

共通テスト「情報」プログラミング問題の解決に必要な知識 —発話プロトコル分析—

寺尾敦^{*1}・秋岡敬子^{*1}・中村優花^{*1}
Email: atsushi@si.aoyama.ac.jp

*1: 青山学院大学社会情報学部

◎Key Words 大学入学共通テスト, プログラミング, 認知的課題分析, 発話プロトコル

1. はじめに

大学入学共通テストでは、令和7年度のテスト(2025年初頭に実施)より、「情報I」の出題が予定されている。

「情報I」の出題について、大学入試センターは、2021年3月24日に「サンプル問題」を、2022年11月9日に「試作問題」を公表した⁽¹⁾。いずれも、試験問題について具体的イメージを共有することを目的としている。

大学入学共通テストは「知識の理解の質を問う問題や、思考力、判断力、表現力等を発揮して解くことが求められる問題を重視した問題作成」を行うとされている⁽²⁾。しかし、思考力、判断力、表現力といった「力」のメタファーはマジックワードであり、教育や学習において有効に機能しない⁽³⁾。生徒から、大学入学共通テストでよい点が取れるようにするためにどうしたらよいかと相談されたとき、「思考力を身につけよう」では、生徒はどうしたらよいかかわからない。どのような知識が必要なのかを具体的に示す必要がある。

本研究では、共通テスト「試作問題」として公表されたプログラミング問題の認知的課題分析と、この問題を解決した大学生の発話プロトコル分析により、共通テストでのプログラミング問題で要求されている知識を明らかにする。認知的課題分析では、「試作問題」を初見で解いた大学教員の発話プロトコルをもとに、この問題の解決に必要な知識をプロダクション・ルールとして明らかにする。プロダクション・ルールは、手続き的知識の表現であり、IF-THEN形式のルールである。条件(IF)節は問題の状態を、行為節(THEN)はその状態で行うべき操作を表す。大学生の発話プロトコル分析では、「試作問題」の解決を試みた大学生の発話プロトコルに基づき、認知的課題分析の妥当性を確認する。すなわち、(1)大学生の問題解決過程は認知的課題分析によって提示されたプロダクション・ルールの使用を示唆するものであるか、(2)認知的課題分析でのプロダクション・ルールは教示に利用できるか、を検討した。

なお、認知的課題分析の結果は小規模な研究会において既発表⁽⁴⁾であるが、大学生の発話プロトコル分析は認知的課題分析の妥当性を確認するために行われており、大学生の発話プロトコル分析単独では論文を構成できない。そのため、既発表の内容を縮小したうえで、本論文の一部として含める。

2. 認知的課題分析

2.1 方法

大学入学共通テストの「試作問題」に含まれるプログラミングの問題では、買い物で客と店が交換する硬貨の枚数の最小値である「最小交換硬貨枚数」を求めるプログラムを作成する。最小交換硬貨枚数を求めるために、関数「枚数」を定義して利用する。この関数は、引数として金額が与えられると、ちょうどその金額となる硬貨の組み合わせの中で、枚数が最小となる硬貨枚数を返す。プログラミングを行う生徒と指導する先生の対話、および、プログラム中に[A]から[セ]まで14の空欄があり、空欄にあてはまる正解を選ぶ多肢選択の問題である。ただし、[ア]だけは数値を入れる。

認知科学を専門とする大学教員1名(本論文の第1著者)が、PC画面に提示した「試作問題」問題を発話思考で(つまり、意識内容を発話しながら)解決した。問題解決中のPC画面と発話は動画配信のためのソフトウェアで収録された。

収録された動画はYouTubeで公開されている⁽⁵⁾。問題解決過程の発話はすべて文字おこしされており、動画とともに表示することができる。

2.2 結果

得られた発話プロトコルから、このプログラミング問題の解決に使われた手続き的知識を推測し、それらをプロダクション・ルールの形式で表現した。問題解決に使われたと推測されるプロダクション・ルールは、日常生活に由来するルール、プログラミングでの領域固有のルール、汎用的な推論ルール、に大別することができた。

