A11-5

アルゴリズム的思考のビジュアル化によるラーニング構造の抽出

川口 順功　高橋 等　永田奈央美　大石義
静岡産業大学　情報学部
{ kawaguti, h-taka, nagata, oishi } @ ssu.ac.jp

概要：アルゴリズム教育では、アルゴリズムを使えるようにすることが優先されがちになるが、アルゴリズムを学ぶ意義はその本質的な理解と応用にあると思われる。アルゴリズムを構成するベースとなるものの捉え方や処理方法を類型化してラーニング構造を定義すると、アルゴリズムの本質的な理解は、そのラーニング構造の理解に還元できる。本稿では、アルゴリズムのビジュアル化によって抽出されるラーニング構造の具体例と、そのラーニング構造の応用について報告する。

1　はじめに

アルゴリズムを教育で扱うとき、それを使って問題を解けるようにすることが優先されがちになる。しかし、アルゴリズムを学ぶ意義は、使えるようになるとともに、アルゴリズムの本質的な理解とその応用にあると思われる。アルゴリズムを構成するベースとなるものの捉え方や処理方法を類型化し、ラーニング構造として定義すると、アルゴリズムの本質的な理解は、それを構成するいくつかのラーニング構造の理解に還元できる。

本稿では、アルゴリズムのビジュアル化によって抽出されるラーニング構造の具体例と、そのラーニング構造の応用について報告する。

2　アルゴリズム的思考およびラーニング構造とその抽出方法について

本稿で対象とするアルゴリズム的思考は、問題解決の手順としてのアルゴリズムを中心に、大きく次のようなものからなる。

① 情報教育の中で扱うプログラミングを前提としたアルゴリズム。
② 数学、理科などの分野で使われる問題を解く手順や手法としてのアルゴリズム。
③ 問題解決の手順ではないが、問題の分析や統合などで用いるアルゴリズム的な手法や思考法。

なお、以下では、アルゴリズム的思考のことを、単にアルゴリズムと呼ぶ場合がある。

ラーニング構造は、アルゴリズムを構成するベースとなるものの捉え方や処理方法を類型化したもので、アルゴリズムから学ぶべき内容をまとめたものである。ラーニング構造を作る目的は、アルゴリズムの本質的な内容を理解し、それを別のアルゴリズムの理解や創出に利用できるようになることである。

アルゴリズムから抽出されるラーニング構造は、通常は「ものの捉え方」と「処理方法」を組み合わせたものになるが、アルゴリズムによっては「ものの捉え方」または「処理方法」のどちらかが相手を含む場合がある。ソートアルゴリズムの場合、配列をどのように捉えて、どのように処理するかによってラーニング構造が決まる。クイックソートのラーニング構造は、例えば「集合型配列＋再帰展開型木構造」となる(詳細は後述)。「集合型配列」が配列の捉え方、「再帰展開型木構造」が処理方法であり、この２つのラーニング構造が、クイックソートのアルゴリズムから学ぶべき内容ということになる。

また、本研究では、次のような手順でラーニング構造を抽出する。

① アルゴリズムが対象とするものの捉え方や処理方法を分析する。
② 分析結果に基づき、アルゴリズムを図やアニメーションなどを使ってビジュアル化する。
③ アルゴリズムの内容とそれをビジュアル化したものからラーニング構造を決め、さらに、抽出されたラーニング構造を再定義する。

3　アルゴリズム的思考からのラーニング構造の抽出例

ここでは、アルゴリズム的思考のビジュアル化について述べ、その後ソートアルゴリズムや濃度算などのアルゴリズムから抽出される具体的なラーニング構造について述べる。

3.1 アルゴリズム的思考のビジュアル化

筆者等は、アルゴリズムのビジュアル化を積極的に行い、情報教育に利用している[1][2][3]。アルゴリズムの中心となる考え方をコア・イメージとして図にしたり、ソートアルゴリズムで展開される配列のデータ移動を flash でアニメーション化したりして、授業やラーニング構造の抽出に利用している。アルゴリズムのコア・イメージについては後述するので、ここではソートアルゴリズムのアニメーション化について述べる。

