そのプライベートチャンネルにメッセージが投稿される仕組みを構成している [5]。これにより、主体はメッセージを送付するまではプライベートチャンネル内のメッセージを閲覧できないが、その「行動」により、Agent からの「書出入力」を閲覧できるようになる。なお、採用したシステムにより学習環境の管理者は、全ユーザが作成したすべての書出行動に加え、その行動時刻という情報を、リアルタイムに把握できる環境が構成されている。

以上により、前項で述べたプログラミング学習環境は、全ユーザ (主体やすべての Agent) がそれぞれ独立に行動を起こしたとしても、「言語処理系を含む循環的因果性」と「主体以外の Agent を含む循環的因果性」の二つの循環的因果性を、理論上、独立に構成・維持できる。

また実践面から見ても学習環境の管理者は、(1)-(3)で述べたシステムを介すことで、全ユーザの情報をリアルタイムに把握可能であり、それらの情報をエビデンスとして、表1で示した循環的因果性の構成・維持が不十分なユーザへの対応に、資源を集中することができる。

3. 2. 提案環境を用いた実践

前節で述べた提案環境の具体的な実践として、次項で述べるワークショップを運営した。本節ではその詳細と得られたデータを紹介する。なお、同ワークショップは、明星大学を代表団体とする任意団体が主催し、明星大学のホームページや複数のメディアを通して参加者を募った [7]。ワークショップの運営は、明星大学情報学部内の演習科目「プロジェクト実習・実験1」の一環として、同学部の2年生11名が中心となり実施し、一部、必要に応じて上級生や社会人が参加した（その結果、1コースの運営は3名のチームで運営を行なった）。同団体の詳細は団体ホームページなどを参照されたい [8]。

3. 2. 1. ワークショップの概要

(1) ワークショップ開始前

次の情報をホームページやメディアで広報し、参加者を募った。

- ・名称 : 「はじめてのプログラミング」
- ・目的 : Python3 と C 言語を用いた初心者向けのプログラミング体験
- ・対象 : 言語処理系を用いたプログラミングを初めて行う方 (中学生～社会人)
- ・期間・回数・時間
 - 期間 : 2020年10月5日(月)～12月25日(金)
 - 回数 : 全5コース(月・火・水・木・金)、各コースは1週間に1回の全12回
 - 時間 : 1回90分
- ・方法 : オンライン実施(使用ソフトウェア : Slack [5] / Bit Arrow [6] / Zoom [9])
- ・参加費 : 無料
- ・上記以外に広報した内容
 - 参加者のプログラミング経験は問わないこと
 - 期間内であれば、参加はいつからでも可能であること
 - 12回の中に参加が難しい回があってもワークショップには参加可能なこと

(2) ワークショップ開催時

ワークショップは Zoom ミーティングルームで開催された。ただし、参加者の Zoom の顔出しは任意であった(その結果、運営者以外は、ほぼ顔出しをしていない)。参加者は、運営者が事前に用意した資料を、参加者自身のペースで進めた。この資料は、「行動」と「知覚」による循環的因果性の構成支援を目的としており、プログラミング文法の解説は記載されていない(具体的には、表1のプロセス1・3・5・7・9・11を行なっている)。また、運営者もプログラミング文法を解説する時間を設けない(ただし、質問には応じる)。実行環境として Bit Arrow の環境を採用したことで、参加者がブラウザを通して、コンピュータプログラム

の実行・保存・編集が行えるだけでなく、運営者がその様子をリアルタイムに確認することが可能となった。加えてSlackにて、気軽に参加者が運営者に質問や問合せができる環境を構成した（参加者には、他の参加者が閲覧可能なメッセージと、運営者のみが閲覧可能なメッセージを使い分けられるよう導入した）。

