# グラフとアルゴリズムとプログラムのやさしいおはなし

渡邉敏正

2021 年 9 月 24 日

**第 6 回**

## 1　グラフ上の基点から連結な点を求めよう

今回は、グラフ $G$ 上の基点 $s$ を固定して、$s$ から連結な点をすべて求めるアルゴリズムを作成すること考えます。そのため

(1) 基礎となる考え方や結果を整理して示すこと
(2) 得られた結果に基づいてアルゴリズムの枠組を作ること

という２つのステップで説明します。

(1) については新しい用語を導入して、いくつかの命題を証明することが中心となりますので、数学的色彩が濃くなります。実例や図を使ってできるだけ分かりやすく説明するつもりです。アルゴリズムを作ることを**アルゴリズムの設計** (design of algorithm) と言います。これには、意図する処理を正しく実行すること (アルゴリズムの正当性) を厳密に示すことが伴います。したがって必然的に数学的にきっちりと説明するという作業が多くなる傾向にあります。

次に (1) の結果に基づいて、基点 $s$ から連結な点をすべて求めるアルゴリズムをまず大まかな形で与えます。これが (2) です。(1) の結果がこのアルゴリズムが正しいこと、すなわちアルゴリズの正当性、を示すことにつながります。

(2) の結果でアルゴリズムのおおまかな形は見えるのですが、実際にアルゴリズムが意図する操作を実行するには手順をさらに詳細かつ具体的に与える必要があります。すなわち、３番目のステップ

（３）さらにより詳細な実際の手順を決めることによりアルゴリズムを完成させること

があります。この (3) のステップは「アルゴリズムの詳細化」と呼ばれることがあります。このステップはアルゴリズムから実際のプログラムを作成することへの橋渡しとなる重要な作業です。しかしそれを説明するためにはさらに準備が必要ですので、今回は (2) までの説明とし、(3) については回を改めて説明することにします。

(1)、(2) のステップについては馴染みのないことだと思われる方が多いと予想します。少しでも理解していただけるように、対象を「グラフ上で基点から連結である点をすべて求める」という直感的に分かりやすい問題にしました。問題自体をわかりやすいものにすることでアルゴリズムの設計や正当性の扱い方が見えるようにする、という狙いです。

## 1.1　基点 $s$ からの距離

グラフ $G$ 上の基点 $s$ から連結な点のすべてを効率よく求めるアルゴリズムは知られていますが、その説明にはいろいろと準備が必要となりますので、ここでは分かりやすさを優先して多くの方が容易に理解できると思われるアルゴリズムを説明することにします。グラフ $G$ における基点 $s$ からの距離という数値を導入して、距離 1 の点の集合、距離 2 の点の集合、$\cdots$ 、と点の集合を求めていくアルゴリズムを考えます。

グラフ $G$ 上の異なる 2 点 $s,t$ に対して、$s$ と $t$ を結ぶ 単純なパスのうち辺数の最も少ないものに着目して、その辺数 (すなわち、このパスの長さ) を $s$ と $t$ の**距離** (distance) と呼び、$dist_G(s,t)$ という記号で表すことにします。$G$ がわかっているときは $G$ を省略して単に $dist(s,t)$ と書きます。$s$ と $t$ が非連結なときは $dist(s,t) = \infty$(無限大) とし、$s = t$ のときは $dist(s,t) = 0$ とします。なお、この着目した辺数最小のパスを $s,t$ 間の**最短パス** (shortest path) といいます。

$G$ 上で、$s$ からの距離が $k$ の点のすべてからなる集合を $A_k$ と表すことにします。$k = 0, 1, \cdots$ です。すぐにわかると思いますが、$i \neq j$ ならば $A_i$ と $A_j$ に共通に含まれる点はありません。このことを記号で

$$A_i \cap A_j = \emptyset \ (i \neq j)$$

と表現します。$\emptyset$ は**空集合** (empty set: 要素を持たない集合) を表す記号です。$A_i \cap A_j$ を $A_i$ と $A_j$ の**共通集合** (intersection of sets) といいます。

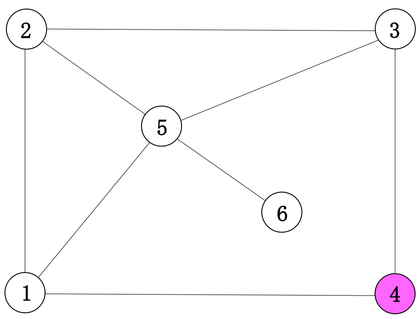


図 44　グラフ $G_4 = (V_4, E_4)$ と基点 $s = 4$

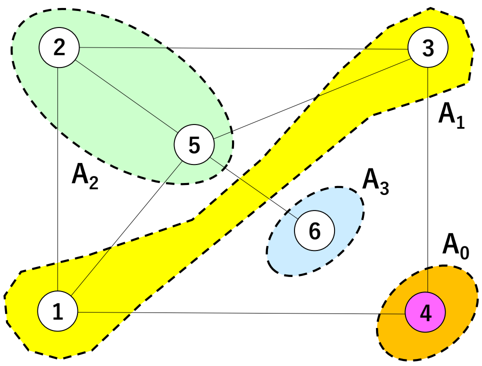


図 45　$G_4 = (V_4, E_4)$ における $s = 4$ から距離が $k$ である点の集合 $A_k$ $(k = 0, 1, 2, 3)$

図 44 のグラフ $G_4$ について、$s = 4$ を基点として $A_k$ ($k \geq 0$ なるすべての整数 $k$ について) を求めると図 45 に示すように

$$A_0 = \{4\},\ A_1 = \{1,3\},\ A_2 = \{2,5\},\ A_3 = \{6\},\ A_j = \emptyset\ (j \geq 4)$$

となります。どの辺もある $A_k$ と $A_{k+1}$ を結ぶ形か、またはある $A_j$ 内の 2 点を結ぶ形か、いずれかであることがわかると思います。

集合 $A_k$ を記号を用いて表すと

$$A_k = \{v \mid dist(s,v) = k\}$$

となります。これは、ある条件を満たす集合を表す一般的な書き方です。縦棒 | の右側部分 $dist(s,v) = k$ が満たすべき条件で、縦棒 | の左側部分の $v$ がこの条件を満たす要素を表し、$A_k$ はこのような点 $v$ の集合である、という意味です。この記号を使うと、例えば上記の $A_2 = \{2,5\}$ は

$$A_2 = \{v \mid dist(4,v) = 2\}$$

と表現できます。記号に少しずつ慣れていただければと思います。

## 1.2 アルゴリズムのポイント

$G$ 上で $s$ から連結な点すべてを集めた集合を $A_G(s)$ と表しましょう。$G$ がわかっているときは単に $A(s)$ と表すことにします。$s \in A(s)$ です。今回の目標は、$A(s)$ を求めるアルゴリズムの設計です。

$A_0 = \{s\}$ として、$s$ に隣接する点を集めると $A_1$ ができます。$A_1$ の点に隣接する点で $A_0$, $A_1$ いずれにも含まれないものを集めれば $A_2$ ができます。以下同様に $k = 3, \cdots$ について、$A_{k-1}$ の点に隣接する点で $A_{k-2}$, $A_{k-1}$ いずれにも含まれないもの集めれば $A_k$ ができます。$A_0, \cdots, A_m$ と進んで最初に $A_{m+1} = \emptyset$ となった時点で、$A_0, \cdots, A_m$ を集めれば $A(s)$ ができる、ということがアルゴリズムのあらすじです。

これは多くの方が直感的にわかることで証明も不要かと思います。しかし、アルゴリズムの正当性に関係しますし、アルゴリズム作成の様子を見ていただくという目的もありますので、分かり切った事柄ですが以下できっちりと説明することにしました。そのため説明の記述が長くなることをお断りしておきます。キーとなる命題の証明を記述していますが、不要とお考えの場合や証明は苦手だという場合にはスキップしていただいて差し支えありません。アルゴリズム作成の流れを伝えることが今回の目的です。

