

プログラミング学習における状態遷移概念の利用とその効果

鎌田 敏之* 田端 将太郎** 本多 満正*

*技術教育講座

**東浦町役場

Effect of Utilizing State Transition Diagram in Learning Computer Programs

Toshiyuki KAMADA*, Shotaro TABATA** and Mitsumasa HONDA*

*Department of Technology Education, Aichi University of Education, Kariya 448-8542, Japan

** Higashiura Town Office, Higashiura, Aichi 470-2192, Japan

1. はじめに

中等教育におけるプログラミングの学習において、フローチャートを用いて手順を考えることが従来から実践されてきた。しかし、プログラム作成にあたり、全体の構想を考える段階が不足していることから、条件や反復を含むフローチャート作成に困難を生じることが先行研究において指摘されている。本論文では、先行研究で扱われつつ、今後の課題とされてきた、状態遷移の概念を学習者に指導する方法について述べる。

平成20年告示の現行学習指導要領¹⁾の下で行われている、中学校技術・家庭科技術分野（以下、中学校技術科と呼ぶ）における「プログラミングによる計測と制御」及び高等学校共通科目情報科「情報の科学」における「モデル化とシミュレーション」では、アルゴリズムから処理の手順を考えるにあたり、フローチャートを用いることが示され、検定教科書においてもフローチャートにより処理の手順が例示されている。これを受け、中学校技術科の市販教材は、専用のアプリケーションソフトウェアを用い、フローチャートをPC画面上で描くことでプログラムを自動生成し、マイクロプロセッサを搭載した模型自動車など、計測制御を行う装置へプログラムを書き込み、実行するように構築されたものが主流である²⁾。しかし、プログラムによる課題解決にあたっては、課題を適切にモデル化し、解決手段をコンピュータの処理に適合した手順として表現しなければならない。

そこで本研究では、こうしたプロセスを「設計」と呼ぶこととした。フローチャートは条件分岐や反復のように具体的手順の表記に適した、いわばマイクロな表記法であるた

め、これらを含まない、より抽象的な表記法を導入することで、より課題の分析に適したマクロな捉えを学習者に促すことが必要であると考えた。

計測と制御を行うシステムを例にした場合、システムはセンサから得られる値に基づき、目的達成に適した制御の指示をコンピュータが与えるよう設計する必要がある。システムの動作をマクロに捉え、分割し、相互に動作を移りかえるモデルが、このようなシステムの実現によく適合することが知られている。本研究ではこの表現である「状態遷移図」を設計段階に活用することで、課題の解決が正確に行われるようになることを仮説とし、その検証を行うことを目的とした。また、解決が正確であることの判定とともに、それを学習者自身に実感させるため、状態遷移図から手順すなわちフローチャートへの変換を行う指導法を開発し、その教育効果について評価することとした。なお、本研究では学校での利用を見据え、教員養成の大学生を対象とした学習プラン作成と評価のための実験授業を行った。

2. 状態遷移の考え方と設計への活用

2.1 状態遷移図

状態遷移の考え方では、コンピュータによって実現されるシステムの処理全体を、一定のまとまりごとに分割し、そのまとまりを「状態」と定義する。何らかの入力や時間経過などをきっかけに、別の「状態」の実行に移る（遷移する）ことが連続と行われ、処理全体が行われると考える。例えばアクセルによって加速し、ブレーキによって減速する、単純な模型自動車を考えた場合、最初は「停止」して

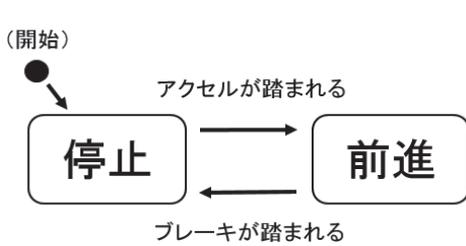


図 1 状態遷移図の例

おり、アクセルを踏むことをきっかけとして「停止」状態から「前進」状態へと遷移する。その後、ブレーキを踏むことをきっかけとして「前進」状態から「停止」状態へと遷移する。ここで「きっかけ」と述べた部分を「イベント」と呼ぶ。イベントが発生しない場合は、遷移が行われないため、同じ状態にとどまる。図 1 に、これを「状態遷移図」として示した。ここからわかるように、「停止」状態にある場合、アクセルを踏まない限り「停止」状態にとどまる。ブレーキを踏んだ場合は定義されていないため、イベントは発生しないとみなし、同様に「停止」状態にとどまる。