図 3-1 は、ヒープソートをアニメーション化したときのアニメーション図の一部である。配列とヒープ木(木構造の一つ)のマッピングの様子を視覚的に捉えることができ、ヒープソートのアルゴリズムの仕組みの理解に使える。また、バブル（単純選択）ソートやクイックソートのデータの動きに注目すると、配列の扱い方からそれぞれが「線形型配列」と「集合型配列」(詳細は後述）になることを視覚的に捉えることができる。

ラーニング構造には、ビジュアル化しなくてもアルゴリズムそのものから決定できるものもある。しかし、本研究では、基本的にビジュアル化したものもアルゴリズムのラーニング構造の決定に使用することにしている。主な理由は、次のとおりである。

① ビジュアル化を考えるとき、アルゴリズムをさまざまな観点から捉え直すことができる。
② ビジュアル化したものは、アルゴリズム理解のツールの一つになる。
③ ビジュアル化したものが、抽出されるラーニング構造の根拠となり得る。
④ ビジュアル化によって予想しなかったラーニング構造が抽出できる場合がある。

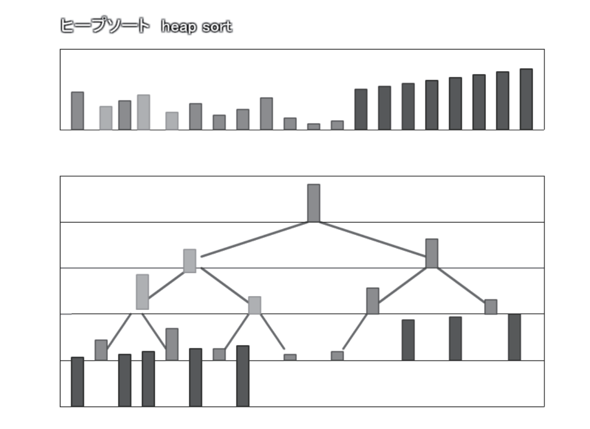

図 3-1　ヒープソートのアニメーション図

3. 2 ソートアルゴリズムと展開型木構造

ここで扱うソート対象のデータは、すべて配列である。ソートアルゴリズムでは、配列をどのように捉えて、どのように処理するかによってラーニング構造が決まるので、まず、「ものの捉え方」としての配列の捉え方と、「処理方法」と関連する木構造について述べる。

配列は、どのように捉えて処理するかによって、図 3-2 のように「線形型配列」「集合型配列」「循環型配列」などの構造に分類できる[1]。配列の探索を例にすると、線形探索に代表されるように配列の要素を個々に連続的に扱う場合を「線形型配列」、2 分探索に代表されるように配列の要素を個別に扱うのではなく集合的に扱う場合を「集合型配列」と呼び、これをラーニング構造とすることができる。

木構造は、図 3-3(a)に示すようなデータ構造で、組織図や分析図など日常的にも利用されている。また、図 3-3(b)(c)のように枝（edge）の部分を矢印にして表記すると、(b)が「展開型木構造」、(c)が「統合型木構造」のラーニング構造として使える。木構造に関しては、アルゴリズム教育のデータ構造で扱う以外に体系的に学ぶことはないが、アルゴリズムの具体的な手順が展開、分析、統合の形になる場合が多いことから、アルゴリズムをビジュアル化すると木構造になることが多くなる。