ただ、証明の記述に関連していくつか証明方法の簡単な説明をしました。その部分だけでも目を通していただけるとありがたいです。

## 1.3 $A_k$ の性質について

$A(s)$ の点 $v\,(\neq s)$ に対して、$dist(s,v) = k$ すなわち $v \in A_k\,(k \geq 1)$ とするとき、$s, v$ 間の最短パス ($dist(s,v) = k$ を決めるパス：複数存在するときはどれか 1 本とします)

$$P:\ v_0 = s - v_1 - \cdots - v_{k-1} - v_k = v$$

に着目します。重要なことは、$1 \leq i \leq k-1$ なるすべての $i$ について、

**$P$ 上の $s$ から $v_i$ までの部分は $s, v_i$ 間の最短パスである**

ということです。したがって、$dist(s, v_i) = i$ ですので $v_i \in A_i$ です。すなわち、($i = 0$, $i = k$ の場合も含めて)

$$A_i \neq \emptyset \ (\, 0 \leq i \leq k \text{ なるすべての } i \text{ について})$$

となります。

一方、ある整数 $k\,(\geq 0)$ について

$$A_k \neq \emptyset \ \text{ かつ } \ A_{k+1} = \emptyset$$

としてみましょう。このとき、

$$A_j = \emptyset \ \ (j \geq k+1 \text{ なるすべての } j \text{ について})$$

が成り立つことはすぐに分かると思います。以上のことを次の命題にまとめておきます。

**命題 6.1** $A_k\,(k \geq 0)$ に対して以下が成り立つ。

(1) $A_k \neq \emptyset$ なる $A_k$ が存在すれば $A_j \neq \emptyset\,(\,0 \leq j \leq k$ なるすべての $j$ について)
(2) $A_k = \emptyset\,(\,k \geq 1)$ なる $A_k$ が存在すれば $A_r = \emptyset\,(\,r \geq k$ なるすべての $r$ について)

## 1.4 $A(s)$ と $A_k$ の関係

命題 6.1 によって、$A(s)$ と $A_k$ の関係が見えてきます。以下でそれを説明します。

$G$ のすべての点について $s$ からの距離をみたときの最大値 (有限値) を $m$ としますと、

$$A_k \neq \emptyset\,(k = 0, 1, \cdots, m \text{ について}),\ \ A_r = \emptyset\,(r \geq m+1 \text{ なるすべての } r \text{ について})$$

が成り立ちます。ここで $A_0, \cdots, A_m$ の**和集合** (union of sets：これらに含まれるすべての点を合わせてできる集合) を記号を使って

$$A_0 \cup \cdots \cup A_m$$

と表します。各 $A_k\,(0 \leq k \leq m)$ の点 $v$ については $s, v$ 間の最短パスがありますので、$v \in A(s)$ です。このことは、各 $A_k$ のすべての点が $A(s)$ に含まれることを意味し、記号を使って

$$A_k \subset A(s) \ \ (\,0 \leq k \leq m \text{ なるすべての } k \text{ について}\,)$$

と表します。あるいは

$$A_0 \cup \cdots \cup A_m \subset A(s) \ \ (A_0 \cup \cdots \cup A_m \text{ は } A(s) \text{ に含まれる})$$

と表すこともできます。

一方、$A(s)$ に含まれる点 $u$ については

$$dist(s, u) = k \text{ かつ } 0 \leq k \leq m$$

となる整数 $k$ が決まります。したがって、$A(s)$ の点は $A_0, \cdots, A_m$ のどれかに含まれることになり、

$$A(s) \subset A_0 \cup \cdots \cup A_m \ \ (A(s) \text{ は } A_0 \cup \cdots \cup A_m \text{ に含まれる})$$

を意味します。以上を合わせると

$$A(s) = A_0 \cup \cdots \cup A_m$$

が示されたことになります。

## 1.5　$A_k$ $(k \geq 1)$ を構成するアルゴリズム

目標である $A(s)$ は

$$A(s) = A_0 \cup \cdots \cup A_m$$

なる結果から、$A_0 \cup \cdots \cup A_m$ を求めればよいことがわかりました。次の目標は、$A_0$ から $A_1$ を求め、$A_1$ から $A_2$ を求め、$\cdots$、$A_k$ から $A_{k+1}$ を求め、と続けていき $A_m$ まで求めていくことです。そのためのアルゴリズム **compo(G, s)** を以下に示します。

**compo(G, s)** は、$N_0 \leftarrow \{s\}\,(= A_0)$ と初期設定し、Step3 ( (3.1)〜(3.4) ) で集合 $N_k$ と $R$ を構成する操作を $k = 1, 2 \cdots$ と反復します。$k \geq 1$ なる各整数 $k$ について $N_k = A_k$ であり、このようにして $A_k$ が構成できる、というのがアルゴリズムの主張です。その主張が正しいこと (アルゴリズムの正当性) については後述します。

なお、アルゴリズムの記述の中にある $/ * \cdots * /$ は**コメント文** (comment) とよばれ、アルゴリズムの実行には影響を与えない記述です。 $\cdots$ の部分に種々の注釈や補足的な説明を書きます。[*1]

**compo(G, s)**

/* $G$ が 1 点 $s$ のみの場合は 1 回目の 3.3 で $N_1 = \emptyset$ となり、2 回目の 3.1 において $N_1 = \emptyset$ ですので Step4 に進み終了します。したがって、以下では $G$ が 2 点以上を含むとして説明します */

Step1.　$k \leftarrow 1$ とする;

Step2.　$N_0 \leftarrow \{s\}$, $R \leftarrow N_0$ とする;　/* $A_0 = \{s\}$ であり、$R$ は最終的には $A(s)$ を格納します */

Step3.

- 3.1.　$N_{k-1} \neq \emptyset$ である間は、以下の 3.2〜3.4 を実行する; /* $N_{k-1} = \emptyset$ ならば Step4 進む */
- 3.2.　$N_k \leftarrow \emptyset$ とする; /* 各 $k$ について、$N_k$ は $A_k$ に入るべき要素を保持します */
- 3.3.　$N_{k-1}$ のすべての点 $u_1, \cdots, u_r$ に対して、$i = 1, \cdots, r$ の順に、各 $u_i$ について以下の 拡大操作 を行う;

  (拡大操作)

  - $u_i$ に隣接する点が存在しないときは何もしない;
  - $u_i$ に隣接する点が $v_1, \cdots, v_d\,(d \geq 1)$ のとき、$j = 1, \cdots, d$ の順に、各 $v_j$ について以下の (i) または (ii) を実行する;

    (i)　$k = 1$ のとき、$v_j$ を $N_k$ に加える; /* 図 46 参照 */

    (ii)　$k \geq 2$ のとき、$v_j \notin R \cup N_k$ ならば $v_j$ を $N_k$ に加え、$v_j \in R \cup N_k$ ならば何もしない;[*2]　/* 図 47 参照 */
- 3.4.　もし $N_k \neq \emptyset$ ならば $R \leftarrow R \cup N_k$ とする。その後に $k$ の値を 1 だけ増やして 3.1 に戻る;

Step4.　終了する; /* $N_{k-1} = \emptyset$ */

---

[*1] コメント文を表す記号はプログラミング言語それぞれで決まっています。これは一例です。

[*2] $v_j$ が $R \cup N_k$ に含まれるか否かは、詳細に言えば、$v_j$ が $N_{k-2} \cup N_{k-1} \cup N_k$ に含まれるか否かの判定です。

