

フローチャートによるアルゴリズムの視覚化と検証システムの開発

山本恒

ICT 活用教育研究所
yamamoto@ei-labo.jp

フローチャートはアルゴリズムを視覚的に表現するのに効果的であるが、フローチャートが正しく動作するかどうかは、実際にプログラミングをしないと検証できない。プログラミングをしながらアルゴリズムの検証をする場合は、プログラミング言語の知識不足や記述ミスなどにより、アルゴリズムの学習なのかプログラミングの学習なのか、学習の目的が曖昧になってしまうことが多々見受けられる。そこで、画面上で部品を組み合わせて表現したフローチャートを、実際に実行させて検証するシステムを開発した。

1. はじめに

フローチャートはアルゴリズムを視覚的に表現するのに効果的である。しかし、フローチャートを試行錯誤しながら描くには面倒であったり、アルゴリズムを即時に検証したりできないので、アルゴリズムを考える手段として活用されることが少ない。

プログラミングをしながらアルゴリズムの検証をする場合は、プログラミング言語の知識不足や記述ミスなどにより、アルゴリズムの学習なのかプログラミングの学習なのか、学習の目的が曖昧になってしまうことが多々見受けられる。

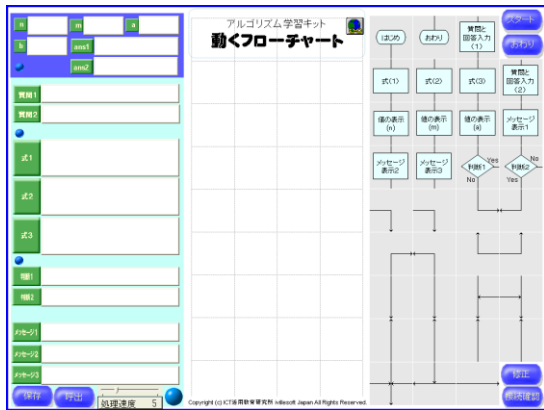


図1 動くフローチャート全体イメージ

そこで筆者は、マウス操作でフローチャートの部品を組み合わせて入力することができ、かつ各部品のプロパティを入力することで、即実行できる図1のようなシステム「動くフローチャート」をMicroWorldEX (Logo 言語) で開発した。

2. フローチャートの部品とそのプロパティ

部品とプロパティの関係を図2に示す。図の左側が各部品のプロパティを入力する部分である。図中の①～⑤の順に部品とプロパティの関係を説明する。

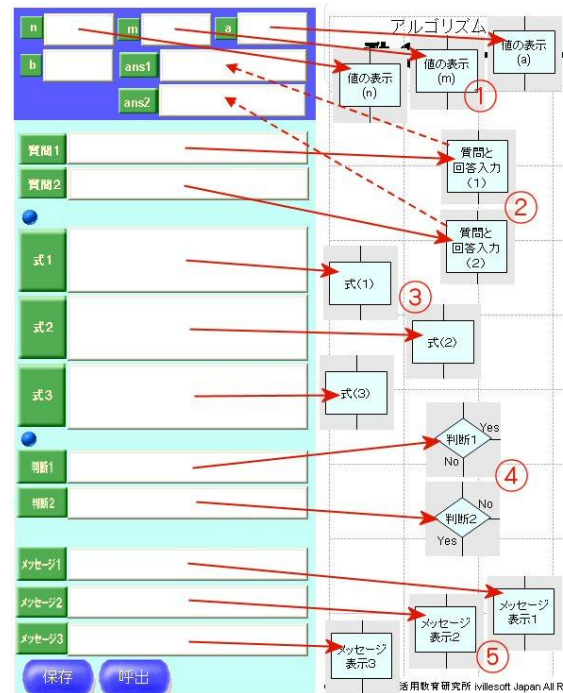


図2 フローチャートの部品とそのプロパティ

1. 左上のエリアに、可視化できる変数 $n, m, a, b, ans1, ans2$ を用意した。これを用いて計算処理ができる。処理中の変数の値はリアルタイムで表示される。この変数のなかで、 n, m, a の値については、部品「値の表示」で変数の中身をメッセージボックスで表示できる。

2. 部品「質問と回答入力」は、図3のようにプロパティの質問1,2に書かれた内容を表示し、回答入力

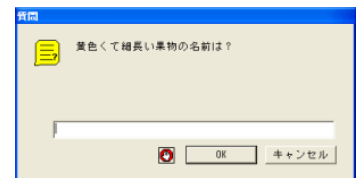


図3 質問と回答入力

求める。入力された値は、それぞれ変数 $ans1, ans2$ に自動的に代入される。

3. 部品「式」は、プロパティの式に入力された

初期値の設定や計算の式を実行する。

4. 部品「判断」は、プロパティの判断 1,2 に書かれた条件文の真偽を判断して、真なら Yes の方向に、偽なら No の方向に処理の流れを制御する。

5. 部品「メッセージ表示」は、プロパティのメッセージ 1,2,3 に書かれた内容を、それぞれメッセージボックスで表示する。

3. フローチャートの作成

スタートボタンを押してから、部品をクリックして反転させて、中央のマスに移動させてフローチャートを作っていく。マスの中央付近に移動すると、マスに吸い付くようになっている。部品間の接続線、合流線の部品も用意している。

