

基本アルゴリズム理解の分析と教育への応用

浅野考平[†] 森戸隆文[†]

基本アルゴリズムの理解の過程を分析する方法を示す。それをバブルソートのアルゴリズムに応用して、理解の構造が記述される。また、分析結果のアルゴリズムの教育への応用についても議論する。

Analysis of the process of understanding basic algorithms with application to education

KOUHEI ASANO[†] TAKAFUMI MORITO[†]

A method to analyze the process of understanding basic algorithms is presented. It is used to bubblesort and the structure of the understanding is described. An application of the analysis to algorithm education is discussed.

1. はじめに

コンピュータがアルゴリズムを実行するときは、プログラムを記憶し、プログラムに記述された手順に従って、条件判断による分岐などを行いながら、逐次的に実行する。しかし、人は、アルゴリズムの手順を暗記して、それを思い出して実行するのではなく、何らかの構造を作って記憶し、また、その構造を利用して思い出し（想起し）、実行すると考えられる。本研究で、その仕組みを分析する方法を提案し、初歩的なアルゴリズムに適用して、その仕組みを記述する。

ここでは「アルゴリズムを理解する」とは、アルゴリズムの記述を参照せずに、任意の入力に対してトレースできるようにすること、であると考え。なぜならば、一般的に、アルゴリズムを理解していることを確認するにはトレースさせるからである。丁寧に言えば、「アルゴリズムの記述を参照することなしに、与えられた適当な大きさの入力に対して、その実行過程を示めすことができる」ようになることである。従って、理解の仕組みとは、人がアルゴリズムをトレースする際に用いる仕組みのことである。

ただし、提案する方法で分析するのは、すべてのアルゴリズムではなく、スタック、キュー、二分探索木などの基本的データ構造とその操作、二分探索や各種ソートのアルゴリズムなど、情報科学の専門教育の初期段階で教える初歩的アルゴリズムである。これらを基本アルゴリズムと呼ぶことにする。

アルゴリズム理解の仕組みは、以下の2つの条件を満たすことが必要である。第一に、当然のこととして、アルゴリズムの正しい理解、すなわち、一定の大きさの入力であれば任意の入力に対して、正当な出力を得る仕組みでな

ければならない。第二に、教育に応用し効果的な教育に結びつけることを目的としているのであるから、心的な負担の少ない理解、すなわち、長期的に記憶に残り、かつ、想起しやすい理解の仕組みでなければならない。

人の持つ知識は異なり、直接関係のないような知識も結びつけて記憶し、それを想起するのであるから、これらの仕組みは、人によって異なり、1つのアルゴリズムに対して、唯一の理解の仕組みが存在するわけではない。そこで、本研究が目的としているのは上の2つの条件を満たす“最適な”仕組み、すなわち、正当な出力を得る、できるだけ心的な負担の少ない理解の仕組みを見いだす方法である。

2. 理解の仕組みの分析方法

理解に仕組みがあり、それを明らかにできるというのは認知心理学の立場であろう。そこで、筆者たちは認知心理学の研究者ではないが、1970年代から1980年代にかけて、研究された物語スキーマという理論を参考にして、アルゴリズム理解の仕組みを説明することを試みる。[1][3]

スキーマの概念は、1930年代に、人が物語を理解するときに、自分の過去の経験に合うように再構成して憶えるという現象を説明するために、Bartlett によって導入された。人が物語を理解するときは、既存の自分の認識の枠組みを用いて理解し、さらに物語の内容を想起するときも、自分の認識の枠組みに基づいて想起する。このような認識の枠組みをスキーマと呼ぶことにした。

1970年代になって知識獲得の仕組みを説明するための概念として再び注目された。例えば、Mandlerらは、物語文法という方法で物語のスキーマを表現した。[2][4] 彼らは、書き換え（分解）規則を定め、この規則を用いて、物語を分解していくと、最終的に物語の文章の表現にまでいたるということを示し、物語には書き換え規則によってあらわされるスキーマがあり、人は彼らの表現したようなスキーマを持っていて、そのスキーマにしたがって物語を