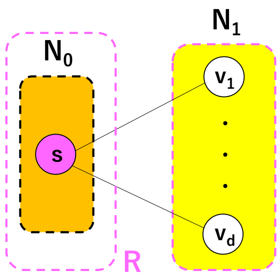


図 46　$N_0 = \{s\}$、および $s\,(= u_1)$ に隣接する点 $v_j$ の集合 $N_1$；まず $R \leftarrow N_0$ と初期設定し、$N_1$ が空でない集合として確定したならば $R \leftarrow R \cup N_1$ と更新します

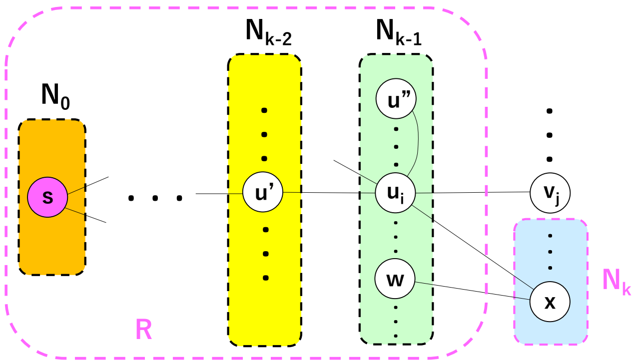


図 47　$N_{k-1}$ の点 $u_i$ に隣接する点 $u' \in N_{k-2} \subset R$, $u" \in N_{k-1} \subset R$, $x \in N_k$ および $v_j \notin R \cup N_k$ (実質的には、$v_j \notin N_{k-2} \cup N_{k-1} \cup N_k$)；このとき、$N_k \leftarrow N_k \cup \{v_j\}$ と更新し、最終的に $N_k$ が空でない集合として確定したならば $R \leftarrow R \cup N_k$ なる更新もします

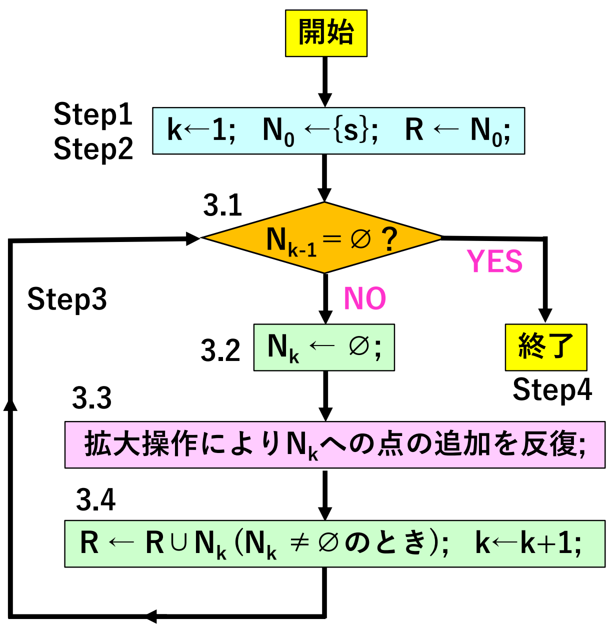


図 48　$N_k$, $R$ に着目した $\mathbf{compo(G,s)}$ の流れ図

$N_k$, $R$ に着目した $\mathbf{compo(G,s)}$ の大まかな流れ図を図 48 に示します。これにより、アルゴリズムの流れが明らかになると思います。

**注意 6.1** (3.1 と 3.4 について) $\mathbf{compo(G,s)}$ の 3.4 で $k$ の値は 1 だけ増えて 3.1 に戻りますので、3.3 で構成された $N_k$ は 3.1 に戻ったときには $N_{k-1}$ として扱われることに留意してください。このことは特に、3.4 で $N_k \neq \emptyset$ の場合には注意が必要です。このような変数や添字を 1 だけ増やしたり減らしたりする操作はアルゴリズムやプログラムでよく出てきます。記号は同じで持っている値が変化する、ということに不慣れな方が多いと思いますが、少しずつ慣れてほしいと思っています。

$\mathbf{compo(G,s)}$ の記述を見てすぐにその動作をイメージできる方はそれほど多くないと思いますので、まず、図 44 のグラフ $G_4$ を例として、実例でその動作を追ってみましょう。そのあとでアルゴリズムがどのような作業をしているかを説明します。最後に、これらをベースにしてアルゴリズムの正当性を示します。

## 1.6 アルゴリズムの実行例

図 44 のグラフ $G_4$ と基点 $s = 4$ に対して $\mathbf{compo(G,s)}$ を実行してみましょう。図 45 に示した $A_0$, $A_1$, $A_2$, $A_3$ がそれぞれ $N_0$, $N_1$, $N_2$, $N_3$ として順次求められていく様子を図 49〜図 52 に示しますので、ゆっくりと各ステップを追ってみてください。皆さんの理解の助けになることを期待しています。

アルゴリズムは以下のように進みます。

$\mathbf{compo(G_4,4)}$ /* $G \leftarrow G_4$, $s \leftarrow 4$ */

Step1. $k \leftarrow 1$;

Step2. $N_0 \leftarrow \{4\},\ R \leftarrow N_0\,(=\{4\})$; (図 49 参照)

Step3.

- 3.1.(1 **回目**) ($k=1$ である) $N_0 \neq \emptyset$ なので 3.2〜3.4 の実行に進む;
- 3.2.(1 **回目**) $N_1 \leftarrow \emptyset$;
- 3.3.(1 **回目**) $N_0 = \{4\}$ の点 $u_1 = 4$ について拡大操作を行う;
  - $u_1 = 4$ に隣接する点 $v_1 = 1, v_2 = 3$ に対して ($k = 1$ なので)
    - (i) $v_1 = 1$ を $N_1$ に加える; /* $N_1 = \{1\}$ */
    - (i) $v_2 = 3$ を $N_1$ に加える; /* $N_1 = \{1,3\}$, $N_1 \cap N_0 = \emptyset$ */ (図 50 参照)
- 3.4.(1 **回目**) $N_1 \neq \emptyset$ なので $R \leftarrow R \cup N_1 (= N_0 \cup N_1 = \{1,3,4\})$ とする。$k \leftarrow 2$ として 3.1 に戻る;

Step3.

- 3.1.(2 **回目**) ($k=2$ である) $N_1 \neq \emptyset$ なので 3.2〜3.4 の実行に進む;
- 3.2.(2 **回目**) $N_2 \leftarrow \emptyset$;
- 3.3.(2 **回目**) $N_1 = \{1,3\}$ の点 $u_1 = 1$, $u_2 = 3$ に対して、$i = 1,2$ の順に、各 $u_i$ について拡大操作を行う; /* $R = \{1,3,4\} = N_0 \cup N_1$ */
  - /* $N_2 = \emptyset$ */ $u_1 = 1$ に隣接する点 $v_1 = 2, v_2 = 4, v_3 = 5$ に対して、($k = 2$ なので)
    - (ii) $v_1 = 2 \notin R \cup N_2$ なので、$v_1 = 2$ を $N_2$ に加える; /* $N_2 = \{2\}$ */
    - (ii) $v_2 = 4 \in R$ なので何もしない; /* $N_2 = \{2\}$ */
    - (ii) $v_3 = 5 \notin R \cup N_2$ なので、$v_3 = 5$ を $N_2$ に加える; /* $N_2 = \{2,5\}$ */

- /*$N_2 = \{2, 5\}$*/ $u_2 = 3$ に隣接する点 $v_1 = 2, v_2 = 4, v_3 = 5$ に対して、($k = 2$ なので)
  - (ii) $v_1 = 2 \in N_2$ なので何もしない; /* $N_2 = \{2, 5\}$ */
  - (ii) $v_2 = 4 \in R$ なので何もしない; /* $N_2 = \{2, 5\}$ */
  - (ii) $v_3 = 5 \in N_2$ なので何もしない; /* $N_2 = \{2, 5\}$, $N_2 \cap (N_0 \cup N_1) = \emptyset$ */ (図 51 参照)

3.4.(2 **回目**) $N_2 \neq \emptyset$ なので $R \leftarrow R \cup N_2 (= N_0 \cup N_1 \cup N_2 = \{1, 2, 3, 4, 5\})$ とする。$k \leftarrow 3$ として 3.1 に戻る;

Step3.