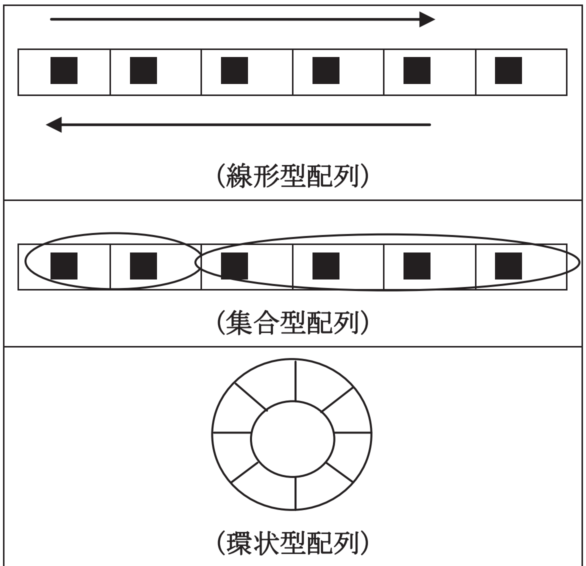

図 3-2　配列のラーニング構造

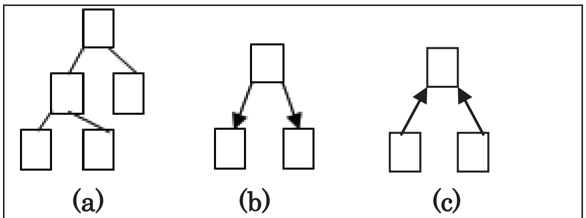

図 3-3　木構造のイメージ

筆者の一人は、図を使ってアルゴリズムのコア・イメージを作成し、アルゴリズム教育の授業で利用している[2][3]。図 3-4 は、その授業で使用しているクイックソートのコア・イメージである。クイックソートは、ソートアルゴリズムの一つで、「ピボット（pivot：基軸となる値）により、データをピボットより小さいグループと大きいグループの 2 つに分け、さらに分けた 2 つのグループに同じ操作を再帰的に適用する」という考え方に基づく。

この図より、クイックソートは、配列を「集合型配列」で捉え、処理が再帰的に木構造で展開されていることがわかるので、ラーニング構造を「集合型配列＋再帰展開型木構造」と決めることができる。「再帰展開型木構造」は、展開の仕方が再帰的アルゴリズムに基づくことに由来してつけた構造名である。単に「展開型木構造」とすることもできるが、この構造はさまざまな展開パターンを含むので、より具体的な展開をイメージできるように、ここでは「再帰展開型木構造」としている。

このアルゴリズムから抽出される「再帰展開型木構造」を、例えば、以下のように定義できる。

① 展開の順序は２分木の深さ優先順であり、プログラムの再帰呼び出しで展開される順序となる。

② ２分木の葉は、再帰呼び出しの終了条件で配列の要素が１個の場合である。

③ 分割のされ方によってはアンバランスな木になり、処理効率が落ちる可能性がある。

もちろん、このラーニング構造は、他のアルゴリズムから抽出される場合もあるので、その内容を加味して再定義される。したがって、「再帰展開型木構造」の定義が段階的に充実し、これがアルゴリズムから学ぶべき内容としての辞書的な役割を果たすことになる。

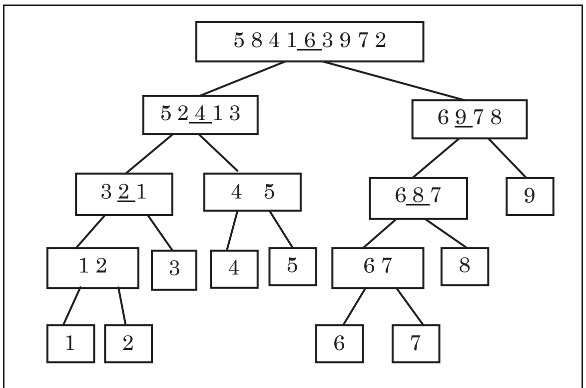

図 3-4 クイックソートのコア・イメージ

図 3-5 は、同じくアルゴリズム教育の授業の中で使用しているマージソートのコア・イメージである。マージソートもソートアルゴリズムの一つで、「配列を前半部と後半部の二つに分け、それぞれをソートした後、それらをマージ（併合）する」という考え方に基づく。

この図は、アルゴリズムにしたがって展開と統合が並行して行われる配列の状態を表し、上半分が展開、下半分が統合である。この図より、マージソートも、配列を「集合型配列」で捉え、処理が再帰的に木構造で展開されていることがわかる。したがって、ラーニング構造を「集合型配列＋再帰展開型木構造＋再帰統合型木構造」と決めることができる。そして、マージソートのアルゴリズムの本質的な学ぶべき内容は、この３つのラーニング構造に還元される。