[†]関西学院大学
Kwansei Gakuin University

理解していることを、実験的に示したとしている。

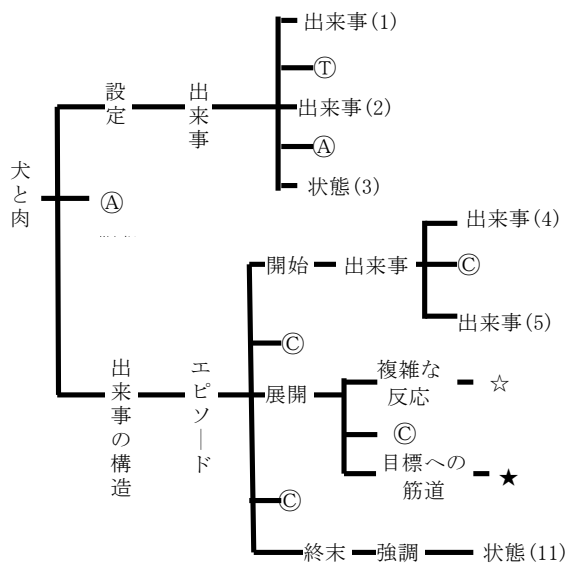
書き換え規則の一部をあげる。ここで、Ⓐ, ㉔, ㉕は AND, CAUSE, THEN の略号であり命題の結合関係を表している。

- (1) 物語 → 設定 Ⓐ 出来事の構造
- (2) 設定 → 状態 Ⓐ 出来事 または、出来事
出来事 → 出来事 (Ⓐ, ㉔, ㉕) 出来事 Ⓐ 状態
- (3) 出来事の構造 → エピソード ㉕ エピソード
- (4) エピソード → 開始 ㉔ 展開 ㉔ 終末
- (5) 開始 → 複数の出来事, エピソード
- (6) 展開 → 簡単な反応 ㉔ 行為, 複雑な反応
複雑な反応 ㉔ 目標への筋道

例：イソップ童話の犬と肉を物語文法によってまとめた図を示す。

- (1) ある犬が、肉をもらい、
- (2) その肉をくわえて家に帰ろうとしていた。
- (3) 家までの間の川に架かった橋をわたらなければ家には帰れなかった。
- (4) 橋を渡りながら、ふと下を見ると、
- (5) 水面に自分自身の姿が写っていた。
- (6) 犬は見知らぬ犬が肉をくわえていると思い、
- (7) その肉が欲しくなった。
- (8) そこで、脅すために吠えた。
- (9) すると、くわえていた肉が川に落ちて、
- (10) 流されてしまった。
- (11) そして、その肉を二度と見ることはできなかった。

Mandler らは、この物語の構造を物語文法によって次のように図式化している。[2]



上の☆, ★は、それぞれ下の☆, ★と繋がっている。

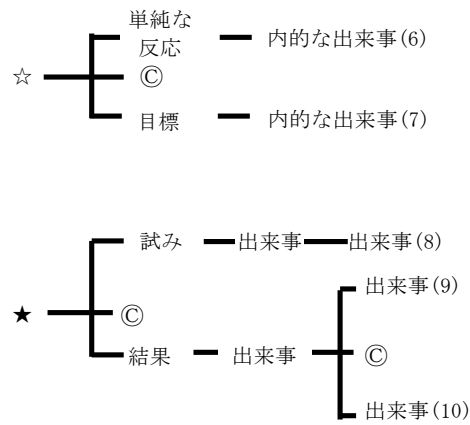


図 1 Mandler-Johnson[2]の例を筆者が翻訳

本研究で提案する方法においては、最初に、上記の物語文法による物語理解のスキーマを参考にして、アルゴリズムのトレースを最下位レベルに持つ階層構造をつくる。その理由は、物語は、時間の経過にそった出来事の展開と完結の記述であり、アルゴリズムのトレースは、時間の経過にそった終了までの処理を順次記述したもので、この点で共通点があるからである。