3.1.(3 **回目**) ($k = 3$ である) $N_2 \neq \emptyset$ なので 3.2〜3.4 の実行に進む;

3.2.(3 **回目**) $N_3 \leftarrow \emptyset$;

3.3.(3 **回目**) $N_2 = \{2, 5\}$ の点 $u_1 = 2$, $u_2 = 5$ に対して、$i = 1, 2$ の順に、各 $u_i$ について拡大操作を行う; /* $R = \{1, 2, 3, 4, 5\} = N_0 \cup N_1 \cup N_2$ */

- /*$N_3 = \emptyset$*/ $u_1 = 2$ に隣接する点 $v_1 = 1, v_2 = 3, v_3 = 5$ に対して、($k = 3$ なので)
  - (ii) $v_1 = 1 \in R$ なので何もしない; /* $N_3 = \emptyset$ */
  - (ii) $v_2 = 3 \in R$ なので何もしない; /* $N_3 = \emptyset$ */
  - (ii) $v_3 = 5 \in R$ なので何もしない; /* $N_3 = \emptyset$ */
- /* $N_3 = \emptyset$ */ $u_2 = 5$ に隣接する点 $v_1 = 1, v_2 = 2, v_3 = 3, v_4 = 6$ に対して、($k = 3$ なので)
  - (ii) $v_1 = 1 \in R$ なので何もしない; /* $N_3 = \emptyset$ */
  - (ii) $v_2 = 2 \in R$ なので何もしない; /* $N_3 = \emptyset$ */
  - (ii) $v_3 = 3 \in R$ なので何もしない; /* $N_3 = \emptyset$ */
  - (ii) $v_4 = 6 \notin R \cup N_3$ なので、$v_4 = 6$ を $N_3$ に加える;
    /*$N_3 = \{6\}$, $N_3 \cap (N_0 \cup N_1 \cup N_2) = \emptyset$*/ (図 52 参照)

3.4.(3 **回目**) $N_3 \neq \emptyset$ なので $R \leftarrow R \cup N_3 (= N_0 \cup N_1 \cup N_2 \cup N_3 = \{1, 2, 3, 4, 5, 6\})$ とする。$k \leftarrow 4$ として 3.1 に戻る;

Step3.

3.1.(4 **回目**) ($k = 4$ である) $N_3 \neq \emptyset$ なので 3.2〜3.4 の実行に進む;

3.2.(4 **回目**) $N_4 \leftarrow \emptyset$;

3.3.(4 **回目**) $N_3 = \{6\}$ の点 $u_1 = 6$ について拡大操作を行う; /* $R = \{1, 2, 3, 4, 5, 6\}$ */

- /*$N_4 = \emptyset$*/ $u_1 = 6$ に隣接する点 $v_1 = 5$ に対して、($k = 4$ なので)
  - (ii) $v_1 = 5 \in R$ なので何もしない; /* $N_4 = \emptyset$ */

3.4.(4 **回目**) $N_4 = \emptyset$ なので $R$ の更新はしない。$k \leftarrow 5$ として 3.1 に戻る;

Step3.

3.1.(5 **回目**) ($k = 5$ である) $N_4 = \emptyset$ なので Step4 に進む;

Step4. 終了する;

/*$N_0 = \{4\}$, $N_1 = \{1, 3\}$, $N_2 = \{2, 5\}$, $N_3 = \{6\}$, $N_4 = \emptyset$*/

/*$N_i \cap N_j = \emptyset$ ($0 \leq i, j \leq 3$, $i \neq j$ なるすべての整数 $i, j$ について)*/

/*$R = N_0 \cup N_1 \cup N_2 \cup N_3$*/

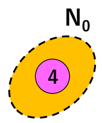


図 49　Step2 終了時の状況 ($R$ は省略)

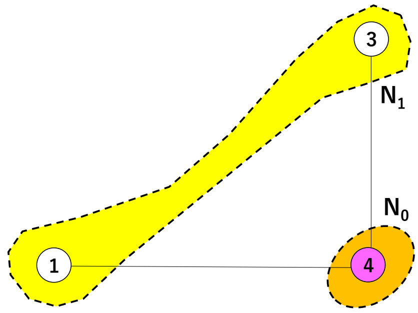


図 50　1 回目の 3.3 終了時の状況 ($R$ は省略)

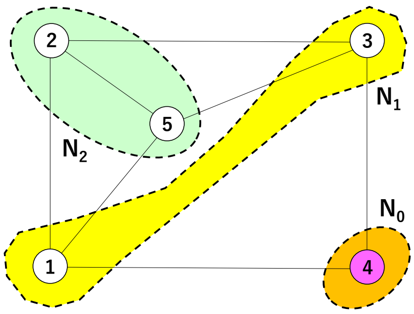


図 51　2 回目の 3.3 終了時の状況 ($R$ は省略)

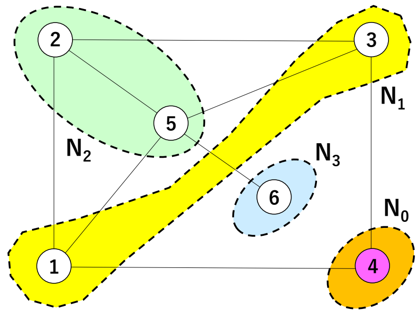


図 52　3 回目の 3.3 終了時の状況 ($R$ は省略)

## 1.7　アルゴリズムの動作について

**compo(G, s)** は、$N_0 \leftarrow \{s\}\,(= A_0)$, $R \leftarrow N_0$ と初期設定し、Step3 ( (3.1)〜(3.4) ) で集合 $N_k$ と $R$ を構成する操作を $k = 1, 2 \cdots$ と反復します。$k$ 回目の Step3 に着目して、$k = 1, 2 \cdots$ と反復する場合にどのような動作をするか見ていきます。

$k = r$ のとき、3.3 の終了時に $N_r = \emptyset$ として説明します。まず、$N_0 = \{s\}$ ですから

(1) $r \geq 1$

です。この場合は次の 3.4 で $k \leftarrow r + 1$ として ( $R$ の更新はせずに ) 3.1 に戻ります。その 3.1 では $N_r = \emptyset$ $(k - 1 = r)$ なので Step4 に進み、アルゴリズムは終了します。したがって、以下が成り立ちます。

(2) $0 \leq i \leq r - 1$ なるすべての整数 $i$ について、$i$ 回目の 3.3 終了時には ( したがってアルゴリズム終了後も ) $N_i \neq \emptyset$

また、アルゴリズム終了後については

(3) $N_i = \emptyset$ ( $i \geq r$ なるすべての整数 $i$ について )

とします。

次に、$k$ 回目の Step3 の $N_k$ に着目しましょう。**compo(G, s)** の拡大操作から、3.3 終了時の $N_k$ は次のような集合であることがわかります。$k \geq 2$ のときは $N_{k-1} = \emptyset$, $N_{k-1} \neq \emptyset$ の場合を合わせて記述しています。

- $k=1$ のとき

  $N_1 = \{v_j \mid v_j \neq s$ かつ $[v_j$ は $s$ に隣接する$]\}$ (図 46 参照)

  ここで、$k \geq 2$ の場合の表現と合わせるために $N_1$ を次のように表現しておきます：

  $N_1 = \{\, v_j \mid v_j \notin R\,(= N_0)$ かつ $[N_0$ の中に $v_j$ に隣接する点がある$]\,\}$

- $k \geq 2$ のとき ($r \geq 2$ とします)

