

アンプラグド プログラミング及びプログラム 実行環境の提案

六沢 一昭^{1,a)} 藤田 茂^{1,b)}

概要：コンピュータサイエンスアンプラグドとは、コンピュータを使わずに情報科学を教えるための教育法である。本稿は、これに基づいたプログラミング及びプログラム実行環境を提案するものである。この環境は、トランプ大の矩形（エリア）が8個程度書かれたシートとカード（例えばトランプ）からなる。エリアには1枚以上のカードを重ねて置くことができ、「カードの移動」や「エリアが空であるかの検査」といった基礎的な操作のみを行なうことができる。エリア上のカード群はスタックを形成し、「カードの移動」は、移動元エリアに対する pop 処理と、移動先に対する push 処理から構成されることができる。プログラムはこれらの操作を使ってフローチャートにより記述し、実行は手作業で行なう。本環境は、千葉工業大学情報工学科 AO 入学試験において 2009 年度から 7 年間使用した。

キーワード：CS アンプラグド、プログラミング教育、アルゴリズム

1. はじめに

コンピュータサイエンス (CS) アンプラグドとは、コンピュータを使わずに、アルゴリズムなどの情報科学の原理をわかりやすく教えることを目的とした教育法である [1]。これまで多くの教材が開発されてきたが、その多くは、ひとつの対象に対してのものであった。つまり、対象ひとつに対してひとつの教材を開発してきたのであった。例えば、整列ソートを教えるための教材、テキスト圧縮を教えるための教材、... を個別に開発したのである。

本稿で提案する「プログラミング及びプログラム実行環境」は、記述できるものは非常に限られるが、アルゴリズム一般を対象とするものである^{*1}。必要な教具は、高々 B4 サイズの大きさの用紙と、

トランプのようなカードだけである。学習者はプログラム（アルゴリズム）をフローチャートにより記述し、実行は、フローチャートに従って自分の手でカードを操作することにより行なう。このため、プログラムの動きを無理なく理解し、プログラムの間違いに自ら自然に気がつくことが期待できる。

本環境は、千葉工業大学情報工学科 AO 入学試験において、アルゴリズム作成能力を問うために開発したものである。2009 年度から 7 年間の AO 入試における実績がある。

2. 実行環境の構成

実行環境は、シートと、シートの上で操作するカードからなる。

シート トランプ大の矩形（エリア）を 8 個程度設けたシートである。各エリアには名前を付け

¹ 千葉工業大学 情報科学部 情報工学科

a) rokusawa.kazuaki@it-chiba.ac.jp

b) fujita@cs.it-chiba.ac.jp

^{*1} 文献 [3] は、本稿が提案する環境と同様に、対象をひとつに限らない学習環境を提案するものである。

SHEET

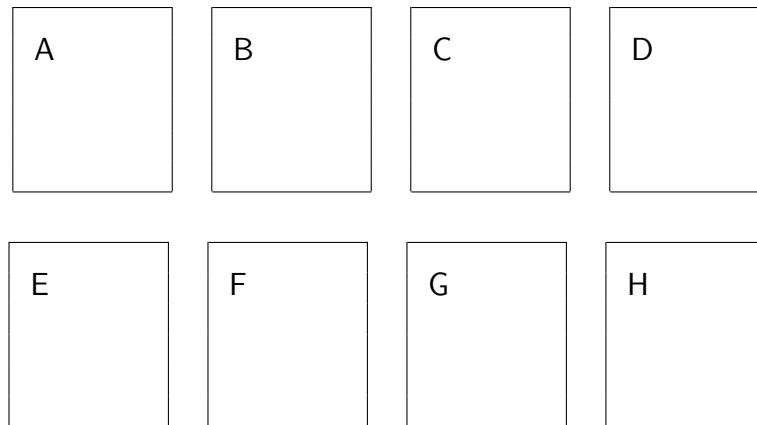


図 1 シートの例

る。シートの例を図 1 に示す。

カード トランプのように数値や記号などの書かれているカードである。

エリアの大きさや数は本質ではない。大きさは、使用するカードが収まればよく、数は 8 個でなくともよい。

3. 実行環境における操作

行なうことのできる操作を以下に示す。いずれも基礎的なものである。

- (a) カードの移動 (1) あるエリアのカードの一番上のカードを、他のエリアの一番上に移動させる。
- (b) カードの移動 (2) あるエリアのすべてのカードを、他のエリアの一番上に移動させる。^{*2}
- (c) 空であるかの検査 あるエリアが空であるか(カードがあるかどうか) 調べる。
- (d) 2 枚のカードの数値や記号の比較 2 つのエリアの一番上のカードの数値や記号を比較する。
- (e) カードの数値や記号の比較 あるエリアのカードの数値や記号と、指定した数値や記号を比較する。