2.2 状態遷移図とプログラムの設計

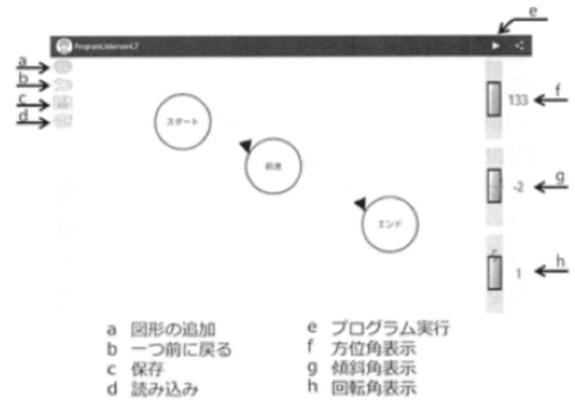
状態遷移図⁵⁾は、1987年にHarelによって整理、提案されるまで、様々な表記が行われてきたが、現代ではこれを拡張し、プログラムに必要な諸概念を図表現する言語的な標準規格「統一モデリング言語」⁶⁾に「ステート図」として取り入れられている。図 1 は、ステート図を理解のため簡略化した表現である。

したがって、本研究における「設計」の前半段階であるモデル化に状態遷移図を用い、これを手順の表現に変換することが後半段階にあたる。

3. 関連研究

3.1 状態遷移図を用いたプログラミング教材

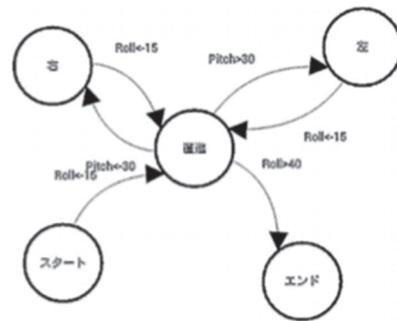
室伏ら⁷⁾は、タブレット端末を利用した計測・制御教材を作成するにあたり、状態遷移図をモデルとしたプログラムの作成及び実行環境を構築した。この教材ではまず、動作を図形の中に入れ矢印でつなぐビジュアル型入力を用いる（図 2(a)）。状態を円により表現した図形を配置し、次の状態に遷移する条件を矢印で描く。状態の名称と具体的な遷移条件は、円または矢印をタップすることで、テキスト入力画面を呼び出し（図 2(b)）、記述する。こうして作成した状態遷移図は、そのままプログラムとして実行させることができる。これを用い、中学生にタブレットの傾きセンサによる計測を用いロボットを制御するプログラム作成を課題として与えた結果、状態遷移の考え方を利用し、センサの値を制御に反映するプログラムを 10 名中 8 名の生徒が作成できている（図 2(c)）。この実験授業の観察から、「同じ課題に対して複数の解決方法が考えられるとき、その考え方の差異を明確に表現できる



(a) ビジュアル型入力画面



(b) テキスト入力画面



(c) 生徒の作成したプログラム

図 2 室伏らの作成した教材(文献⁷⁾より)

ことが明らかになった。さらに、生徒の多様な考えを認めることができるとともに、自分の考えを他者に伝える言語活動にも活用ができると考えられる」ことを発見している。

これは、生徒ごとの問題解決における思考過程の違いが状態遷移図モデルを共通言語とすることで、手順よりマクロな理解の水準で、直感的に考えを伝え合う学習効果について述べたものと考えられる。しかし、従来のフローチャートを用いた手続きを中心とした学習との比較がないことから、手続きに関する学習に対する優位性や、状態遷移モデルのみを用いることの限界については明らかにされていない。