  $N_{k-1} = \emptyset$ ならば

  $N_i = \emptyset$ ($i \geq k-1$ なるすべての整数 $i$ について)

  $N_{k-1} \neq \emptyset$ ならば

  $N_k = \{\, v_j \mid v_j \notin R$ かつ $[N_{k-1}$ の中に $v_j$ に隣接する点がある$]\,\}$ (図 47 参照)

なお、3.3 (ii) においては、$v_j$ が $R \cup N_k$ に入るかどうか (実質的に $v_j$ が $N_{k-2} \cup N_{k-1} \cup N_k$ に入るかどうか) をチェックしていますが、これはアルゴリズムの操作として, $v_j \notin R$ なる $v_j$ がまだ $N_k$ に入っていない場合に $N_k$ に加えることを表しています。いま考えている $N_k$ を集合として表現する場合には、つまり、集合がどのような要素から構成されるかを表す場合には特に「まだ $N_k$ に入っていないこと」を記載する必要はありません。

$k=1$ の場合と $k \geq 2$ の場合を合わせると、$N_{k-1} \neq \emptyset$ のときの $N_k$ は以下のようになります。

$$N_k = \{\, v \mid v \notin R \text{かつ} [N_{k-1} \text{の中に } v \text{ に隣接する点がある}]\,\}$$

**注意 6.2** 図 47 では、$R = N_0 \cup \cdots \cup N_{k-1}$ としていますが、このことはあとで証明しますので、この段階では $R$ のみに注目してください。

**注意 6.3** $N_k$ 内の異なる 2 点が隣接する可能性はありますが、この隣接性は $N_k$ の構成に関与しません。

$k$ 回目の Step3 の 3.1 開始時には、$k-1$ 回目の Step3 で構成された $R$ を引き継ぎ、3.2 では $N_k = \emptyset$ です。そして 3.3 ではこの $R$ を使って

$v_j \notin R$ かつ $[N_{k-1}$ の中に $v_j$ に隣接する点がある$]$

を満たす点 $v_j$ を $N_k$ に加えていきます。したがって、3.3 終了時 ( 3.4 開始時 ) には $N_k \neq \emptyset$ であっても

$N_k \cap R = \emptyset$

が成り立ちます。続いて 3.4 において ($N_k = \emptyset$, $N_k \neq \emptyset$ いずれでも) $R \leftarrow R \cup N_k$ となります。なお、$N_k = \emptyset$ のときには実質的には更新は生じません。

ここでのポイントは、3.3 における $N_k$ への点 $v_j$ の追加の仕方により、3.3 終了時では $N_k \cap R = \emptyset$ ということです。この $R$ は $k-1$ 回目の Step3 で構成された集合です。

これから以下のことを証明しましょう。集合 $R$ の具体的な形が明らかになります。

**命題 6.2** $1 \leq k \leq r$ なるすべての整数 $k$ について以下が成り立つ。

$k$ 回目の 3.1 開始時には

$R = N_0 \cup \cdots \cup N_{k-1}$

であり、3.3 終了時には $N_k \cap R = \emptyset$ である。さらに 3.4 において

$R = N_0 \cup \cdots \cup N_{k-1} \cup N_k$

となって $k+1$ 回目の 3.1 に進む。

なお、3.3 終了時に $N_k \cap R = \emptyset$ であることはすでに説明しましたが、以下では再度これにも言及します。また、3.3 終了時に $N_k = \emptyset$ であることも含まれています。主に $R$ と $N_0, \cdots, N_k$ の関係に焦点を当てて、以下に証明を述べます。

(**帰納法のベース**：$k=1$ のとき) 1 回目の 3.1 開始時には $R = N_0 = \{s\}$ であり、3.3 終了時には $N_1 \cap R (= N_1 \cap N_0) = \emptyset$、3.4 で $R \leftarrow R \cup N_1 (= N_0 \cup N_1)$ となって 2 回目の 3.1 に進みます。

(**帰納法の仮定**：$k=i,\ 1 \le i \le r-1,$ のとき) 以下のことが成り立つと仮定します。

> $i$ 回目の 3.1 開始時には $R = N_0 \cup \cdots \cup N_{i-1}$ で、3.3 終了時には $N_i \cap R (= N_i \cap (N_0 \cup \cdots \cup N_{i-1})) = \emptyset$、3.4 であり $R \leftarrow R \cup N_i (= N_0 \cup \cdots \cup N_{i-1} \cup N_i)$ となって $i+1$ 回目の 3.1 に進む。

(**帰納ステップ**：$k=i+1,\ 1 \le i \le r-1,$ のとき) $i+1$ 回目の 3.1 開始時には $R = N_0 \cup \cdots \cup N_{i-1} \cup N_i$ です。3.3 終了時には $N_{i+1} \cap R (= N_{i+1} \cap (N_0 \cup \cdots \cup N_i)) = \emptyset$ であり、3.4 において $R \leftarrow R \cup N_{i+1} (= N_0 \cup \cdots \cup N_i \cup N_{i+1})$ となって $i+2$ 回目の 3.1 に進みます。

以上のことからまず帰納法のベースにより、$k=1$ のときに命題が成り立ちます。帰納法の仮定と帰納ステップにより、$k=2$ のときにも命題が成り立ちます。同様に帰納法の仮定と帰納ステップにより、$k=3$ のときも命題が成り立ち、$\cdots$ と繰り返すことにより、$1 \le k \le r$ なるすべての整数 $k$ について命題が成り立つことになります。以上が命題の証明です。 (証明終)

$k=r$ のとき、3.3 終了時に $N_r = \emptyset$ となった場合について、$\mathbf{compo(G,s)}$ における Step3 の 3.1〜3.4 に焦点を当てて、$k=1,2,3$ の場合と終了前後の $N_k, R$ の様子を表 2 にまとめておきます。表はわかりやすいように例題に合わせて $r \ge 3$ の場合のイメージで記述しています。

**注 6.4** 整数 $k \ge 1$[*3] を含む命題 Prop(k) について、

| | | |
|---|---|---|
| (**帰納法のベース**) | $k=1$ のとき | Prop(1) が成り立つ |
| (**帰納法の仮定**) | $k \ge 2$ のとき | Prop(k-1) が成り立つと仮定する |
| (**帰納ステップ**) | $k \ge 2$ のとき | この仮定の下で Prop(k) が成り立つ |

ということを示すことによって、$k \ge 1$ なるすべての整数 $k$ について Prop(k) が成り立つことを証明する方法を ($k$ に関する) **数学的帰納法** (mathematical induction ( on $k$ )) とよびます。

なお、数学的帰納法の説明として、「Prop(k-1) が成り立つと仮定する」の部分を「$1 \le i \le k-1$ なるすべての整数 $i$ について Prop(i) が成り立つと仮定する」と置き換えた表現もあります。

## 1.8 アルゴリズムの正当性

アルゴリズム $\mathbf{compo(G,s)}$ の正当性は直感的に理解できることだと思います。しかしながら所望の動作をしていることを保証するためにその正当性を記述します。内容の性質上、数学的色彩が濃くなります。証明は不要とお考えの場合や証明は苦手だという方はスキップして差し支えありません。

$\mathbf{compo(G,s)}$ は、$N_0 \leftarrow \{s\} (= A_0)$ と初期設定をし、$k \ge 1$ なる整数 $k$ について、集合 $N_k$ と $R$ を構成する Step3 を反復します。$k$ 回目の Step3 は $N_k = \emptyset$ で開始し、3.3 で $N_{k-1}$ の各点 $v$ について $v$ に隣接する点で $R$ に含まれていないものを加えることを反復して $N_k$ を構成します。