図 2 は、これらの操作をまとめたものである。

^{*2} この操作は、操作 (a) と (c) から構成することができるので本質的に必要な操作ではない。

4. 課題例

入試などで試みた課題のいくつかを以下に示す。

4.1 カードの数値あるいは記号を考慮するもの

課題 (1) A にダイヤとクラブのカードがある。2 枚ある数値のカードをすべて見つけて H に置く。(A のカードが ♠4, ◇2, ♣8, ◇3, ♣2, ◇6, ◇8 ならば, ◇2, ♣2, ◇8, ♣8 を H に置くことになる。) 解答例を図 3 に示す。

課題 (2) A にあるカードから最も大きい数値のカードを見つけて H へ置く。解答例を図 4 に示す。

課題 (3) A には、ある数値 x のカードが 1 枚と、 x 以外のいくつかの数値のカードがそれぞれ 2 枚あるいは 3 枚ずつある。A から数値 x のカードを見つけて H に置く。

(A のカードが ♠5, ◇2, ♣3, ◇5, ♣5, ♥3 ならば, ◇2 を H に置くことになる。)

4.2 エリアのカードの枚数を考慮するもの

課題 (4) A に m 枚の、B に n 枚のカードがある。 m, n とも 1 以上であり、 $m > n$ である。

$m - n$ 枚のカードを H へ置く。解答例を図 5 に示す。

課題 (5) A に n 枚のカードがある。 n は 1 以上

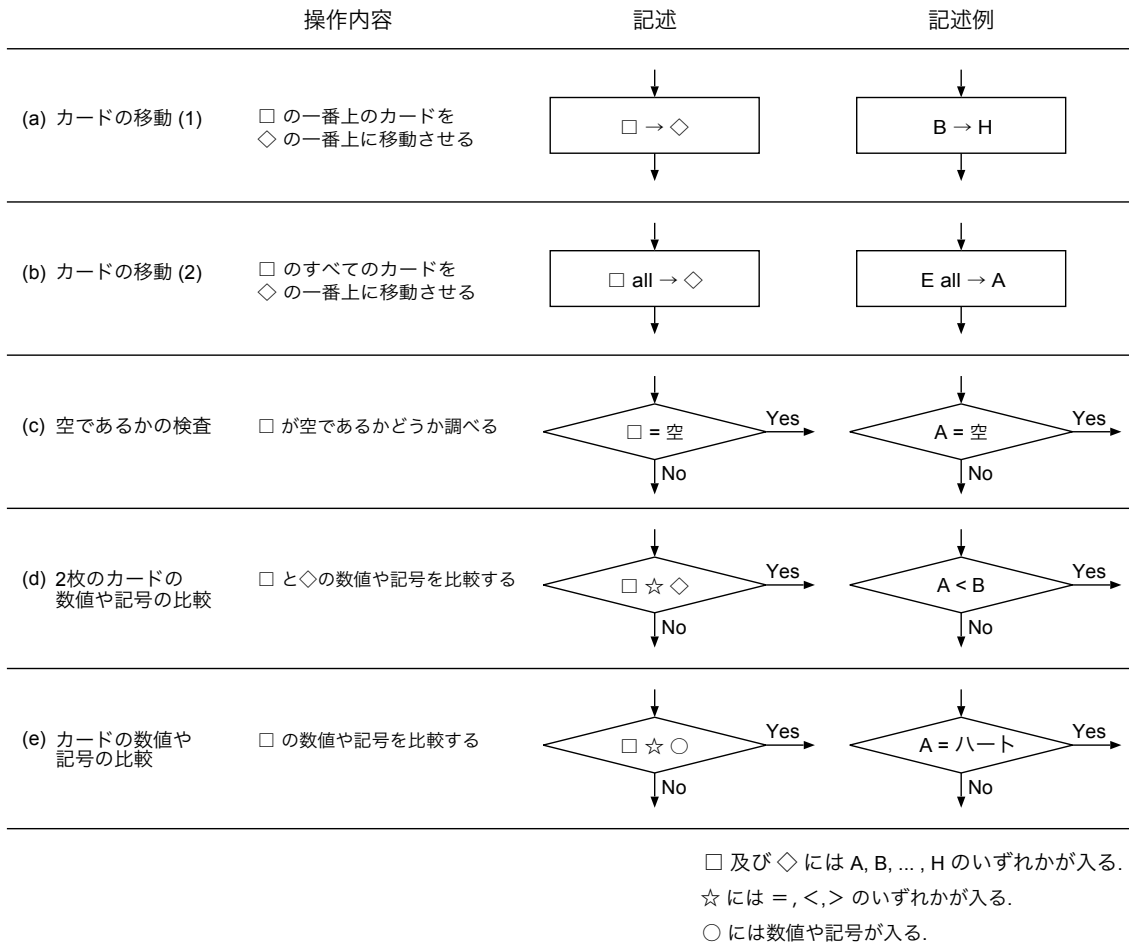


図 2 実行環境における操作

の整数であり、3 の倍数である。カードを操作し、 $n/3$ 枚のカードを A から H へ移す。

課題 (6) A に n 枚のカードがある。 n は 1 以上の整数である。 n が奇数ならば A のすべてのカードを H へ、偶数ならば D へ移す。 解答例を図 6 に示す。

5. 環境に対する考察

- エリアは、プログラムにおける変数に、エリアの名前は、変数の名前に対応すると考えることができる。
- エリア (変数) の値として以下の二つが考えられる。
 - エリアの一番上のカードの数値あるいは記

号を値とする。

– エリアのカードの枚数を値とする。

- エリアの一番上のカードの数値あるいは記号を値とする場合、エリアに積んだカードはスタックを構成すると考えることができる。従って、(一番上の) カードを取る操作は、スタックに対する **pop** 操作であり、エリア (の一番上) にカードを置く操作は、スタックに対する **push** 操作である。そして、カードの移動は、移動元エリアに対する **pop** 操作と、移動先エリアに対する **push** 操作から構成される。