運営者の多くは、表1のプロセスを順に進めるよう参加者を誘導しつつ、参加者の心理的安全性を下げる声掛けを連続的に実施することを心掛けていた。また、参加者が表1のプロセスに慣れてきた時期を見計らって、工程11-12で参加者が作成した「問い」に運営者が取り組み、その様子を画面で共有していた（運営者が試行錯誤する姿を、意図的に参加者に共有した）。このように運営者は、参加者と運営者は双方が問いを出し合う関係性であることを全体に共有することで、「行動」と「知覚」の循環的因果性の構成・維持に取り組んでいた。

(3) ワークショップ開催外の時間帯

週1回のオンラインミーティングの時間以外にも、Bit ArrowとSlackは制限なく利用でき、参加者自身のペースで資料を進めることができる環境を構築した。

3. 2. 2. ワークショップの実践結果

本項では、前項で述べたワークショップにおける、具体的な実践結果について述べる。提案環境では、「言語処理系を含む循環的因果性」と、「主体以外のAgentを含む循環的因果性」の二つが主体を介して結合されており、それぞれの循環的因果性が構成されていることを得られたデータから確認する。なお、全12回を4分割した以下の各期間で確認を行う。

表2：期間名とその期間

期間名	期間
1Q：Week 01-03	2020年10月04日00:00～2020年10月24日23:59
2Q：Week 04-06	2020年10月25日00:00～2020年11月14日23:59
3Q：Week 07-09	2020年11月15日00:00～2020年12月05日23:59
4Q：Week 10-12	2020年12月06日00:00～2020年12月27日23:59

表3にワークショップ全体のデータを示す。なお、このデータは、表2の期間毎に50回以上の実行回数があったアカウントのみの数値を示している。ここで、表3の3～6行目の実行回数やエラー回数は「言語処理系を含む循環的因果性」に関するデータであり、7～8行目の書出行動数は「主体以外のAgentを含む循環的因果性」に関するデータであると言える。

3～6行目の実行回数やエラー回数は、図2の「行動」の回数を示したものであり、ワークショップにて文法などの解説をする時間を運営者が設けていないことを踏まえれば、参加者自身が循環的因果性を構成しているものと推測できる。また、1Qから4Qに時間が進むにつれて、実行回数・エラー回数の上下動はあるものの、4行目の1ユーザ当たりの実行回数が最も少ない4Qでも150回程度となっていることは、「言語処理系を含む循環的因果性」の維持についても、一定の効果があったと考えられる。

表3：ワークショップ全体のデータ

期間	1Q	2Q	3Q	4Q	1Qと4Qの 変化率
アクティブユーザ	116	124	104	71	61.2 %
実行回数合計	22808	29500	19993	10713	47.0 %
実行回数/User	196.6	237.9	192.2	150.9	76.8 %
エラー回数合計	5050	6289	4304	2066	40.9 %
エラー回数/User	43.5	50.7	41.4	29.1	66.9 %
書出行動数合計	2502	1283	1080	1062	42.4 %
書出行動数/User	21.6	10.3	10.4	15.0	69.4 %

また、7～8行目の書出行動数は、図2の「書出行動」の回数を示したものであり、「主体以外の Agent を含む循環的因果性」の構成に関するものである。7行目の書出行動数合計は1Qから4Qに時間が進むにつれて上下動があるが、最も少ない4Qでも1000回を超えていることが確認できる。これらの書出行動は、参加者同士で書出を共有することを前提とした上で「行動」しているものであるから、このデータは参加者が他者の書出を一定数「知覚」することができたことを示しており、循環的因果性の構成を支援していると考えられる。また、1ユーザ当たりの書出行動数にも上下動があるが、2Qと3Qで値がほとんど変化せず、4Qで上昇していることを踏まえれば、循環的因果性の維持についても効果があったと考えられる。