物語文法によってつくられる階層構造は下位レベルのノードをまとめて上位レベルのノードをつくるので、上位レベルは全体として、下位レベルの要約になっている。一方、アルゴリズムの理解として、望ましい、心的負担の小さい仕組みは、トップダウンによる想起の列によって実現できると考えられる。

しかしながら、他方では物語とアルゴリズムとは異なっている。まず、特定の入力に対するトレースは、時間経過にそった1つの流れの展開であるが、アルゴリズムは条件による分岐があり複数の展開がある。現在のところ、分析を試みた基本アルゴリズムについては、実行過程では分岐し、その後の実行過程が異なっても、高位のレベルのスキーマとしては、共通の構造を持っていた。また、高位のレベルでも構造が異なる場合は、複数の構造をつくれれば良いと考えている。

物語文法は、(単純な)物語に共通の構造を抽出することを目的としていた。物語文法の書き換え規則による図式では、文脈とは無関係な構造をつくるので、高位のノードに、対応する具体的な意味は考慮されなかった。しかし、アルゴリズム一般に共通の構造があるとは考えられない。そこで、筆者たちは、書き換え規則によって、アルゴリズムそれぞれに特有の構造をつくると考える。高位のノードについても独自の“ひとまとまり”の意味を持っているものとし、その意味も重要視する。

上記の議論を踏まえて、アルゴリズムの理解の仕組みは、次のような手順で求めることを提案する。

- (1) 適当な大きさの入力に対してトレースする。また、

処理の単位に分割し、最下位レベルのノードの列をつくる。

(2)各ノードがひとまとまりの意味を持つように、順次、下位レベルのノードをまとめて上位レベルのノードをつくり、階層構造をつくる。

(3) この階層構造がアルゴリズムの知識の構造を示していると見なして、上位レベルの要約から下位レベルに向かって、アルゴリズムをトレースするために想起する過程を作成する。

具体的には「下位レベルの手順を忘れたときには、何を手がかりにして思い出すか」という問いに答えるのである。このことによって、上位レベルと下位レベルを関連づけられる適当な手がかりを探し、もし答えが見出せるならば関連づけるために必要な知識を明らかにする。

3. バブルソートへの適用

例 バブルソート

```
for (i = 0; i < n-1; i++)
    for (j = n-1; j > i; j--)
        if (a[j-1] > a[j]) {swap(j-1, j, a)}
```

(1) 入力に対するトレース

次のような配列を入力する場合にトレースする。ただし、この段階で、いくつかの処理をまとめて記述している。

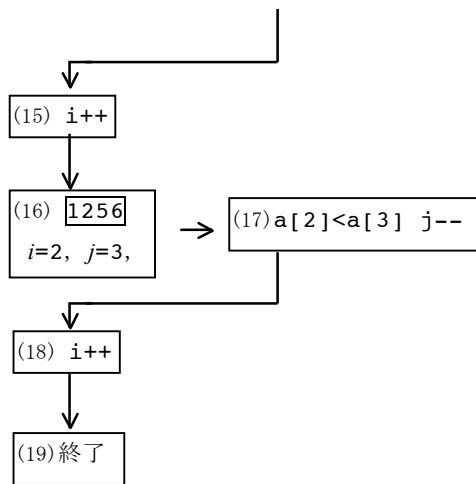
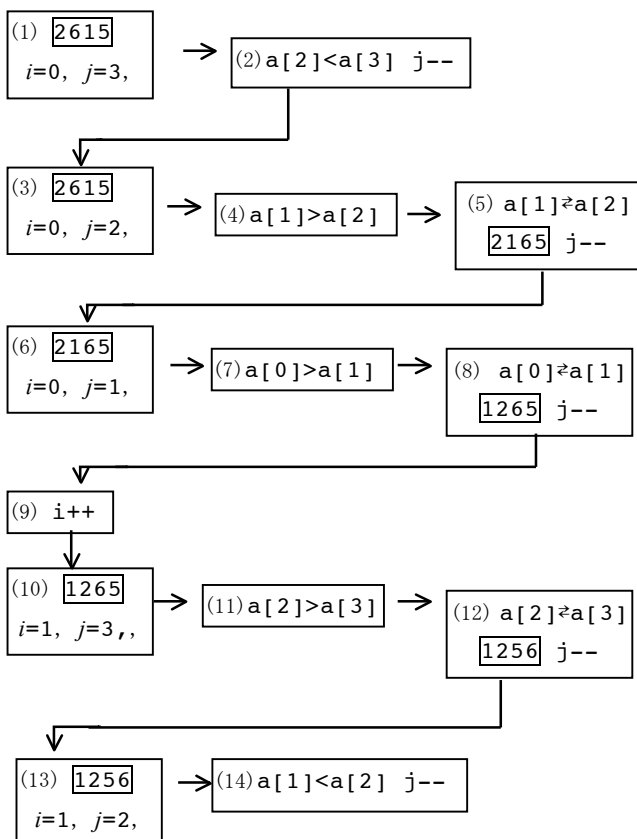
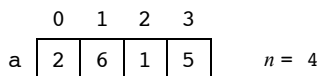