日常生活に由来するルール:関数「枚数」は、引数として金額が与えられると、ちょうどその金額となる硬貨の組み合わせの中で、最小となる枚数を返す。空欄[A]では、入力が46のときの、関数「枚数」の戻り値が問われている。この正解(6)は問題での生徒と先生の会話で明示的に示されているが、そうでなくても、日常生活で獲得されていると考えられるプロダクション・ルールにより、容易に知ることができる。このプロダクション・ルールを表1に示す。P1aから始まり、P1bが繰り返し使用されて、最後にP1cによって結果が返される。

プログラミング固有のルール:「試作問題」では、与えられた金額ちょうどを支払う最小の硬貨枚数を求めるプログラムが示されている(図1)。 $[キ]$ から $[コ]$ の空欄にあてはまるものを選択肢から選ぶ。

表1 日常生活由来のプロダクション・ルール例

P1a: 最小硬貨枚数の計算開始	
IF	目的が金額 =M を支払う最小硬貨枚数を知ることであり、 注目している硬貨がないならば、
THEN	残金を =M として、 額面 =X が最も大きい硬貨に注目せよ
P1b: 各硬貨の枚数	
IF	目的が金額 =M を支払う最小硬貨枚数を知ることであり、 残金 =R が 0 でなく、 額面 =X の硬貨に注目しているならば、
THEN	=R / =X の商 =P をその硬貨の枚数として、 残金を余り =Q に更新し、 =X 以下で最大の額面 =Y の硬貨に注目せよ
P1c: 最小硬貨枚数	
IF	目的が金額 =M を支払う最小硬貨枚数を知ることであり、 残金 =R の値が 0 ならば、
THEN	各硬貨の枚数の合計を最小硬貨枚数とせよ

Note. 等号のついた文字は変数を表す

```

(1) Kouka = [1,5,10,50,100]
(2) kingaku = 46
(3) maisu = 0, nokori = kingaku
(4) i を キ ながら繰り返す:
(5) | maisu = ク + ケ
(6) | nokori = コ
(7) 表示する(maisu)
    
```

図1 与えられた金額をちょうど支払う最小の硬貨枚数を求めるプログラム

このプログラムを書くとき、表1に示したプロダクション・ルールは実質的にその書き方を示している。ルール P1a は、最初に配列 Kouka の最後の要素である 100 に注目することを示唆している。ルール P1b は注目している硬貨を変更しながら繰り返し使われるので、プログラムでは何らかの繰り返し処理を行うことになる。繰り返しのたびに残金(プログラムでは変数 nokori)を更新する点も同じである。最小の硬貨枚数の計算方法は、表1のプロダクション・ルールと図1のプログラムで、多少の違いがある。表1のプロダクション・ルールでは各硬貨の枚数を記憶しておいて最後に加えるが、図1のプログラムでは繰り返しのたびに枚数を更新している。もちろん、各硬貨の枚数を記録する配列を使えば、ループを抜けた後で各硬貨の枚数を合計するようにプログラムを書き換えることは容易である。表1のプロダクション・ルールと図1のプログラムでのもうひとつの違いは、残金(nokori)の値が0になっても処理を終了しないことである。

表2 プログラミング固有のプロダクション・ルール例

P2a: 配列の最終要素からのループ構成	
IF	=K 個の要素が配列 =A に保存されており、 配列 =A の要素を最後から順に処理したくて、 配列の添字が 0 から始まっており、 ループを制御する変数が =I ならば、
THEN	=I を (=K)-1 から 0 まで 1 ずつ減らすループを構成せよ
P2b: 各硬貨の枚数 (プログラミング版)	
IF	目的が金額 =M を支払う最小硬貨枚数を知ることであり、 残金が =R、現在までの硬貨の枚数は =N で、 硬貨の額面は配列 =A に記録され、 配列中の要素 =X に注目しており、 繰り返し処理で変数の値を更新するならば、
THEN	=R / =X の商 =P をその硬貨の枚数として、 硬貨の枚数 =N を =N + =P に更新し、 残金を余り =Q に更新せよ
P2c: 最小硬貨枚数 (プログラミング版)	
IF	目的が金額 =A を支払う最小硬貨枚数を知ることであり、 繰り返し処理で変数の値を更新し、 硬貨の枚数 =N はループ中で更新されており、 繰り返しは終了しているならば、
THEN	最小硬貨枚数を =N として返せ