3.2 状態遷移概念の利用有無を比較した教育実践

井戸坂ら⁸⁾は、計測・制御学習に状態遷移概念を利用した教材を作成し、中学 3 年生を対象としたプログラミングの授業を通してその学習効果を検証した。具体的には、画面上の図形操作と自動車型自律ロボットを用いた全 6 時間の授業計画に対し、既存の手続き型

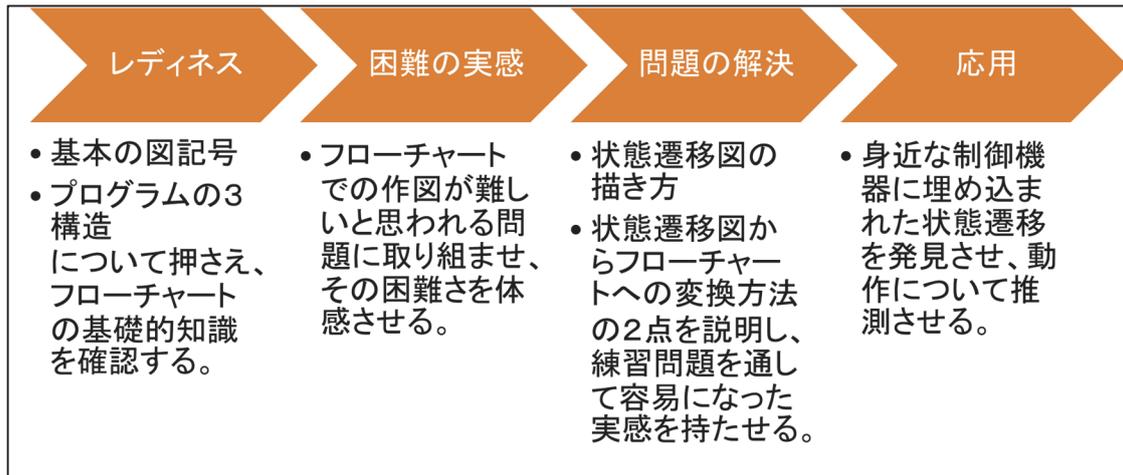


図3 作成した学習プラン

ビジュアルプログラミング言語「Squeak EToys」⁹⁾と、これを拡張し、状態遷移に基づく記述を可能としたプログラミング言語「Rtoys」¹⁰⁾を用い、単純な課題から徐々に複雑となる課題と、最後の自由課題を対象に、Squeak Etoys を用いた対照群と、Rtoys を用いた実験群間の課題達成率を比較した。その結果明らかになったこととして、以下の2点を挙げている。

- (1) 手続き型のプログラミングに比べ、状態遷移の考え方に基づく方法は、複雑な動きをするプログラムを記述できるようになる。
- (2) ただし、状態遷移の考え方は自然に気付くものではなく、指導する必要がある。

すなわち、状態遷移図概念を加えることで複雑なプログラム達成度が優位に高まることを示しながら、「この考え方は指導を通して初めて身につくものである」と考察している。一方、実験授業における指導は、「プログラム作成過程で不具合が生じた際指導した」の記述にとどまっており、具体的な指導法の探求については触れられていない。

他の文献を含め、状態遷移の考え方を伝える効果的な指導法について書かれた論文は管見の限り見当たらなかった。よって、状態遷移概念を導入した教材の教育効果として、学習者が、より複雑なプログラムの作成を達成できるものの、概念そのものを身につけさせるための指導法は未だ確立されていない。

4. 研究の目的と方法

本研究で提案する「状態遷移図」を、プログラムによって解決しようとする課題の分析段階に用いることで、複雑な課題に対しても正確な手順を組み立てられるという仮説の検証と、状態遷移の考え方を身につけさせる指導法について開発し、その教育効果を検証することを本研究の目的とする。研究は、以下の手順をとる。

