

小学校プログラミング教育を見据えた 算数科指導の指針を得るための一考察

A study of a mathematic instructions guideline
in anticipating elementary programming education

近藤 一貴, 加藤 直樹

Kazuki KONDO and Naoki KATO

東京学芸大学教職大学院, 東京学芸大学

Tokyo Gakugei University

〈あらまし〉 2020年度より小学校においてプログラミング教育が始まった。このねらいの一つに、教科学習との結びつきがある。本稿ではプログラミングにおいて多く用いられる条件分岐の条件式に等号・不等号と変数が用いられていることに着目し、これらが扱われている算数教育とプログラミングでの扱われ方や考え方の相違点について考察を行った。その結果、算数教育とプログラミング教育がうまく結びつかない可能性が示唆された。

〈キーワード〉 小学校教育 プログラミング教育 教科教育 算数教育 情報教育

1. はじめに

2020年度から小学校においてプログラミング教育が必修化された。このプログラミング教育では「教科等で学ぶ知識及び技能等をより確実に身に付けさせること」をねらいとしていることから、教科学習との結びつきを考慮する必要がある。

その一つに、変数と等号・不等号を用いる条件分岐の条件式と算数科の結びつきがある。しかし、小学校算数科での学習内容とプログラミングで求められる条件式の立式には考え方が異なる。

本稿では、プログラミング教育を見据えた算数科指導の指針を得ることを目標に、主に算数教育とプログラミング教育における変数と不等号の扱い方や考え方の相違点を考察する。

2. プログラミングで扱われる条件式

条件式は大きく範囲を伴うもの ($n < 10$, $1 < n < 10$ など)、範囲を伴わないもの ($n = 10$, $n \% 2 = 0$ など) の2種類に分けられる。その特徴について考察を行う。

2.1 範囲を伴う条件式

範囲を伴う条件式は、不等式によって条件式に役割を付与している。例えば「 $1 < n < 10$ 」という条件式であれば、条件式は「1より大きく10より小さい数のとき」にある処理を行わせる役割を付与されていることになる。この役割はプログラムに行わせたい処理の必要性から導かれる。

また、範囲を伴う条件式には2種類の表し方がある。例えば、 n が整数であるとき、「 $0 < n < 10$ のとき」 n を発音する(内包的記法)ことと「 $n = 1, 2, 3, 4, 5, 6, 7, 8, 9$ のとき」 n を発音する(外延的記法)ことは、その意味することは全く同じである。しかし、後者のような記法をプログラミングで用いることはほとんどない。す

なわち、範囲を伴う条件式の立式に当たっては、ある範囲に収まっている数を数の集合と捉え、その性質を抽象化し内包的記法に落とし込む能力が必要になる。

2.2 範囲を伴わない条件式

範囲を伴わない条件式は、等値比較によって条件式に役割を付与している。「 $n = 10$ 」のような単純な等値比較であれば、for文で用いられる「～回まで繰り返す」の終了条件、すなわちカウンタとしての役割が付与されている。「 $n \% 2 = 0$ 」のように変数に演算を加えたうえでの等値比較は、ある値が処理の条件となる性質を備えているか(今回であれば偶数かどうか)を判定する役割が条件式に付与されている。

どちらも範囲を伴う条件式と同様に、プログラムに行わせたい処理の必要性から役割は導かれ、前者は終了条件以外の変数が取り得る値を集合として捉えられる必要があり、後者は条件式に当てはまる数全体を集合として捉えられる必要がある。したがって、範囲を伴わない条件式の立式に当たっては、範囲を伴う条件式と同様に数の集合の性質を抽象化する能力が必要になっている。

3. 算数科における不等号と変数の扱い

3.1 不等号の指導

算数科において不等号を扱うのは第2学年である。第2学年では、「 $18 < 34$ 」のような具体的な数同士の大小比較で扱われており、その後の学年でも同様である。

3.2 変数の指導

プログラミングで扱われるような文字としての変数は第6学年で扱われるが、それ以前に言葉の式や Δ , \square などの記号が変数に近いものとして扱われている。

また、平林らは変数の意味を

(1) 数の抽象化、一般化としての変数

- (2) 未知数としての変数
- (3) 変わるもの（変量）としての変数
- (4) 元を代入し得る空所としての変数
- (5) 任意の元の代表としての変数

（原文では「変数」を「文字」としていたが、文意として「変数」でも差支えがないため「変数」とした）の5つに分類している[1].