Note. 等号のついた文字は変数を表す

図1のプログラムを完成させるプロダクション・ルールを表2に示す。ルール P2b は P1b の、ルール P2c は P1c のプログラミングバージョンである。ルール P2c はプログラムの 7 行目を書くためのものであり、この問題への解答(空欄 [キ] から [コ] を埋める)では利用されない。

汎用的推論のルール: この問題の核心は、商品の価格と釣り銭を足した金額の最小硬貨枚数と、釣り銭の最小硬貨枚数の和が、「最小交換硬貨枚数」になるということである。この関係は繰り返し問われている。問題での生徒と先生の会話では、商品価格が 46 円のときの最小交換硬貨枚数が「枚数(51)+ 枚数(5)」となることが述べられる。これは空欄 [ア] の解答となっている。直後に、商品価格を x 円、釣り銭を y 円としたときの、最小交換硬貨枚数の計算式が問われる(空欄 [ウ] および [エ])。これは「枚数(x+y)+ 枚数(y)」である。最後に、商品価格の変数 kakaku と、釣り銭の変数 tsuri から、変数 shiharai = kakaku + tsuri が作られたとき、最小交換硬貨枚数が「枚数(shiharai)+ 枚数(tsuri)」であることが問われる(空欄 [ス] および [セ])。

表3 汎用定期的推論のプロダクション・ルール例

P3: 集合間の対応関係による式構成

IF

集合=A の要素と集合=B の要素に対応関係があり、集合=A の要素についての式が存在するならば、

THEN

その式の要素を集合=B での対応する要素に置き換えた式を構成せよ

Note. 等号のついた文字は変数を表す

問題解決を行った大学教員は、この関係を最後に問われたときの解答が、それよりも前に問われたときの解答と整合しているかを確認していた。変数の対応関係は、 x と $kakaku$ 、 y と $tsuri$ であり、さらに $x+y$ と $shiharai$ が対応している。したがって、 x 、 y 、 $x+y$ を用いた式は、 $kakaku$ 、 $tsuri$ 、 $shiharai$ を用いた式に書き換えられ、対応する変数は同じ位置を占めなければならない。大学教員はすでに得た解答の確認を行っていたが、まだ解答が得られていないときにこうした対応関係を利用することも可能である。表3のプロダクション・ルールは、解の確認ではなく、発見のためのルールである。

3. 発話プロトコル分析

大学入学共通テストの「試作問題」として公表されたプログラミング問題の認知的課題分析により、この問題の解決に必要な知識をプロダクション・ルールとして記述することができた。しかし、この分析は2つの点で不十分であり、分析の妥当性を確認する必要がある。

第1に、実際の受験生の多くは、前節で記述したプロダクション・ルールとは異なる知識を使うかもしれない。もちろん、別解が存在することはかまわない。しかし、われわれの分析があまり一般的ではない解き方を示しているのであれば、もっと一般的な解き方と、そこで使われる知識を示すべきだろう。

第2に、われわれが示した知識は教示に利用できないかもしれない。われわれの認知的課題分析の目的は、「思考力」というマジックワードに代えて、生徒が獲得すべき知識を具体的に示すことである。したがって、示された知識は教示や学習に役立つものでなくてはならない。しかし、課題分析だけでは、これは不明である。

そこで、大学生に「試作問題」を発話思考法で解いてもらい、その発話のプロトコル分析から以上2つの問題点を解決した。具体的には、(1) 問題解決に成功したときの発話プロトコルが、認知的課題分析によって提示されたプロダクション・ルールの使用を示唆するものであるか、(2) 問題解決に失敗したとき、認知的課題分析でのプロダクション・ルールに基づいたヒントを提示することは問題解決を助けるか、を検討した。大学生は大学入学共通テストの受験者とは異なるが、プログラミング課題の解決において、実際の受験者と大きな違いは生じないだろう。

3.1 方法

12名の大学生が1人ずつ単独で実験に参加した。発話思考について説明を受けたのち、PC画面に提示された「試作問題」問題を発話思考で解決した。発話思考について

説明では、認知的課題分析のために行われた大学教員の発話思考での、空欄 [ア] への解答部分の動画が具体例として提示された。問題解決中のPC画面と参加者の発話は動画配信のためのソフトウェアで収録された。