- エリアのカードの枚数を値とする場合、エリアに積んだカードは枚数のみ意味を持つ。カードを取る操作は値をひとつ減らす操作であり、

カードを置く操作は値をひとつ増やす操作である。また、操作 (c) 空であるかの検査は、(変数の) 値が 0 であるかの検査に相当する。

課題 (4) は、+1, -1 及び 0 であるかの検査のみで実現した減算である。同様に、課題 (5) 及び (6) は除算及び奇数/偶数の判定である。

6. アンプラグド実行環境の使用実績

6.1 入学試験

千葉工業大学 情報工学科 AO 入学試験において、2009 年度の試験から 7 年間使用した。

合否の決定は、アルゴリズムの作成能力だけではなく多くの観点から総合的に行なったのであるが、限られた時間 (およそ 100 分) に、受験生が環境の操作などを理解し与えられた課題に取り組むことに困難を感じることはなかったと考えられる。

6.2 大学院の授業

昨年度と今年度 (2016 年度と 2017 年度) 情報工学専攻大学院授業「アルゴリズム特論」において、2 時間 (90 分 × 2 回)、本環境を使ったプログラム (アルゴリズム) の記述と実行を行なった。

- 受講者 15 人ほど (修士 1 年生)。
- 学生に感想を求めたところ概ね好評であった。
- カードの枚数をエリア (変数) の値とする課題と、カードの数値あるいは記号を値とする課題の両方を出題した。
- 学生のおよそ半数は、全問は正答に至らなかった。カードの枚数を値とする課題は比較的容易に解けたようであるが、カードの数値あるいは記号を値とする課題は難しいと感じた学生が多かったようである。
- ほとんどすべての学生が、自分の手による実行で自分のプログラム (アルゴリズム) の間違いに出会い修正を行なった。

7. 今後の課題

エリアのマトリクス状の配置を生かした処理はまだ全く考えられていない。配列の添字のようなものを導入することなく、上下左右といった相対位置のみを用いた記述を考えたい。

前述したマトリクス状の配置にも関係することであるが、配列あるいは配列に準ずるデータ構造を実現したい。基本操作が複雑になることなく、子供からお年寄りまで理解できるような単純で簡単な仕組みのみを用いての実現を目指したい。

二つのエリア A と B それぞれのデータ列 (カード群) をつなげてひとつのデータ列と考えることはできる。A の一番下のカードがデータ列の先頭で、B の一番下はデータ列の最後尾、それぞれの一番上のカードはデータ列の途中の二つとなる。このデータ列は、一ヶ所であるが途中を参照することができ、参照位置の移動も、エリア間のカードの移動で実現できる。これを生かした処理を考えたい。