[*3] $k \ge 0$ として説明する場合もあります。

表 2　**compo(G, s)** における $N_k$, $R$ の構成および終了時の状況について

| | |
|---|---|
| 初期設定 | Step1. $k \leftarrow 1$;　Step2. $N_0 \leftarrow \{s\},\ R \leftarrow N_0$; |
| Step3 $(k=1)$ | (3.1) $N_0 \neq \emptyset$;<br>(3.2〜3.3) $N_1 \cap R = \emptyset$ なる $N_1$ を構成;　/* $N_1 \cap N_0 = \emptyset$ */<br>(3.4) $N_1 \neq \emptyset$ なので $R \leftarrow R \cup N_1\ (= N_0 \cup N_1),\ k \leftarrow k+1$;<br>/* $N_1 = \emptyset$ ならば $R$ の更新はなし ($R = N_0$ のまま) */ |
| Step3 $(k=2)$ | (3.1) $N_1 \neq \emptyset$;<br>(3.2〜3.3) $N_2 \cap R = \emptyset$ なる $N_2$ を構成;　/* $N_2 \cap (N_0 \cup N_1) = \emptyset$ */<br>(3.4) $N_2 \neq \emptyset$ なので $R \leftarrow R \cup N_2\ (= N_0 \cup N_1 \cup N_2)$,　$k \leftarrow k+1$;<br>/* $N_2 = \emptyset$ ならば $R$ の更新はなし ($R = N_0 \cup N_1$ のまま) */ |
| Step3 $(k=3)$ | (3.1) $N_2 \neq \emptyset$;<br>(3.2〜3.3) $N_3 \cap R = \emptyset$ なる $N_3$ を構成;　/* $N_3 \cap (N_0 \cup N_1 \cup N_2) = \emptyset$ */<br>(3.4) $N_3 \neq \emptyset$ なので $R \leftarrow R \cup N_3\ (= N_0 \cup N_1 \cup N_2 \cup N_3)$,　$k \leftarrow k+1$;<br>/* $N_3 = \emptyset$ ならば $R$ の更新はなし ($R = N_0 \cup N_1 \cup N_2$ のまま) */ |
| | $\vdots$ |
| Step3 $(k=i)$ | (3.1) $N_{i-1} \neq \emptyset$;<br>(3.2〜3.3) $N_i \cap R = \emptyset$ なる $N_i$ を構成;　/* $N_i \cap (N_0 \cup \cdots \cup N_{i-1}) = \emptyset$ */<br>(3.4) $N_i \neq \emptyset$ なので $R \leftarrow R \cup N_i\ (= N_0 \cup \cdots \cup N_{i-1} \cup N_i)$,　$k \leftarrow k+1$;<br>/* $N_i = \emptyset$ ならば $R$ の更新はなし ($R = N_0 \cup \cdots \cup N_{i-1}$ のまま) */ |
| | $\vdots$ |
| Step3 $(k=r)$ | (3.1) $N_{r-1} \neq \emptyset$;<br>(3.2〜3.3) $N_r \cap R = \emptyset$ なる $N_r$ を構成;　/* $N_r \cap (N_0 \cup \cdots \cup N_{r-1}) = \emptyset$ */<br>(3.4) $N_r = \emptyset$ なので $R$ の更新はせずに ($R = N_0 \cup \cdots \cup N_{r-1}$ のまま),　$k \leftarrow k+1$;<br>/* $N_r \neq \emptyset$ ならば $R \leftarrow R \cup N_r\ (= N_0 \cup \cdots \cup N_{r-1} \cup N_r)$ */ |
| Step3 $(k=r+1)$ | (3.1) $N_r = \emptyset$;　Step4 に進み、終了する;<br>/* $N_i \neq \emptyset$ ($0 \leq i \leq r-1$ なるすべての整数 $i$ について) */<br>/* $N_i \cap N_j = \emptyset$ ($0 \leq i, j \leq r-1,\ i \neq j$ なるすべての整数 $i, j$ について) */<br>/* $R = N_0 \cup \cdots \cup N_{r-1}$ */ |

これまでと同様に、$k=r$ のとき、3.3 の終了時に (したがって、3.4 の終了時にも) $N_r = \emptyset$ とします。まず、

$$N_i = \emptyset \quad (i \geq r \text{ なるすべての整数 } i \text{ について})$$

です。次に、$0 \leq k \leq r-1$ のとき、$k$ 回目の Step3 の $N_k$ を考えます。3.3 終了時では

$$N_k = \{\, v \mid v \notin R\,(= N_0 \cup \cdots \cup N_{k-1})\ \text{かつ}\ [N_{k-1}\ \text{の中に}\ v\ \text{に隣接する点がある}]\,\} \neq \emptyset$$

です。

**compo(G, s)** が作成するのは $N_k$ であり、構成したいターゲットは $A_k = \{v \mid dist(s,v) = k\}$ なる集合です。$N_0 = \{s\} = A_0$ ですので、アルゴリズムの正当性としては以下が成り立つことを示すことになります。

$N_k$ が真に $A_k$ 等しい ($k \geq 1$ なるすべての整数 $k$ について)

このことは次の (a) と (b) がともに成り立つことと同じです。

(a) $v \in N_k$ ならば $dist(s,v) = k$
(b) $dist(s,v) = k$ ならば $v \in N_k$

(a) は $N_k \subset A_k$ を、(b) は $A_k \subset N_k$ をそれぞれ意味しますので、$N_k = A_k$ となります。

これから述べる正当性の証明は以下の命題 (1)〜(k-1) を示すことがポイントです。ここで $2 \leq k \leq r-1$ とします。

(1) $N_1 = A_1$ ならば $N_2 = A_2$ である
(2) $N_1 = A_1,\, N_2 = A_2$ ならば $N_3 = A_3$ である
(3) $N_1 = A_1,\, N_2 = A_2,\, N_3 = A_3$ ならば $N_4 = A_4$ である
$\cdots$
(i) $N_1 = A_1,\, N_2 = A_2, \cdots, N_i = A_i$ ならば $N_{i+1} = A_{i+1}$ である
$\cdots$
(k-1) $N_1 = A_1,\, N_2 = A_2, \cdots, N_{k-1} = A_{k-1}$ ならば $N_k = A_k$ である

これらの命題 (1)〜(k-1) を証明し、かつ $N_1 = A_1$ が確かに成り立つことがわかったとしますと、目標の

(1) により $N_2 = A_2$, (2) により $N_3 = A_3$, (3) により $N_4 = A_4$, $\cdots$, (k-1) により $N_k = A_k$

が成り立つことになります。ここで、命題 (1)〜(k-1) は以下のような形にまとめることができます。

(i) $N_1 = A_1,\, N_2 = A_2, \cdots, N_i = A_i$ ならば $N_{i+1} = A_{i+1}$ である ($1 \leq i \leq k-1$ なる整数 $i$ について)

したがって、証明は、$1 \leq i \leq k-1$ なる整数 $i$ について、以下のようになります。

(i) $N_1 = A_1$ を示す
(ii) $N_1 = A_1,\, N_2 = A_2, \cdots, N_i = A_i$ を仮定する
(iii) (ii) の仮定が成り立つならば $N_{i+1} = A_{i+1}$ であることを示す

すなわち、($k$ に関する) 数学的帰納法による証明です。

補足の説明をします。前述の通り、正当性の証明の途中で、$k \geq 2$ のときに $1 \leq i \leq k-1$ なるすべての整数 $i$ について $i$ 回目の 3.3 終了時に $N_i = A_i \neq \emptyset$ が成り立つことを仮定します。$i$ 回目の 3.3 終了時には $R = N_0 \cup \cdots \cup N_{i-1}$ かつ $N_i \cap R = \emptyset$ で、3.4 終了時には $R = N_0 \cup \cdots \cup N_{i-1} \cup N_i$ です。重要なことは、この仮定の下では $N_i$ の構成によって $A_i = \{u \mid dist(s,u) = i\}$ が正確に求められており、$1 \leq i \leq k-1$ なるすべての整数 $i$ について以下の (a)' と (b)' がともに成り立つことです。