設問に対して回答を与えることができなかった場合には、認知的課題分析に基づいて作成されたヒントが与えられた。設問に対して何らかの回答が与えられた場合には、それが不正解であっても、すべての設問(空欄の [ア] から [カ]) に回答を終えるまでフィードバックは与えられなかった。すべての設問への回答がなされた後で、不正解だった設問が指摘され、もう一度解決が試みられた。必要に応じてヒントが与えられた。

発話がほとんどなされなかったことにより、2名のデータは分析に用いることができなかった。

3.2 問題解決に使われた知識

設問に正解が与えられたとき、先に行われた認知的課題分析で得られたプロダクション・ルールによるものとは異なる解決を示唆する発話プロトコルは、問題中の1か所の例外を除き得られなかった。このことは、最初の認知的課題分析が妥当であったことを示している。もちろん別解が存在する可能性はあるが、10名の発話プロトコルが課題分析の結果と矛盾しないのだから、われわれの分析は一般的な問題解決方法を示していると言える。

1か所の例外とは、この問題の最後にある空欄 [ス] および [セ] である。ここでは、商品価格の変数 $kakaku$ と、釣り銭の変数 $tsuri$ から、変数 $shiharai = kakaku + tsuri$ が作られたとき、最小交換硬貨枚数が「枚数($shiharai$) + 枚数($tsuri$)」であることが問われている。大学教員は、ここよりも前の設問との対応関係に気がつき、類推によって正解を得ていた。これに対して、大学生の問題解決者はこの対応関係に気がつかず、あらためて考えて正解を得ていた。大学教員と、大学生の1人による解決プロトコルを表4に示す。

表4 類推による解決と類推でない解決のプロトコル
類推による解決プロトコル (大学教員)

そんでさっきと矛盾してないはずだよ。これですよ。えっと、[ウ] と [エ] は (2) 番と (1) 番だから、 $x + y$ と y でしょ。そうでしょ。だから、価格と釣り銭足したやつと釣り銭だよ。そうだよ。大丈夫ですね。だからここは、えーと、 $kakaku$ と $tsuri$ 銭を足したやつは $shiharai$ になってるから、 $shiharai$ と、 $shiharai$ ってる (0) 番と、それから次の (2) 番だよ。

類推でない解決プロトコル (大学生)

今回せつかくなんで51で考えてみると、 $kakaku$ が46で、 $tsuri$ が5で、えっと、ま、 $shiharai$ には51が入っているという状況ですね。はい。で、 $maisuu$ はさっきのこれ関数を使うんですよ。 $kakaku$ と $tsuri$ 、え、ちゃうわ。お釣りと支払った合計とそのお釣りの枚数をそれぞれ知りたいので、・・・(中略)・・・

えっと「枚数 ($shiharai$)」と、えっと「枚数 ($tsuri$)」を知りたいので、(0) と (2) 番。(0) 番と (2) 番が [ス] [セ] どちらかそれぞれに入ります。

表5 ループでの変数の値更新のヒント

1 : プログラミングでは、等号は等しいという意味ではなく、左辺の値を右辺に代入するという意味です。プログラムの (4) 行目から (6) 行目が繰り返されるたびに、変数 maisu と変数 nokori の値が更新されていきます。
2 : 繰り返しが終わったとき、maisu と nokori には何が保存されていますか？ これら変数の、最初の値は何ですか？ 繰り返しのたびに、これら変数の値をどのように更新すればよいですか？
3 : i が 4 のとき、プログラムの (4) 行目と (5) 行目では、どのような処理が行われますか？ (わからなければ) 46 円を支払うのに、100 円玉が何枚使えるかを計算して、maisu と nokori の値を更新します。これら変数の値はどのように更新されますか？

表6 ヒント提示後の問題解決プロトコルの例

これ " = " じゃねえんだ。あ、それ当たんねえわ。

(中略)

「プログラミングの (4) 行目から (6) 行目が繰り返されるたびに、変数mais�と変数nokoriの値が・・・」なるほどね。maisûを右辺に代入するから、だから右辺にmaisûがなきゃだめなんだ、これ。あ、そりゃ違うわ。だから [ク] 絶対maisûだ。