フローチャートが完成したら、接続確認のボタンを押して、物理的に接続ができていないかどうかを調べる。物理的に接続ができておれば、必要なプロパティを入力する。

4. フローチャートの実行

「はじめ」の部品をクリックすると、部品が流れに従って反転しながら、順に実行していくので、処理の様子がよくわかる。図 4 は、数あてゲームのフローチャートで、数が合わなかったので、再度入力に戻っていく様子である。

図 5 は処理の速度を制御するスライダとボタンである。ボタンをクリックすると処理の流れを止めることもできる。

また、プロパティが誤記入で処理できない場合は、その旨のメッセージを表示する。考えたアルゴリズムと違った動きをした場合は、修正ボタンを押して部品を並べなおしたり、プロパティを修正し、再度実行できる。

5. 作成したフローチャート例

図 6 は、正しい回答が得られるまで質問をくりかえすアルゴリズムのフローチャート例である。

質問と回答入力(1)では、計算問題を質問し、値の入力を求める。入力された値は変数 ans1 に自動的に代入される。判断 1 では判断 1 に書かれた条件文の真偽を判断して、真なら Yes の方向に、偽

なら No の方向に、処理の流れを制御する。メッセージ表示 1 では、間違いのメッセージをだし、フローチャートの流れを質問まで戻す。メッセージ表示 2 では、正解のメッセージをだし、終わりの方向にフローチャートの流れを制御する。

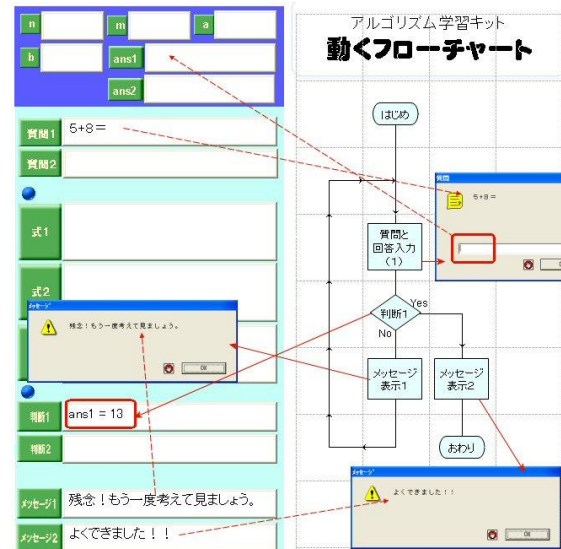


図 6 正しい回答になるまで質問をくりかえす

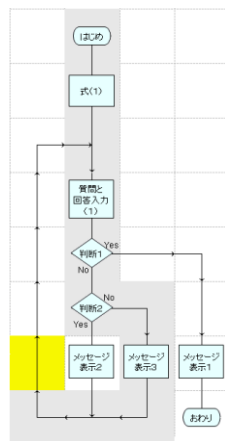


図 4 流れの様子

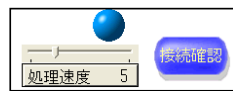


図 5 処理速度の制御

実際にプログラミングの作業まで進める場合は、検証できたフローチャートに従って、どのように記述すればいいのかに目的を絞ることができる。図 7 のように部品ごとに手順を作らせて、構造化された理解しやすいプログラムを Logo で書かせることができる。

このことが可能なのは、このシステムを MicroWorldEX で開発した理由でもある。

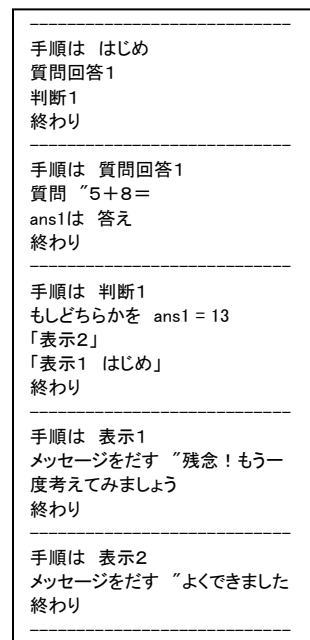


図 7 プログラミング例

6. おわりに

技術家庭科や情報 B の教科書では、情報処理の手順の例をフローチャートで表現している。フローチャートを読み解く力だけでなく、アルゴリズムを自ら考え出す力の育成に、このシステムが役立つと考えている。また、Logo はマイナーな言語になっているが、このシステムが Logo で開発されているように高度な機能を持つ言語として発展しており、教育用言語としても有効である。

今後、実践をとおして効果を検証していきたい。