図 2 バブルソートのトレース

(2) これらを前記の方法で図式化する。まず、最下位のレベルのノードは、トレースにおける各処理とする。そして、いくつかのノードがひとまとまりの意味を持つようにまとめて上位レベルのノードをつくる。

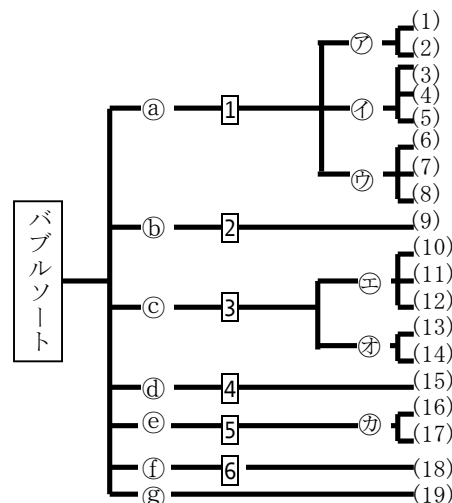


図 3 バブルソート理解の階層

各ノードの意味は以下の通りである。

- ㉞~㊹: 「 $a[i] > a[j]$ かつ $i < j$ 」のとき逆位と言うことにすると、隣接した逆位の修正である。
- ㊱, ㊳, ㊵: 最小値を配列の左端に移動する。
- ㊲, ㊴, ㊶: 配列の大きさを1だけ減らす。
- a, c, e: 集合から最小値を取り出す。
- b, d, f: 集合の要素数を1だけ減らす。

階層構造の作り方から、各レベルごとに横断的にノードを並べると、バブルソートの要約になっており、上位は下位の要約になっていることがわかる。

バブルソートの各レベルごと要約は①, ②, ③のよう考えられる。

① 最上位レベルの要約: バブルソートは、数の集合から最

小値を取り出し、残りの集合から最小値を取り出すという操作を繰り返す。

② バブルソートは、数が並んだ配列の左端に最小の数を移動し、対象とする配列の長さを1だけ減らすという操作を繰り返す。

③ バブルソートは、隣接した逆位の修正を繰り返して、配列の左端に最小の数を移動する。これを繰り返すことによってソートする。ただし、このレベルでは、逆位の修正の順序が具体的に指定される必要がある。

注意：この場合トレースの段階では入力によって、処理内容は異なるが、上位のレベルには反映されない。

(3) 図式に表された最上位のレベルの要約は、最も素朴なソートの戦略であり、日常経験から容易に類推できる。それより下の階層について、もし下位レベルを忘れたとき、上位から何を手がかりにして下位を想起すれば良いか、という問いの答えを考えることによって、上位レベルから下位レベルへの関連を見出す。