1. 予備調査
2. 学習プラン作成
3. 事前調査
4. 実験授業
5. 事後調査
6. 追加調査

まず、単純ながらフローチャートのみでは混乱しやすいと考える例題を与え、どのような誤りが多いかを明らかにする予備調査を行い、その結果から、学習プランを作成する。学習プランに基づき実験授業を行い、実験授業の前後に実施した事前・事後調査を分析することで学習効果を検証する。また、追加調査により、現段階の学習プランが働く範囲の探求を行う。

4.1 予備調査と学習プラン

予備調査として、図1に示したアクセルとブレーキの例を与えた。フローチャートにより手順を図示させた。その後、車体の前方に距離センサが設置されていることとし、前の車との距離が一定になるよう、前進と停止を行う場合どうなるかを描かせた。

その結果、前者は5分以内に被験者（大学4年生）11名全員が正解したが、後者については、1名のみが正解であり、他の10名は図を完成させることができなかった。正解となった1名も、動作の前に距離の判定を行う条件を配置したことで、距離判定の分岐の結果、前の処理へ戻る線をどう引くべきかの試行錯誤を繰り返し、最終的に、動作を先に行い、その後、距離判定を行う形を見出す経過をたどったことがわかった。

この結果より、状態遷移図を用いて課題を分析し、それをフローチャートへ変換するパターンを与える指導法に思い当たった。これを含む、学習プランと実験授業の結果については、次節で述べる。

1. 図の●を「開始」と描き、●から出る線に対応し、開始の下へ線を引く
2. 「停止」状態に対応する処理「停止」を描く
3. 状態遷移の条件に対応し、判断「アクセルが踏まれる」を描き、処理「停止」と線で結ぶ
4. 判断「アクセルが踏まれる」がYesの場合、遷移先「前進」に対応し、処理「前進」と線で結ぶ
5. Noの場合、同じ処理を繰り返すため、停止を繰り返すよう、矢印を停止の上に引く
6. 前進についても同様に、3～5の手順で判断「ブレーキが踏まれる」の描画をする。ただしYesの場合、処理「停止」に戻る矢印を描く