(a)' $v \in N_i$ ならば $dist(s,v)=i$

(b)' $dist(s,v)=i$ ならば $v \in N_i$

そしてこの仮定のもとで $k$ 回目の 3.4 終了時に $N_k = A_k$ ( $k \le r-1$ ならば $N_k \ne \emptyset$, $k=r$ ならば $N_k=\emptyset$ ) が成り立つことを示します。

では、以下の命題が成り立つことを証明しましょう。

**命題 6.3** $N_0 = \{s\}$ とするとき、$r \ge 1$ なるある $r$ について、$r+1$ 回目の 3.1 において $N_r = \emptyset$ (つまり、$r$ 回目の 3.4 終了時に $N_r=\emptyset$) で Step4 に進み $\mathbf{compo(G,s)}$ が終了したとする。このとき以下が成り立つ。

(1) $N_k = A_k\ (\ne \emptyset)$ ($0 \le k \le r-1$ なるすべての整数 $k$ について)

(2) $N_j = \emptyset = A_j$ ($j \ge r$ なるすべての整数 $j$ について)

(3) $R = N_0 \cup \cdots \cup N_{r-1} = A(s)$

命題の言っていることは直感的に「そうだろうな」とわかると思います。ただ、当たり前のようなことでもアルゴリズムの正当性ですので証明を示しています。なお、$N_0 = A_0\,(\ne \emptyset)$ ですので、以下では $k \ge 1$ のとき、$k$ 回目の 3.4 終了時に $N_k = A_k$ となることを示します。

いま $r \ge 1$ として $r+1$ 回目の 3.1 において $N_r = \emptyset$ で Step4 に進み $\mathbf{compo(G,s)}$ が終了していますので、直前の $r$ 回目の 3.4 終了時に $N_r=\emptyset$ で、アルゴリズム終了後は $j \ge r$ なるすべての整数 $j$ について $N_j = \emptyset$ ということです。

まず、$r=1$ とします。このとき、$N_1=\emptyset$ ですので、$G$ は 1 点のみであるか、もし $G$ が 2 点以上ならば $s$ と異なる点は $s$ に隣接していません。($\mathbf{compo(G,s)}$ については $G$ が 2 点以上の場合を対象とした説明をしていますが、$G$ が 1 点の場合にも適用できます。) したがって、$A_j = \emptyset = N_j$ ( $j \ge r = 1$ なるすべての整数 $j$ について ) となります。また、$R = N_0 = A_0 = \{s\} = A(s)$ です。よって、(1)、(2)、(3) が成り立ちます。

次に、$r \ge 2$ とします。はじめに (1) を証明します。

$r \ge 2$ ですので、それまでの 1 回目 , $\cdots$ , $r-1$ 回目の 3.3 終了時ではそれぞれ $N_1 \ne \emptyset, \cdots, N_{r-1} \ne \emptyset$ です。したがって (命題 6.2 より)、$1 \le k \le r-1$ なるすべての整数 $k$ について、$k$ 回目の 3.3 終了時 ( かつ 3.4 開始前 ) には

$$R = N_0 \cup \cdots \cup N_{k-1} \quad (0 \le i,j \le k-1 \text{ かつ } i \ne j \text{ ならば } N_i \cap N_j = \emptyset \text{ である})$$
$$N_k = \{\, v \mid v \notin R \text{ かつ } [N_{k-1} \text{ の中に } v \text{ に隣接する点がある}]\,\}\ (\ne \emptyset)$$

です。このとき、3.4 終了時には $N_k = A_k$ であることを $k$ に関する数学的帰納法で証明します。$N_k = A_k$ が成り立つことを示すために、以下の (i) と (ii) がともに成り立つことを示します。

(i) $v \in N_k$ ならば $dist(s,v)=k$

(ii) $dist(s,v)=k$ ならば $v \in N_k$

**帰納法のベース** ($k=1$ のとき) $N_1 = \{\, v \mid v \ne s$ かつ $[\,v$ は $s$ に隣接する$\,]\,\} = \{\, v \mid v \notin R\,(=N_0)$ かつ $[N_0$ の中に $v$ に隣接する点がある$]\,\}\ (\ne \emptyset)$ です (図 46 参照)。よって、$v \in N_1$ ならば $dist(s,v)=1$ です。つまり $N_1 \subset A_1$ です。逆に、$dsit(s,v)=1$ とします。$v$ は $s$ に隣接しますので、3.3 (i) で $v$ は $N_1$ に加えられますから、$v \in N_1$ です。すなわち $A_1 \subset N_1$ となります。したがって、3.3 終了時 ( および 3.4 終了時 ) には

$$N_1 = A_1 \ (\neq \emptyset)$$

です。

**帰納法の仮定**　($k \geq 2$ のとき)： ここで $1 \leq i \leq k-1$ なるすべての整数 $i$ について、3.4 終了時に

$$N_i = A_i$$

が成り立っていると仮定します。

**帰納ステップ**　($k \geq 2$ のとき)： 上記の仮定の下で $k$ 回目の 3.3 終了時 ( かつ 3.4 開始前 ) の $N_k$ に着目します。この時点ではまだ、$R = N_0 \cup \cdots \cup N_{k-1}$ です。表 3 を見てください。

表 3　$2 \leq k \leq r-1$ のときの帰納法の仮定 (1 回目〜$k-1$ 回目) と帰納ステップ ($k$ 回目)

| | | | |
|---|---|---|---|
| | 初期設定 | $N_0 = \{s\} = A_0$ | $R = N_0$ |
| | 1 回目の 3.3 終了時 | $N_1 = A_1 \neq \emptyset$ | $N_1 \cap R = N_1 \cap N_0 = \emptyset$ |
| | 1 回目の 3.4 終了時 | | $R \leftarrow N_0 \cup N_1$ |
| 帰納法の仮定 | $\vdots$ | $\vdots$ | |
| | | | $R \leftarrow N_0 \cup \cdots \cup N_{k-2}$ |
| | $k-1$ 回目の 3.3 終了時 | $N_{k-1} = A_{k-1} \neq \emptyset$ | $N_{k-1} \cap R = N_{k-1} \cap (N_0 \cup \cdots \cup N_{k-2}) = \emptyset$ |
| | $k-1$ 回目の 3.4 終了時 | | $R \leftarrow N_0 \cup \cdots \cup N_{k-2} \cup N_{k-1}$ |
| 帰納ステップ | $k$ 回目の 3.3 終了時 | $N_k \neq \emptyset$ | $N_k \cap R = N_k \cap (N_0 \cup \cdots \cup N_{k-2} \cup N_{k-1}) = \emptyset$ |
| | $k$ 回目の 3.4 終了時 | | $R \leftarrow N_0 \cup \cdots \cup N_{k-1} \cup N_k$ |
| | $\vdots$ | $\vdots$ | |
| | | | $R \leftarrow N_0 \cup \cdots \cup N_{r-2}$ |
| | $r-1$ 回目の 3.3 終了時 | $N_{r-1} \neq \emptyset$ | $N_{r-1} \cap R = N_{r-1} \cap (N_0 \cup \cdots \cup N_{r-2}) = \emptyset$ |
| | $r-1$ 回目の 3.4 終了時 | | $R \leftarrow N_0 \cup \cdots \cup N_{r-2} \cup N_{r-1}$ |
| | $r$ 回目の 3.3 終了時 | $N_r = \emptyset$ | |
| | $r$ 回目の 3.4 終了後 | | $r+1$ 回目の 3.1 から Step4 に進み、終了する |
| | | | /* $N_0 \neq \emptyset, N_1 \neq \emptyset, \cdots, N_{r-1} \neq \emptyset$ */ |
| | | | /* $R = N_0 \cup \cdots \cup N_{r-1}$ */ |