3.3 プロダクション・ルールに基づく教示の有効性

認知的課題分析によって得られたプロダクション・ルールが妥当である (つまり、獲得すべき知識を正確に表現できている) ならば、自力での問題解決に失敗したとき、プロダクション・ルールに基づいた教示を行うことが解決の助けとなるはずである。こうした考えは知的チューターシステム (Intelligent Tutoring System) が学習者に与える教示やヒントに使われてきた⁶⁾。教示やヒントが有効ならば、それは基盤となっている認知的課題分析の妥当性を示す証拠となる。

図1は変数 maisu の値をループで更新する処理である。ここでは、4名の大学生が解決に行き詰まってヒントを必要とした。実験参加者がここで解決に行き詰まったときには、表5に示すヒントを順に出すことにしていた。ここでのヒントは6段階から構成されており (表5に示されているのは最初の3段階)、ある段階のヒントで問題解決ができなければ、次の段階のヒントを出した。初期の段階でのヒントは問題状況 (つまり、プロダクション・ルール条件節 (IF) に焦点を当て、後の段階のヒントほどルールの行為節 (THEN) に焦点を当てている。

解決に行き詰まった実験参加者の一人が、表5に示された第1段階のヒントを提示された後の、問題解決プロトコルを表6に示す。この学生は、変数の値の更新を行うための代入を、数学での等号の意味に誤って理解していた。ヒントによってこの誤りを正すことで、表2に示した変数の値更新のためのプロダクション・ルールが使用できる状況であることに気がつき、正解を与えることができたと考えられる。最初の2段階のヒントで、どの参加者も正解を得ることができた。

4. おわりに

本研究では、大学入学共通テスト「情報I」のプログラミングの「試作問題」の解決に必要な知識を、プロダクション・ルールの形で明らかにした。これらプロダクション・ルールは、常生活に由来するルール、プログラミングでの領域固有のルール、汎用的な推論ルール、に大別することができた。

これらプロダクション・ルールを見ると、共通テストが測定すると称している「思考力」「判断力」「表現力」は、それほど複雑ではない手続き的知識の適用であることが示唆される。これらプロダクション・ルールは、いずれも日常生活あるいは授業で獲得が期待できるものであり、獲得が非常に困難と思われるものはない。本研究は「試作問題」だけを扱ったが、実際の試験で今後の出題される問題でも、この点について大きな違いはないだろう。

残された大きな問題のひとつは、必要な知識をどのようにして獲得できるか (あるいは、問題解決の場で即興的に生成できるか) ということである。特に難しいと考えられるのは、日常生活に由来する知識の獲得あるいは生成である。たとえば、「最小硬貨交換枚数」で買い物を実行することは容易だが、それを行うためにどのような計算を行っているのかを明確に記述できなければ、プログラムを書くことはできない。プログラミングの学習において、あまり意識しないで実行できている日常的行為のプロセスを明示的に記述して、プログラムに翻訳するという練習が重要かもしれない。

謝辞

本研究は JSPS 科研費 JP23K02718 の助成を受けたものです。

参考文献

- (1) 大学入試センター：“(参考) 試作問題等令和4年度までの検討状況”，https://www.dnc.ac.jp/kyotsu/shiken_jouhou/r7/r7_kentoujoukyou/ (2024年6月30日アクセス)
- (2) 大学入試センター：“共通テストの役割”，https://www.dnc.ac.jp/kyotsu/shiken_gaiyou/yakuwari.html (2024年6月30日アクセス)
- (3) 鈴木宏昭：“私たちはどう学んでいるのか：創発から見る認知の変化” 筑摩書房 (2022)。
- (4) 寺尾敦，秋岡敬子，中村優花：“共通テスト「情報」のプログラミング問題の解決に必要な知識”，情報コミュニケーション学会第34回研究会，(2023)。
- (5) 寺尾敦：“大学入学共通テスト「情報I」試作問題 (プログラミング) はこうやって考える”，<https://youtu.be/rqxm2KcHQwY> (2024年6月30日アクセス)
- (6) Koedinger, K. R. & Corbet, A.: “Cognitive Tutor テクノロジーが教室に学習科学を実現する”，R. K. ソーヤー (編)，『学習科学ハンドブック』，培風館，pp.53-65, (2009)。