表4にコース毎のデータを示す。表3同様、期間毎に50回以上の実行回数があったアカウントのみを計測し、1ユーザあたりの実行・エラー回数を示した。この結果を見ると表3では見えなかったコース毎の特色があることに気づく。例えば、水曜コースの1Qと4Qの変化率を見ると共に100%を超えており、同コースの参加者一人一人を見ると、ワークショップが進むにつれて、実行回数を増やしていることが分かる（換言すれば、循環的因果性が高い水準で「維持できている」と言える）。また、月曜コースは、変化率がいずれも90%を超えていることに加え、一人当たりの実行回数が他の班と比べ高い水準で維持できていることが分かる（換言すれば、循環的因果性が高い水準で「構成できている」と言える）。これらの結果より、情報システムが循環的因果性の構成を支援しても、運営者がその構造を把握して運営しなければ、二つの循環的因果性の構成・維持が崩れる可能性があることが示唆されている。

4. 結論

本稿では、「行動」と「知覚」に基づく循環的因果性の構成を支援する仕組みを包含する、プログラミング学習環境の構築について、その提案内容の理論的概要と、その提案の具体的な実践の結果を示した。実践結果より、プログラミング文法の伝達を行わずとも、「行動」と「知覚」の循環的因果性の構成・維持を目的とすることで、外部環境としてのプログラミング（厳密には言語処理系）の学習を推進できることを示した。以後、今回の論文では明らかにできなかった本手法が持つ特徴についても、実証実験とともに明らかにしていきたい。

表4：コース毎のデータ

期間	各項目	1Q	2Q	3Q	4Q	1Qと4Qの 変化率
月曜 コース	アクティブユーザ	24	29	23	15	62.5 %
	実行回数/User	207.4	275.5	228.9	193.0	93.1 %
	エラー回数/User	41.5	59.3	49.7	41.7	100.5 %
火曜 コース	アクティブユーザ	22	20	21	18	81.8 %
	実行回数/User	172.3	300.7	202.2	149.7	86.9 %
	エラー回数/User	35.1	63.9	38.4	26.4	75.2 %
水曜 コース	アクティブユーザ	21	30	25	16	76.2 %
	実行回数/User	130.8	176.7	131.6	145.8	111.5 %
	エラー回数/User	24.0	34.6	25.3	29.4	122.5 %
木曜 コース	アクティブユーザ	16	16	11	11	68.8 %
	実行回数/User	245.9	179.6	193.6	135.6	55.1 %
	エラー回数/User	63.9	40.4	32.9	24.7	38.7 %
金曜 コース	アクティブユーザ	33	29	24	11	33.3 %
	実行回数/User	223.1	252.6	211.0	118.1	52.9 %
	エラー回数/User	53.2	55.3	56.6	20.2	38.0 %

謝辞

本研究は JSPS 科研費 19K03153 の助成を受けたものです。また、明星大学重点支援研究費 (2020)、明星大学教育新構想事業 (2021) の支援を受けております。

参考文献

- [1] 吉田 正俊, 田口 茂: 自由エネルギー原理と視覚的意識, 日本神経回路学会誌, 25, 3, pp. 53-70. (2018)
- [2] Friston K, Adams RA, Perrinet L, Breakspear M.: Perceptions as hypotheses: saccades as experiments. *Front Psychol.* 3:151. (2012)
- [3] 乾 敏郎: 自由エネルギー原理-環境との相即不離の主観理論-, 認知科学, 26, 3, pp. 366-386. (2019)
- [4] 乾 敏郎, 阪口 豊: 脳の大統一理論 自由エネルギー原理とはなにか, 岩波書店. (2020)
- [5] Slack Apps: <https://slack.com/intl/ja-jp/help/articles/360001537467> (2022年3月15日 閲覧)
- [6] Bit Arrow: <https://bitarrow.eplang.jp/> (2022年3月15日 閲覧)
- [7] COPERU Project: はじめてのプログラミング, <https://note.coperu.net> . (2022年3月14日 閲覧)
- [8] COPERU Project: <http://coperu.net> . (2022年3月15日 閲覧)
- [9] Zoom: <https://zoom.us> . (2022年3月15日 閲覧)