

Peirce の探究段階論に基づく「情報 I, II」における単元間の構造分析

大西 洋
関西大学

saireya@gmail.com

共通教科「情報」では、各科目で導入の単元が設けられ、他の単元で個別の問題解決方法を学ぶ。だが、各単元の内容面での関連性が明確でないため、多様な問題解決の方法のうち、これらを扱う合理的な説明に乏しい。本稿では、Peirce の推論分類と探究段階に関する理論に基づいて各単元の特徴を整理し、共通教科「情報」の授業を体系的に展開する指針を提案する。

1. はじめに

新学習指導要領⁽¹⁾では、共通教科「情報」に「情報 I」「情報 II」の 2 科目が設置される。両科目の単元のうち、「情報 I」の単元「情報社会の問題解決」と「情報 II」の単元「情報社会の進展と情報技術」は「科目の導入として位置づけ」るものとされ、「情報 II」の単元「情報と情報技術を活用した問題発見・解決の探究」は「科目のまとめとして位置づけ」るものとされている。

図 1 に、学習指導要領に示された「情報 I, II」の単元の関係性を示す。図 1 にあるように、前述の 3 単元以外の各単元は、「問題の発見・解決を行う学習活動」に対応するものであり、それぞれ異なる問題解決の方法を扱う。両科目で「情報デザイン」「プログラミング」「データの活用」の 3 種の問題解決方法を扱うことで、教科内での内容の系統性が確保されている。

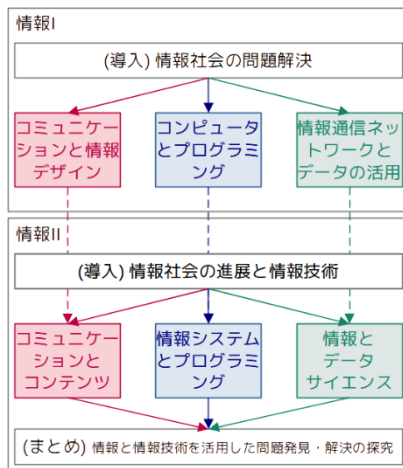


図 1 「情報 I, II」の単元の関係性

だが、問題解決方法は、これら以外にも多数ある。そのため、共通教科「情報」の授業を実施するにあたり、次の問題が生じる危険性がある。

- 多様な問題解決方法のうち、共通教科「情報」で上記の 3 種類を扱う妥当性が明らかでない。
- 上記の各方法間の関連性が明らかでない。

そこで本稿では、Peirce による推論分類と探究段階論に基づき、共通教科「情報」で扱う問題解決方法の妥当性と、各方法間の関連性を明らかにし、各単元を授業で扱う際の指針を示す。

2. 先行研究

2.1 問題と問題解決

Simon⁽²⁾⁽³⁾⁽⁴⁾は、問題(problem)を「現状(As-Is)と目標(To-Be)」の間の差異」と定義した。現状から目標へ至る経路を問題の解(solution)とし、問題の解を発見することを問題解決(problem solving)と呼んだ。Simon による問題解決の概念を図 2 に示す。

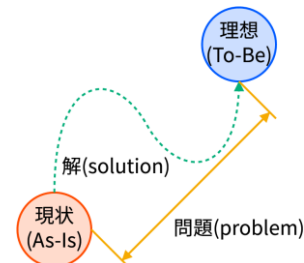


図 2 Simon による「問題」と「解」の概念

2.2 Peirce の推論分類

Peirce⁽⁵⁾⁽⁶⁾は、論理学で扱う推論について、事例(case)・規則(rule)・結果(result)の 3 つの命題のいずれかを推論するかにより、推論を表 1 の 3 種に分類した。

表 1 Peirce による推論の分類

推論の種類	前提	結論
アブダクション (abduction)	規則と結果	事例
演繹 (deduction)	事例と規則	結果
帰納 (induction)	結果と事例	規則

表 1 の 3 種の推論のうち、演繹のみが、前提が真なら結論が真だといえるため、厳密性がある推論である。帰納とアブダクションは、厳密性がない一方で、前提に含まれない一般化した結論や新たな発想を得られる点で、拡張性がある推論である。

2.3 Peirce の探究段階論

Peirce⁽⁷⁾⁽⁸⁾は 2.2 節の推論分類を基に、探究(inquiry)活動の段階を、次のように設定した⁽⁸⁾。

- アブダクションによる仮説の考案
既知の知識(規則)と観察した現象(結果)から、現象を説明する仮説(事例)を推論する
- 演繹による仮説の検証可能化
考案した仮説(事例)と既知の知識(規則)から、観察可能な現象(結果)を推論する
- 帰納による仮説の実証