4. 算数科とプログラミングにおける差異

4.1 不等号について

不等号については、算数科では、前節で述べたように具体的な数同士の大小比較の表記で活用されるに留まっており、プログラミングで用いるような不等式を用いた数の集合の性質を抽象化して表現する扱いは学習されていない。このことから、算数科の学びとプログラミングがうまく結びつかないと考えられる。

4.2 変数における差異

4.2.1 「数の抽象化・一般化としての変数」における算数科とプログラミングの差異

2.2, 2.3 で述べたように条件式の立式には数の集合の性質を抽象化が必要になってくる。これは変数の意味では「(1) 数の抽象化、一般化としての変数」にあたり、第2, 4, 5学年で学習されている。

しかし、第2学年では変数を用いた立式までは扱っていない。また、第4学年において公式を作り上げる過程で、いくつかの数量の組から関数関係を見出し、言葉の式として抽象化・一般化することを学習する。これは具体的な数の組を作り一般的な性質を導くことから帰納的な思考に分類される。だが、プログラミングにおける条件式の立式は帰納的な思考だけではなく、演繹的な思考を基に行われる場合もあると考えられる。なぜなら、条件式に「特定の場合においてある処理を行わせる」役割を持たせるために、数の集合の性質を内包的記法で表現する必要があるからである。

また、第5学年において伴って変わる二つの数量を表やグラフから読み取り、□、△や言葉の式を用いて一般的な式に表す学習を行う。これも、第4学年のときと同じく帰納的な思考によって関数関係を抽象化・一般化しているのであって、プログラミングにおける演繹的な思考による内包的記法による数の集合の性質の表現は学ばれていない。

以上から、「(1)数の抽象化・一般化としての変数」は算数科の中で学ばれてきてはいるが、それを「特定の場合においてある処理を行わせる」役割を持たせるために数の集合の性質を内包的記法で表現する目的で扱っている単元は存在していない。このことから、算数科の学びとプログラミングがうまく結びつかないと考えられる。

4.2.2 「変わるもの（変量）としての変数」における算数科とプログラミングの差異

センサーを活用したプログラミング活動を行うとき、条件式はセンサーから入力される値を用いて制御す

る役割をもつ。例えば、「もし（光センサーの値）<80なら ライトをつける」のようなものである。この時、センサーの値を示す変数は「変わるもの（変量）としての変数」である。なぜなら、センサーから入力される値はプログラム自身ではなく、その他の外的要因により変化するからである。この意味の変数は算数科において第4から6学年で扱われている。

まず、第4学年の算数科では、長方形の求積公式などの公式を作り出すことや公式を扱うことを学習する。その中で、ある数量同士に関数関係を見出し、言葉によって立式し、それが一般的に成り立つことを認められたとき、言葉の式を用いて公式として表現する。

しかし、センサーの値を用いた条件式は、数量同士に関数関係を見出した結果立式されるものではなく、「ある数量がある値になったときにどうするか」という役割を持ったものである。この点において、「(3)変わるもの（変量）としての変数」自体は学習されているが、その活用方法が算数科の学習とプログラミングにおいて異なっていることがわかる。

第5学年では、表やグラフから見出した関数関係を□や△などの記号を用いて表すことを学習するが、第4学年と同様に、その活用方法が算数科の学習とプログラミングにおいて異なっている。

第6学年では、□や△などの代わりに a, x などの文字を使うことで、関数関係を式に表せることを学習するが、これも第4,5学年と同様に、その活用方法が算数科の学習とプログラミングにおいて異なっている。

以上から、「(3)変わるもの（変量）としての変数」は算数科の中で扱っているが、プログラミングにおける「ある数量がある値になったときにどうするか」という役割を持たせる立式は扱っていない。このことから、算数科の学びとプログラミングがうまく結びつかないと考えられる。

5. おわりに

本稿では、プログラミング教育を見据えた算数科指導の指針を得ることを目標に、主に算数科教育とプログラミング教育における変数と不等号の扱い方や考え方の相違点を考察した。

その結果、算数科においては

- 不等号を具体的な数同士の大小比較の表記でしか扱っていない
- 内包的記法によって数の集合の性質を表現することを扱っている単元が存在しない
- プログラミングで求められる、式に役割を持たせるための立式を扱っている単元が存在しない

ため、プログラミング教育で用いられる算数科に関する概念が従来の算数教育では扱われていない可能性が示唆された。これらの問題を解決することができる算数科の単元の特定と指導方法の考案が課題である。

参考文献

- [1] 平林一栄他：算数・数学科の重要用語 300 の基礎知識（明治図書），p.243, 1981年