図4 状態遷移図からフローチャートへの変換方法

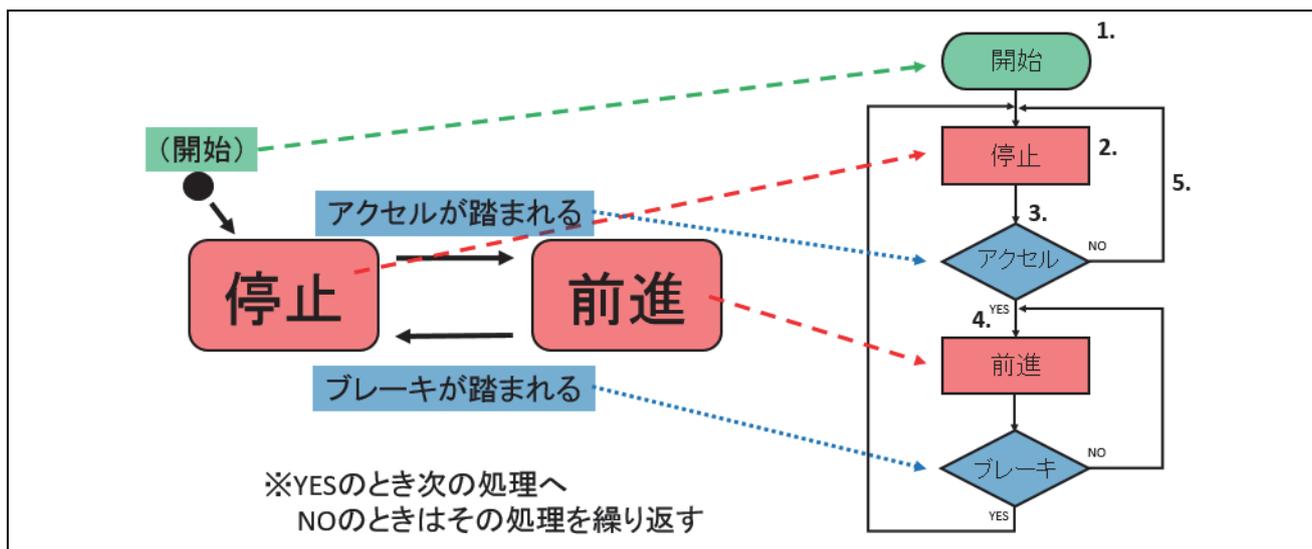


図5 フローチャート変換方法の簡略図

5. 学習プラン

5.1 構成

作成した学習プランを図3に示す。各ステップのねらいは、フローチャートによる手順まで自ら導く技能の獲得を目的とするため、最初にフローチャートの描き方をすべての学習者に身に付けさせること、次にフローチャートでは正しい手順を考えにくい課題を与え、他の解決法の必要性を強く自覚させること、その後、状態遷移図の描き方とフローチャートへの変換方法を指導することで、これまでできなかった問題が解決できること、さらに複数の異なる性質をもつ練習問題を解決することで自信を持たせること、最後に自ら状態遷移の考え方を応用する場面を設け、自立して課題に取り組む態度を促すことである。

5.2 教授する状態遷移図からの変換方法

学習プランのステップ3で教授する、状態遷移図からフローチャートへの変換方法（パターン）を、図4と図5に示す。この段階では、ひとつの状態のなかに複数の処理が含まれる課題は難しいと判断し、状態名と処理内容をあえて区別せず、そのまま対応するかのよう扱うことで直感的理解をはかった。また、状態遷移は条件の成立によってイベントが発生することから、フローチャートの「判断」に対応するとし、同じ

状態にとどまる場合の反復が、「上に戻る矢印を描く」ことに相当するとした。

6. 実験授業とその結果

6.1 調査内容と被検者

実験授業で与える問題は、予備調査の前半、アクセルとブレーキの例をそのまま採用した。学習プランのステップ2に相当する「困難の実感」には、後半の「前車との距離を一定に保つ」を採用し、その結果を事前調査とした。また、ステップ3を終え、「距離を保つ」状態遷移図作成の方法を学び、作図した結果とステップ4の応用問題を回答させた結果を事後調査とした。ただし、ステップ4については同一の基準で評価するため、同一の課題「エアコンの動作を状態遷移図として描く」を与えた。さらに、事後調査用紙に自由な感想を記述させる欄を設けた。

被験者として、愛知教育大学中等教育教員養成課程技術専攻の大学3年生13名を対象とし、2017年10月に1時間程度で実施した。被験者の中に状態遷移について事前に知っていた学生はおらず、約半数はフローチャートについても本実験授業で初めて学習する者であった。

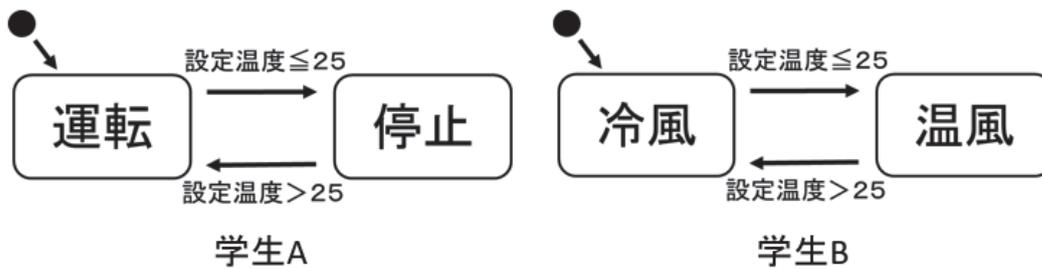


図6 エアコンの状態遷移図の例

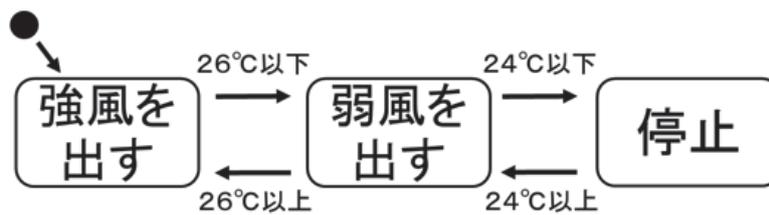


図7 3つの状態を持つ状態遷移図の例

6.2 事前調査の結果

事前調査の結果のうち、アクセルとブレーキの例について先に述べる。予備調査同様、13名中1名の正解にとどまった。判断条件が不適切、不要な反復の使用、条件と処理の結合の誤りがあり、正しい処理の流れに至る問題分析ができていないことが分かった。また、誤答の多くは、アクセルの判断のみで動作を決定し、ブレーキを記述しなかった。「終了」を記述しようとして反復構造に誤りが出てしまうことも多くみられた。