「展開型木構造」の例として、順列を求める樹形図がある。図 3-6 は、「ABC の 3 人を並べる」順列を求める場合の樹形図である。この図より、順列を文字の並びで捉え、その経過を木構造で展開していることがわかるので、順列を求めるアルゴリズムのラーニング構造を「樹形図展開型木構造」と決めることができる。この樹形図のポイントは、すべての場合を重複せずに網羅する、いわゆる MECE(mutually exclusive collectively exhaustive)であるので、ラーニング構造「樹形図展開型木構造」の定義の一つに、「MECE」を加えることができる。

以上 3 つのケースの「展開型木構造」に含まれるラーニング構造について述べた。「展開型木構造」に含まれるラーニング構造はさまざまなアルゴリズム的思考の中に現れるので、これを体系的にまとめて類型化できれば、利用価値の高いものになると思われる。

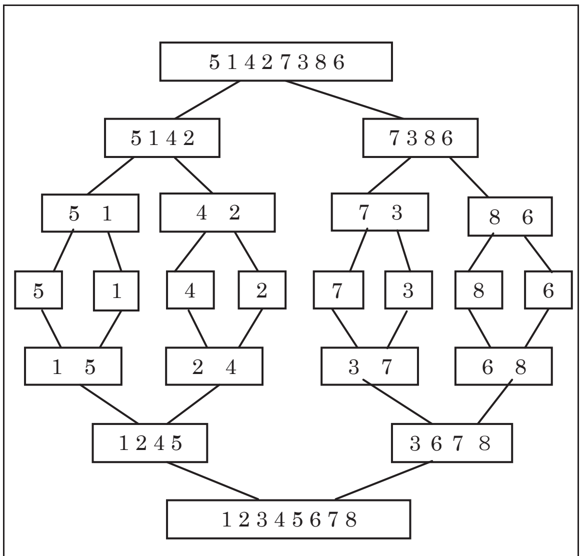

図 3-5　マージソートのコア・イメージ

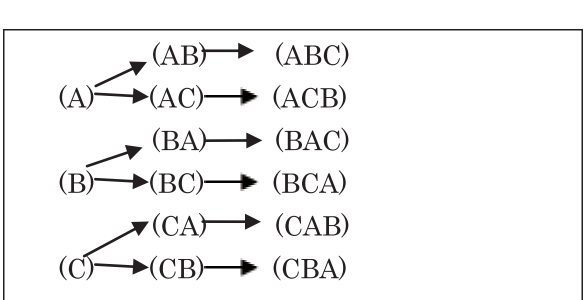

図 3-6　順列の樹形図

3.4 濃度算と計算型表構造

濃度算は、「5%の食塩水 400g と 9%の食塩水 600g を混ぜると何%の食塩水になるか？」というような問題である。図 3-7 は、アルゴリズム教育の授業の中で使っている濃度算のコア・イメージである。2 つの食塩水を混ぜたイメージを左に描き、それぞれの状態を「食塩水(g)×濃度（%）＝食塩（g)」の式で表現し、その合計を下段に求めるというアルゴリズムのイメージである。図 3-7 に基づく具体的なアルゴリズムは、次のようになる。

①「食塩水(g) × 濃度（%）＝食塩（g)」に基づき、図 3-8 のような表を作成する。そのとき、求めるもの（未知数）を x とおく。
② 食塩水と食塩の合計を求める
③ 最終的な一次方程式により未知数の値を求める。

図 3-8 は、この濃度算のコア・イメージに基づくアルゴリズムを使って作成した表である。この表と図 3-7 のコア・イメージより、それぞれの食塩水の状態を表の 1 行で表し、全体の表が処理方法を表していると考えられることから、濃度算のラーニング構造を「計算型表構造」と決めることができる。「計算型表構造」は、Excel などの表計算ソフトで扱う 2 次元の表のイメージである。

また、図 3-9 は売上の表計算の例である。商品名の列があるが、実質的な構造は 4 行 3 列の濃度算のものと同じである。この 2 つの表の例を参考に、ラーニング構造「計算型表構造」を、例えば、次のように定義することが考えられる。

① 列の項目には、合計が意味を持つものと持たないものがある。
② 列の項目には、他の項目から自動的に計算されるものがある。
③ 列の合計を求めることがアルゴリズムの一部に含まれる。
④ 2つ以上の個々の状態を表の一行に対応させることができ、それらの全体の状態を求めたい場合、「計算型表構造」を使える可能性がある。