実験で観察した現象(結果)と実験に用いた対象(事例)から、仮説(規則)の正しさを推論する

3. 共通教科「情報」で扱う問題解決方法

共通教科「情報」で扱う問題解決を Peirce の探究段階に当てはめると、表 2 のようになる。

3.1 情報デザイン

Rowe⁽¹⁰⁾は建築を例に、アブダクションは「デザインにおいて、極めて一般的なこと」と指摘した。また吉川⁽¹¹⁾は、「それ(デザイン)を誤りの可能性のあるアブダクションによって提案し、それを検証することによって実在のものとする」ことが人類の歴史だと述べた。

このように、情報デザインによる問題解決の過程では、アブダクションの段階が特に重要である。

3.2 プログラミング

文科省⁽¹²⁾が初等教育段階でのプログラミング教育の目標として掲げる「プログラミング的思考」は、中等教育にも準用される。「プログラミング的思考」は、「自分が意図する一連の活動を実現するために、(中略)論理的に考えていく力」と定義される。ここでいう「論理」は演繹推論を指している。

このように、プログラミングによる問題解決の過程では、演繹の段階が特に重要である。

3.3 データの活用

データの活用で用いる統計学は、真であることの証明が困難な対象に対し、Hume⁽¹³⁾による自然の斉一性原理に依拠して、帰納推論の妥当性を示すことを目的とする⁽¹⁴⁾。

このように、データの活用による問題解決の過程では、帰納の段階が特に重要である。

4. おわりに

本稿では、Peirce の探究段階論と新学習指導要領での共通教科「情報」の各単元の関連性を指摘し、各単元で焦点化する内容や、授業全体で取り扱う際の指針を示した。今後の課題には、本稿の指針に基づく授業の実践や、各単元の実習で扱う教材の開発が挙げられる。

参考文献

(1) 文部科学省: “高等学校学習指導要領”, p.190-195 (2018).
https://www.mext.go.jp/content/1384661_6_1_3.pdf
 (2) Herbert A. Simon: “the new science of

management decision”, p.70, Prentice Hall (1977), ISBN: 0136161367.
 (3) Herbert A. Simon: “意思決定の科学”, p.95-96, 産業能率大学出版部 (1979), 稲葉元吉・倉井武夫訳.
 (4) Herbert A. Simon: “システムの科学”, パーソナルメディア (1999), ISBN:489362167X.
 (5) Charles Sanders Peirce: “Deduction, Induction and Hypothesis”, Collected Papers of Charles Sanders Peirce vol.2: Elements of Logic, p.372-375 (CP2.619-624) (1878).
 (6) Charles Sanders Peirce: “Lessons from the History of Science”, Collected Papers of Charles Sanders Peirce vol.1: Principles of Philosophy, p.28-29 (CP1.65-68) (1896?).
 (7) Charles Sanders Peirce: “Reality of God”, Collected Papers of Charles Sanders Peirce vol. 6: Scientific Metaphysics, p.319-323 (CP6.467-473) (1908).
 (8) 米盛 裕二: “アブダクション——仮説と発見の論理”, p.103-105, 勁草書房 (2007), ISBN: 4326153938.
 (9) Charles Sanders Peirce: “The Fixation of Belief”, Collected Papers of Charles Sanders Peirce vol. 5: Pragmatism and Pragmaticism, p.229-232 (CP5.370-375) (1877).
 (10) Peter G. Rowe: “デザインの思考過程”, p.121-123, 鹿島出版会 (1990), ISBN: 430606106X, 奥山健二訳.
 (11) 吉川 弘之: “一般デザイン学”, p.250-253, 岩波書店 (2020), ISBN: 4000050621.
 (12) 文部科学省: “小学校段階におけるプログラミング教育の在り方について” (2016).
https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm
 (13) David Hume: “人間本性論 第1巻: 知性について”, p.108-111, 法政大学出版局 (2011), ISBN: 4588120808, 木曾好能訳.
 (14) 大塚 淳: “統計学を哲学する”, p.19-22,33,43, 名古屋大学出版会(2020), ISBN:4815810036.

表 2 Peirce の探究段階論に基づく問題解決過程の例

	情報デザイン	プログラミング	データの活用
(1) アブダクション	<u>デザインで解決すべき問題を決定する</u>	プログラムで解決すべき問題を決定する	データから導き出したい仮説を決定する
(2) 演繹	デザインの形態や表現を検討し、実際に作成する	<u>プログラムを作成し、意図する動作になるまで修正する</u>	仮説を示すための統計量を計算する
(3) 帰納	設計したデザインが正しく使われたことを確認する	プログラムが正しく動作したことを確認する	<u>仮説検定や推定などの統計手法で仮説の正しさを示す</u>