「距離を一定に保つ」例では、15分の時間内に正答にたどり着いたのは12名中2名であった。「判断」を先に設定して分岐の行き先が不明となり途中で記述が止まってしまった被検者が4名、書き終えたが分岐の行き先を誤って記述した被験者が3名いた。残り3名は正答者から解き方を聞いた可能性があるため無効な回答とした。正答者が少なかった理由について、問題文が「ある条件を満たしたとき、動作する」となっているため、その順序のまま「判断」を先に、「処理」を後に置いてしまったことが大きな要因と考えられる。

6.3 事後調査の結果

5分以内に12名全員が正答した。エアコンの課題については、12名中10名が、図6の回答となった。また、1名は図7のように、設定温度との差を考慮した3状態の図を描くことができた。

6.4 被検者の感想

12名中11名が肯定的な回答であった。典型的な回答として、「プログラミングをしていると頭の中が混乱してくることがある。その時に状態遷移図を教えれ

ば頭の中が整理できる」があった。否定的な回答として、中学校での授業を想定し、「中学生には難しすぎるような気がする」という回答があった。

6.5 結果のまとめ

ステップ1において困難の実感を持たせるという目的を達成したと考えられるため、この設問は適切であると判断した。事前と事後の調査結果から、問題の分析に状態遷移図を用いることで、正しい処理を行うフローチャートを描くことができる効果が認められた。エアコンの状態遷移図は、回答が1名を除き2状態となった。これは、それ以前に与えた課題が2状態で回答できるものであるためと考えられる。よって、3状態以上の課題について追加調査が必要である。

7. 追加調査

7.1 調査の目的と内容

実験授業では、例題として最も単純な、状態数が2個となる場合のみを扱った。本研究で開発した学習プランを用い、3つ以上の状態をもつ課題についても適切な回答を導き出すことができるかを確かめるため、以下の2つの課題に対し、状態遷移図を描かせ、フローチャートへ変換させた。

- (1) 駅に電車が停止している。駅にいる車掌の笛の合図で電車は加速をはじめ、次の駅まで500mで減速し、駅の停止線を越えると停止します。これを繰り返す電車の動作を作図してください。
- (2) 車の左右に、斜め前方との距離を測定するセンサーがついています。前進中に、車の左右に障害

物がはみだしています。障害物を「左回り」「右回り」で避ける動作を作図してください。

7.2 結果

(1)は、「停止」→「加速」→「減速」→「停止」を繰り返す、円環状の状態遷移図となる。結果は、11名が正しい状態遷移図を記述し、うち10人がフローチャートに変換できた。どちらも描けなかった1名は「加速」の前に設問にない「前進」状態を置き、「加速」へ移る際の遷移条件を独自に設定していた。また繰り返しのない図であり、円環状の繰り返しを表現できていなかった。

(2)は「前進」状態から、左右2つのセンサの値により、障害物を避けるよう「右回り」「左回り」のいずれかに遷移する。また、これらの動作の結果、障害物が避けられた（距離が十分遠くなった）ことをきっかけに、「前進」状態へ戻る遷移を行う。よって、「前進」状態を中心に、左右に遷移が振り分けられ、そこから中心の「前進」戻り形状の状態遷移図となる。結果は、7名が正しい状態遷移図を描いた。しかしフローチャートに正しく変換できたのは2名であり、うち両方を正しく記述したのは1名にとどまった。