3.5 鶴亀算と計算型図形

鶴亀算は、「鶴と亀が合計 20 羽（匹）いる。足の数は全部で 72 本である。亀は何匹いるか？」というような問題である。一般的には、方程式を使って解くが、アルゴリズム教育の授業の中ではアルゴリズムの観点から、方程式を使わないで解く方法を中心に扱っている。

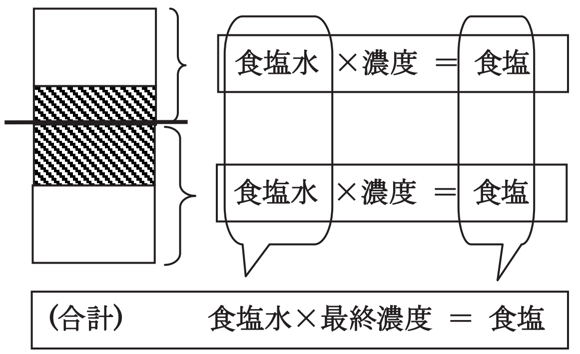

図 3-7 濃度算のコア・イメージ

食塩水	濃度	食塩
400	(×) 5 / 100	20
600	(×) 9 / 100	54
1000	(×) x	74

図 3-8 濃度算の表計算

商品名	単価	個数	金額
A	400	(×) 5	2000
B	600	(×) 9	5400
合計			7400

図 3-9 売上の表計算

図 3-10 は、授業の中で使っている鶴亀算のコア・イメージである。この図を使って、まず、基本的な考え方として、図の(B)の部分の足の数から亀の数を求めることを説明し、次に、下記のような具体的なアルゴリズムで解くことを説明する。

①（足の数の）大小差
大きい方と小さい方の差(大小差)を求める。
② 仮の合計（図の(A)部分の足の数）
すべて小さい方と仮定し、仮の合計を求める。
③ 仮の差（図の(B)部分の足の数）
「合計－仮の合計＝仮の差」
④ 大きい方（亀）
「仮の差÷大小差＝大きい方」
⑤ 小さい方（鶴）
「全体－大きい方＝小さい方」

上述した問題の解は、次のようになる。
①（大小差）4－2＝2
②（仮の合計） 2×20＝40

③（仮の差）72－40＝32
④（大きい方：亀）32÷2＝16
⑤（小さい方：鶴）20－16＝4

ここで、鶴亀算のコア・イメージを、足を１辺の長さが１の正方形で表した図に変換して考えると、図 3-11 のようになる。図に示しているように鶴と亀の足の数を横の辺の長さ、頭の数を縦の辺の長さ、全体の頭の数を面積で捉えると、鶴亀算の問題が面積と辺の長さを使った計算問題になることがわかる。前述した問題をこの図で解くと、
・（A）の部分の面積＝ 2×20 ＝40
・（B）の部分の面積＝ 72－40 ＝32
・（B）の部分の縦の長さ＝ 32÷2 ＝16（亀）
・（A）の上部分の縦の長さ＝ 20－16 ＝4（鶴）
となる。つまり、鶴亀算のアルゴリズムは、図形の面積や辺の長さを求める計算問題と構造的に同じになるということである。この図から鶴亀算のラーニング構造を、例えば「計算型図形」として、次のように定義することができる。
① 面積は三角形や四角形に分けて求める。
② 辺の長さや面積が、具体的にどのようなものと対応するかを考える。
③ 図形の中で、与えられた条件（前提条件）と求めるもの（解）を明確にし、最短で解を求める方法がアルゴリズムとなる可能性が高い。

3.6 集合演算と共有型集合

２つの集合 A と B の関係は、図 3-12 のように３つのパターンになる。(a)は「A が B を包含する」、(b)は「A と B が共通部分を持つ」、(c)は「A と B が共通部分を持たない」を表す。以下では、これらに対応するラーニング構造を、それぞれ「包含型集合」「共有型集合」「分離型集合」とする。