①から②は、配列を用いること、昇順にするということを利用すれば容易に想起できる。③では、逆位の修正を行う順序が指定されている。従って、②から③を導くためには、配列の左端に最小の数を移動するためにはどのような順序で、逆位の修正を行うのかを想起する必要がある。②から③の関連づけは容易ではないと考えられる。

4. 分析結果の評価と今後の検討課題

前の節で、バブルソートという具体的な基本アルゴリズムに提案している方法を適用して、最適であると考えられる理解の仕組みを作成した。この仕組みによって、任意の入力に対してアルゴリズムを正しくトレースできることは明らかである。また、心的な負担が少ないことは、筆者たちを含めた複数によって主観的に確認しているが、現段階では実験によって確かめることはできていない。

本研究の最終的な目的は、すべての基本アルゴリズムに対して、適切な理解の仕組みを作成する方法をつくることである。それを実現するためには次のような問題点がある。

(1) 先にも指摘しているように、アルゴリズムのトレースからはじめている。従って、異なる入力では異なる階層構造になる可能性がある。

(2) トレースを最下位レベルとして、ひとまりの意味を持つようにまとめる作業は機械的にはできない。それ故、うまくまとめられない、あるいは、まとめても適当な意味を見いだせない場合がある可能性がある。

(3) 要約の階層構造が作成されても、上位から下位への関連づけは、個別に、対象となるアルゴリズムに沿って考察する以外の方法はなく、一般的方法はない。

(4) クイックソートなど再帰を含むアルゴリズムに対する分析は困難である。

(5) 基本アルゴリズムには、データ構造が含まれている。

データ構造についても、概念の理解の仕組みを作成する必要がある。

筆者たちは、データ構造とデータ構造の操作を含む、いくつかの基本アルゴリズムについて、この方法を適用して仕組みを分析しつつある。また現時点では、仕組みを作成することは可能であると考えている。

ここで述べたような方法で図式化し、分析することの必要性について述べたい。結果を見ればアルゴリズムを直接分析することによって同じような仕組みを思いつくことも不可能ではないように見える。しかし、ここで述べた方法で階層構造をつくることによって、飛躍することなく段階的に関連づけることができる。

5. 教育への応用の可能性

ここまで、理解の仕組みを分析する方法を提案し、それに基づいてバブルソートの理解の仕組みを作成した。バブルソートを理解させるためには、その仕組みを学習者が形成すればよいのだから、分析に基づいて、次のように教えることが考えられる。

(1) バブルソートは、数の集合から最小の要素を取り出す。残りの集合に対しても同様にすることによってソートするアルゴリズムであることを教える。

(2) 配列で(1)を実現するためには、最小の要素を左端に移動し、配列から除外すれば良いことを教える。

(3) (2)を実現するために逆位の修正を利用すること、修正の順序を教える。ただし、修正の順序を想起することは容易ではない。それ故、例えば、具体的に、右端に近いところに最小の要素がたくわえられている配列を与えて修正の順序を考えさせるなどの工夫して、想起する手がかりも同時に教えておく。

上記の教育への応用を一般化することもできる。例えば、アンプラグドの教育内容は、上位レベルの要約であると考えられる。段階を明確にすることによって、アンプラグドの教育に、どのような知識を付け加えればアルゴリズムの理解に到達できるか、具体的に明らかにできる。また、逆にアルゴリズムのアンプラグドの教材を構成することもできる。

参考文献

- 1) 岸学：スキーマ 日本教育工学会（編）教育工学事典，実教出版 pp.336 (2000)
- 2) Mandler, J.M. and Johnson, N.S.: Remembrance of things parsed: Story structure and recall, *Cognitive Psychology* 9, pp.111-151 (1977).
- 3) 佐伯胖：スキーマ 東洋（編）教育の心理学的基礎，朝倉書店 (1982)
- 4) Thorndyke P. W.: Cognitive structures in comprehension and memory of narrative discourse, *Cognitive Psychology* 9, pp.77-110 (1977).