いま $v \in N_k$ とし、$v$ に隣接する $N_{k-1}$ の点を $u$ とします ( 図 47 で $u \leftarrow u_i \in N_{k-1}$ とし, $v_j$ あるいは $x$ を $v$ と考えてみてください)。仮定により、$1 \leq i \leq k-1$ なるすべての整数 $i$ について各 $N_i$ は $A_i = \{v' \mid dist(s, v') = i\}$ として正確に求められていて $R = N_0 \cup \cdots \cup N_{k-1} = A_0 \cup \cdots \cup A_{k-1}$ ですので、

$$dist(s, u) = k-1, \ v \notin R$$

です。したがって

$$dist(s,v) = dist(s,u) + 1 = k$$

となります。このことは $N_k$ のすべての点は $A_k$ に含まれること、すなわち $N_k \subset A_k$ を意味します。

逆に、$v$ を $dist(s,v) = k$ なる任意の点とします。$s, v$ 間の最短パス

$$v_0 = s - v_1 - \cdots, - v_{k-1} - v_k = v$$

に着目すれば、このパス上で $s$ から $v_{k-1}$ の部分は $s, v_{k-1}$ 間の最短パスですから

$$dist(s, v_{k-1}) = k - 1$$

です。再び仮定から、$1 \leq i \leq k-1$ なるすべての整数 $i$ について各 $N_i$ は $A_i = \{v' \mid dist(s,v') = i\}$ に等しく、かつ $R = N_0 \cup \cdots \cup N_{k-1} = A_0 \cup \cdots \cup A_{k-1}$ ですので、

$$v_{k-1} \in N_{k-1},\ \ v \notin R$$

です。したがって、3.3 (ii) でまだ $v \notin N_k$ であったならば $v$ は $N_k$ に加えられますから、3.3 終了時には必ず

$$v \in N_k$$

となります。したがって $A_k \subset N_k$ です。以上で、3.3 終了時 ( したがって 3.4 終了時 ) には

$$N_k = A_k$$

であることが示されました。ここまでの説明をまとめますと、$1 \leq k \leq r-1$ なる $k$ 回目の 3.4 終了時の $N_k\,(\neq \emptyset)$ に対して次のことを示したことになります。

| | | |
|---|---|---|
| (**帰納法のベース**) | $k = 1$ のとき | $N_1 = A_1$ |
| (**帰納法の仮定と帰納ステップ**) | $k \geq 2$ のとき | $1 \leq i \leq k-1$ なるすべての整数 $i$ について<br>$N_i = A_i$ が成り立つならば $N_k = A_k$ が成り立つ |

帰納法のベース ($k = 1$) で $N_1 = A_1$ が成り立ちますので、$k = 2$ とすると帰納法の仮定と帰納ステップにより $N_2 = A_2$ となります。したがって、$k = 3$ とすると再び帰納法の仮定と帰納ステップにより $N_3 = A_3$ となります。同様に考えていけば、以下が成り立つことを証明したことになります。

$$N_k = A_k\,(\neq \emptyset)\ \ (\,1 \leq k \leq r-1 \text{ なるすべての整数 } k \text{ について}\,)$$

次に、(2) を証明します。$r$ 回目の 3.3 終了時 ( および 3.4 終了時 ) における $N_r = \emptyset$ に着目します。(このあと $r+1$ 回目の 3.1 で Step4 に進み、アルゴリズムは終了します。) このとき、$N_j = \emptyset$ ($j \geq r$ なるすべての整数 $j$ について) です。また、(1) の証明から $N_{r-2} = A_{r-2}$ かつ $N_{r-1} = A_{r-1}$ が成り立っています。

$r$ 回目の 3.3 終了時では $N_r = \emptyset$ かつ $N_i \neq \emptyset$ ( $i \leq r-1$ なるすべての整数 $i$ について ) ということですので、$N_{r-1} = A_{r-1}\,(\neq \emptyset)$ の中の点に隣接する点 $v$ が存在すれば、$v \in N_{r-2} = A_{r-2}$ または $v \in N_{r-1} = A_{r-1}$ です ( $k = r$ とし、$v$ として図 47 の $u' \in N_{k-2}$ または $u" \in N_{k-1}$ を考えてみてください)。したがって、$dist(s,v) \leq r-1$ となります。すなわち、$A_r = \emptyset$ です。[*4]よって、命題 6.1 (2) を考え合わせれば

[*4] 別の考え方としては、$v \in A_r$ と仮定すると、$s, v$ 間の最短パス $v_0 = s - v_1 - \cdots, v_{r-1} - v_r = v$ に着目すれば、$v_{r-1} \in A_{r-1} = N_{r-1}$, $v \notin A_{r-2} = N_{r-2}$, $v \notin A_{r-1} = N_{r-1}$ (すなわち、$v \notin R = N_0 \cup \cdots \cup N_{r-1}$) ですので、もしまだ $v \notin N_r$ ならば $r$ 回目の 3.3 で $v \in N_r$ となって、この 3.3 終了時に $N_r = \emptyset$ であるという前提に矛盾します。よって、$A_r = \emptyset$ が成り立ちます。このように、証明しようとする命題に対して、その否定形の命題が成り立つと仮定すると矛盾が生じることを示すことにより元の命題を証明する方法を**背理法** (proof by contradiction) とよびます。

$$N_j = \emptyset \ = A_j \, (j \geq r \text{ なるすべての整数 } j \text{ について})$$

となります。したがって (2) が成り立ちます。

(1) と (2) を合わせれば

$$R = N_0 \cup \cdots \cup N_{r-1} = A_0 \cup \cdots \cup A_{r-1} = A(s)$$

すなわち、(3) が成り立ちます。

以上で $r \geq 2$ のときにも (1)、(2)、(3) が成り立つことが示され、命題の証明が完了しました。(証明終)

## 1.9 まとめ

今回は、アルゴリズムを設計するとはどういうことか、について解説することを目指しました。求める対象を明確に規定してそれを求めるためのアルゴリズムを作成し、さらにその正当性を示すという流れです。

具体的には、グラフ $G$ 上の基点 $s$ から連結である点すべてを集めた集合 $A(s)$ を求めるアルゴリズム $\mathbf{compo(G, s)}$ の設計を例として説明しました。$N_0 = \{s\}$ として、$N_0$ の隣接点集合[*5] $N_1$ を構成して基点から距離 1 の点の集合 $A_1$ を求め、$N_1$ の隣接点集合 $N_2$ を構成して基点から距離 2 の点の集合 $A_2$ を求め、という具合に「$N_{k-1}$ の隣接点集合 $N_k$ を構成して基点から距離 $k$ の点の集合 $A_k$ を求める」という拡大操作の反復で $A(s)$ を求めるアルゴリズムを作成し、さらにその正当性を示すという一連の作業を説明しました。

アルゴリズムの正当性は直感的にわかることで証明も不要と思いましたが記述しました。アルゴリズムの設計の各ステップを具体的な事例で皆さんに説明するという意図からできるだけわかりやすい説明を心がけたつもりですが、意図通りになっているかどうかわかりません。当初、集合などに関する記号をできるだけ使わないで説明することを試みましたが、逆にまわりくどくて結果的にわかりにくい表現になってしまうと判断し、最小限の記号を使って記述することにしました。

今回はアルゴリムズムの枠組みは与えましたが、まだプログラムを意識した「アルゴリズムの詳細化」さらには「プログラム作成」という重要なステップが残っています。次回以降にその説明をしていきたいと思っています。

[*5] ここでは便宜的に $N_i = \{v \mid v \notin R = N_0 \cup \cdots \cup N_{i-1}$ かつ $[v$ に隣接する点が $N_{i-1}$ にある$]\}$ を $N_{i-1}$ の隣接点集合とよんでいます。