集合演算の問題に、「A の合格者は 20 名、B の合格者は 30 名で、AB 両方の合格者が 10 名であるとき、A または B に合格したのは何名か？」というものがある。この問題の A と B は、図 3-11(b)の集合関係を表す。この集合関係は処理方法も含むと考えて、この問題のラーニング構造を「共有型集合」と決めることができる。したがって、この「共有型集合」の定義には、次の基本的な演算式も含まれる。

$$n(A\cup B)=(n(A)-n(A\cap B)) + n(A\cap B)+(n(B)-n(A\cap B)) \quad (式1)$$

（ただし、n(A)は集合 A の要素数を表す）

(式１)は次の(式２)のように公式化されるが、アルゴリズムとして考える場合は、後述する理由により(式１)の方がよい。もちろん、等号で連結して考えるのであればよい。

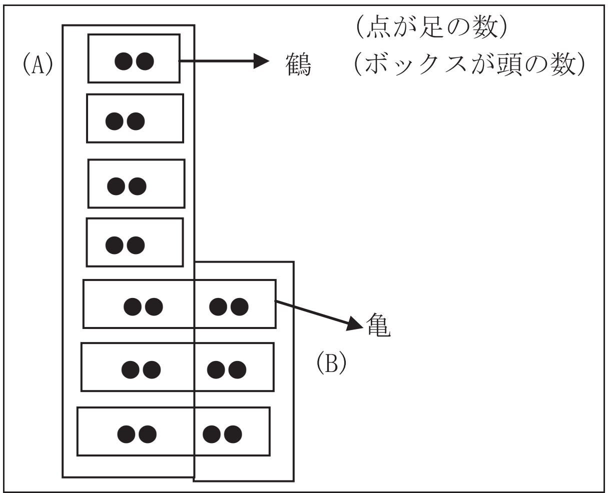

図 3-10 鶴亀算のコア・イメージ

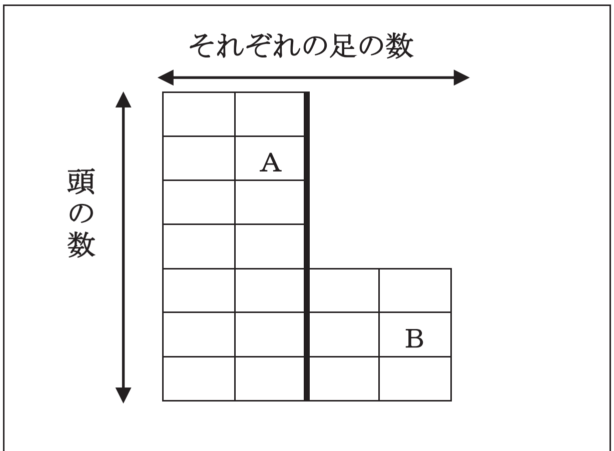

図 3-11 鶴亀算のコア・イメージ２

$$n(A\cup B)=n(A)+n(B)-n(A\cap B) \quad (式2)$$

ここで、ラーニング構造「共有型集合」について述べる。２つの集合 A と B を比較する（例えば特徴を取り出す）場合、一般的に「A と B に共通する部分」「A のみの部分」「B のみの部分」と分けて考える。このことを集合の記号で表すと、「A∪B」を、「A∩B」「A∩￢B」「￢A∩B」（￢X は X の補集合）の３つの集合で捉えることに対応する。このことも考慮して、ラーニング構造「共有型集合」を定義すると、例えば、次のようになる。
① 「A∩￢B」「A∩B」「￢A∩B」の３つに分けて考える。
② 分けた３つの集合のそれぞれの属性（集合の要素数など）を捉えてまとめる。
③ 要素数では、(式１）が成り立つ。

この定義から考えると、(式２)よりも(式１)の方がよいことがわかる。

また、「共有型集合」は、小学２年生で扱われる「A 君が前から 10 番目、後から 15 番目であるとき、この列の人数は？」のような問題にも応用できる。これは序数の扱いをテーマとした問題で、