誤ったフローチャートのほとんどは、前進処理のあと、左右のセンサの値判定へ別れ、同時にそれを行う条件記号及びそれに続く回避動作と、前進へ復帰する上に戻る線を左右対称に描いていた。左右を同時に判定するためには並行処理が必要だが、一般的なフローチャートでは並行処理を描く方法が定められていないことから誤答と判断した。正答と想定したのは、前進へ戻ることを含む3分岐であり、左右センサを順番に条件判定した後、いずれかに合致すれば回避動作に移り、どちらも合致しない場合に前進へ直接戻るものであった。他の誤答としては、判断がNOの場合の行き先が欠けていたものがあった。

3分岐のフローチャートを描く技能は教えられずに自ら気づくことが難しい例として一般に知られているが、本研究における調査により、状態遷移図を描くことができた場合でも、3分岐に気づくことは難しいことがわかった。その理由のひとつとして、ステップ3でパターンとして与えた変換方法が、処理を先に行い、その後分岐で遷移するか戻るかを判断する形であったことが影響したと考えられる。

並行処理を意図したと思われるフローチャートの誤りに関しては、統一モデリング言語で定義される13種類の図のうち、フローチャートに相当する「アクティビティ図」に並行処理に関する記号が定義されている。平成29年公示の新学習指導要領解説¹¹⁾では、フローチャートにかえて統一モデリング言語の利用を記述しており、この発想が一般的なのであれば、アクティビティ図とステート図の組み合わせを指導する方法であれば誤答ではなくなることが予想される。ただし、

並行処理を前提としない3以上の分岐に対応する解とはならないこと、こうした多重分岐が必要となる例は特殊なものではないことから、今後、変換方法の指導法について、さらに検討する必要がある。

8. まとめにかえて

中等教育段階のプログラミング教育においては、従来方法論が不足していたとの考えから、本研究に取り組んできた。①問題の分析過程を「プログラムの設計段階」として、状態遷移図を用い、その必要性を実感させるため、フローチャートだけでは難しい課題を与え、状態遷移図を介してフローチャートへ変換させる指導法を用いた。大学生を対象とした実験授業を通し、正答率が上昇した。②被験者は、プログラミングの対象となる問題の分析に状態遷移図を描くことの意義を実感を伴って理解できたと調査に回答した。このことは開発した指導法の教育効果を示すものと考えられる。③3つ以上の状態において、多重分岐の処理に還元する必要のある問題に対して限界があることが分かった。今後、これを克服する指導法の開発を進めていく予定である。

参考文献

- 1) 文部科学省：中学校学習指導要領（2008）
- 2) 文部科学省：高等学校学習指導要領（2018）
- 3) ヤマザキ教育システム：制御学習プロロボ USB プラス
<http://www.yamazaki-kk.com/technique/detail.php?id=58&t=1&c=5>（2018年9月24日閲覧）
- 4) ヴィストーン株式会社：ビュートレーサー
http://www.vstone.co.jp/products/beauto_racer/（2018年9月24日閲覧）
- 5) David Harel, Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231274 (June 1987)
- 6) OMG: OMG Unified Modeling Language (OMG UML) version 2.5.1 (Dec. 2017)
<https://www.omg.org/spec/UML/2.5.1>（2018年9月24日閲覧）
- 7) 室伏 春樹・高木 薫：タブレット端末を利用したプログラムによる計測・制御教材の開発、日本産業技術教育学会誌，第57巻，第3号，pp.179-186 (2015)
- 8) 井戸坂幸男・青木浩幸・李元揆・久野靖・兼宗進：状態遷移概念を利用した制御プログラミングの学習効果、日本産業技術教育学会誌，第53巻，第3号，pp.179-187 (2011)
- 9) Squeak Etoys <http://www.squeakland.org/>（2018年9月24日閲覧）

- 1 0) Hiroyuki Aoki, DongHee Park, WonGyu Lee:
Robot Programming with Squeak “Rtoys”^{*} -
Connection between Drawing Worlds and the
Real World, International Conference on
Technology Education in the Asia Pacific
Region Conference Proceedings, pp.390-398
(2009)
- 1 1) 文部科学省：中学校学習指導要領解説 技
術・家庭編 (2017)

(2018年9月25日受理)