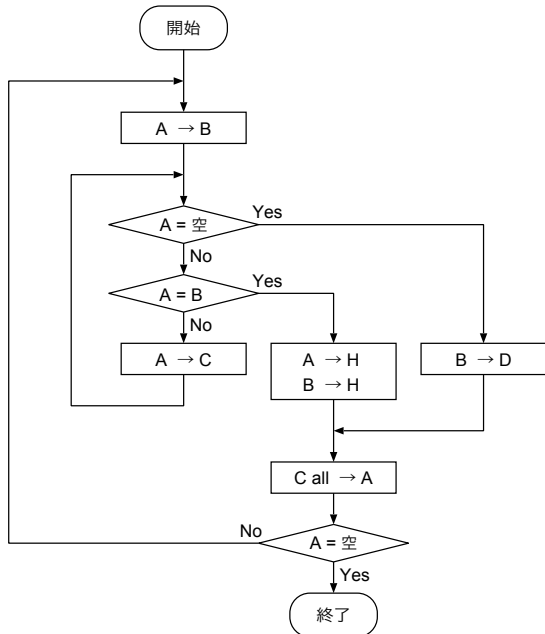
8. まとめ

CS アンプラグド プログラミング及びプログラム実行環境を提案した。本環境の特徴は、対象を特定しないことと操作が極めて単純であることである。

フローチャートにより記述したプログラムは、本環境上で手作業により容易に実行することができ、このため、本環境の利用者は、プログラムの動きの詳細を自然に理解することができ、またプログラムの間違いを自ら自然に発見することが期待できる。

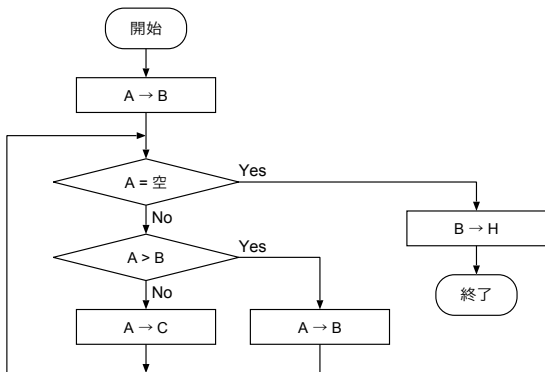
参考文献

- [1] Computer Science Unplugged Homepage,
<http://csunplugged.org>,
<http://csunplugged.jp>.
- [2] 六沢 一昭, “アンプラグド プログラミング及びプログラム実行環境の提案,” 第 10 回全国高等学校情報教育研究会全国大会, ポスター No.20, 2017.
- [3] 米田 貴, 御家 雄一, 伊藤 一成, “人型ピクトグラムを用いた CS アンプラグドの統一学習環境の提案,” 第 10 回全国高等学校情報教育研究会全国大会, pp.32-33, 2017.



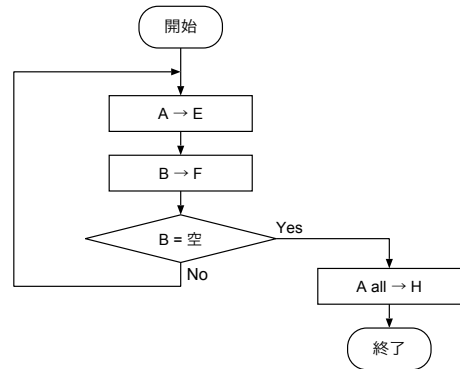
A にダイヤとクラブのカードがある。
2枚ある数値のカードをすべて見つけて H に置く。

図 3 課題 (1) の解答例



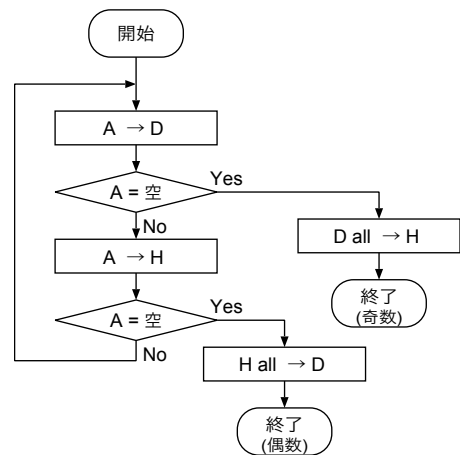
A にあるカードから最も大きい数値の
カードを見つけて H へ置く。

図 4 課題 (2) の解答例



A に m 枚の, B に n 枚のカードがある ($m > n$).
 $m - n$ 枚のカードを H に置く。

図 5 課題 (3) の解答例



A に n 枚のカードがある. n が奇数ならば
A のすべてのカードを H へ, 偶数ならば D へ移す。

図 6 課題 (4) の解答例