A 君が重複していることを理解させた上で「10＋15－1＝24」という計算ができるようになることが目標となっている。しかし、一列に並ぶ状態を「A君より前の部分」「A君より後の部分」の 2 つの集合の関係で捉えると、「共有型集合」の定義が使える。したがって、この計算は、「(10－1) ＋1＋(15－1)＝24」の観点から指導する方がよいことになる。もちろん、小学 2 年生であることを考慮して式を 3 つに分ける必要があるが、少なくとも一列を「A 君より前」「A 君」「A 君より後」という 3 つの集合で捉えるような指導が望まれる。もう一つ、本論とは直接関係ないことであるが、「10＋15－1＝24」よりも「(10－1) ＋1＋(15－1)＝24」の式の方が良いのには、もう一つの理由がある。前者の式は、「10 番目＋15 番目－1 人＝24 人」と、本来加算が成立しない内容に誤解される恐れがあるからである。

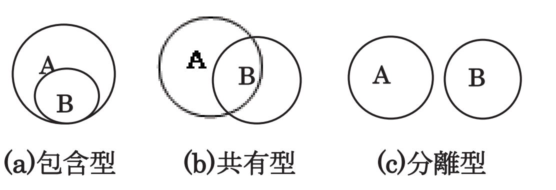

図 3-12　2 つの集合の集合関係

4 ラーニング構造の応用

ラーニング構造は、アルゴリズムから本質的に学ぶべき内容として定義するものである。しかし、個別のアルゴリズムを理解するためのベースになると同時に、他のアルゴリズムの理解にも応用できるものでなければならない。したがって、ラーニング構造を応用するには、次の 2 つのことが前提となる。

- アルゴリズムを構成しているラーニング構造は、一つ一つがアルゴリズムを理解するためのユニットになっている。
- ユニットとしてのラーニング構造の定義は段階的に更新されるものであり、定義に含まれる内容は、ある程度汎用性を持つものでなければならない。

このことを前提として、次のようなラーニング構造の応用が考えられる。

① アルゴリズムの本質的な理解のユニットとして使う。
② ラーニング構造を組み合わせて新しいアルゴリズムの創出に使う。
③ ラーニング構造を意識して、情報教育や数学などでのアルゴリズム教育を行う。
④ ラーニング構造を応用して、アルゴリズムに関する教材や e-ラーニングのコンテンツを効果的に作成する。
⑤ ラーニング構造を考慮して扱うアルゴリズム内容を絞り込み、限られた時間内で行う教育の効果を高める。
⑥ 学習者のアルゴリズム理解の分析に使う。例えば、理解されていない複数の個別のアルゴリズムに共通するラーニング構造を調べることによって、理解のネックになっているものを特定できる。

5 まとめ

本稿では、アルゴリズムを構成するベースとなるものの捉え方や処理方法に注目し、アルゴリズムの内容とそれをビジュアル化したものからラーニング構造を抽出する試みを紹介した。対象としたアルゴリズムと抽出したラーニング構造をまとめると、次のようになる。

アルゴリズム名	ラーニング構造
クイックソート	集合型配列 再帰展開型木構造
マージソート	集合型配列 再帰展開型木構造 再帰統合型木構造
順列	樹形図展開型木構造
濃度算	計算型表構造
鶴亀算	計算型図形
集合演算	共有型集合

もちろん、これらはすべて現時点での名称と内容である。より多くのアルゴリズムから抽出されるラーニング構造により、これらは類型化され、名称も含め内容が段階的に再定義されていく。このようなことを考慮して、今後の課題として、次のようなことが考えられる。

① ラーニング構造の有効性を評価・検証する。
② ラーニング構造の作成を前提したアルゴリズムのビジュアル化の方法を研究する。
③ アルゴリズムとそのラーニング構造の定義をセットにしたアルゴリズム学習データベースを作成する。ラーニング構造の定義は、タイムリーに更新可能なものにする。
④ ラーニング構造の効果的な応用に関する研究を行う。

参考文献

[1] 川口、「論理的データ構造とアルゴリズムのコア・イメージを利用したアルゴリズム教育」、第 26 回ファジィシステムシンポジウム講演論文集、2010
[2] 川口、渡邉、高橋、「言葉の理解を意識した情報教育の試み」、pp. 282−285 、平成 22 年度情報処理教育研究集会論文集、 2010
[3] 川口、「アルゴリズムのコア・イメージを使ったアルゴリズム教育」、第 27 回ファジィシステムシンポジウム講演論文集、 2011