Graph など

Ryo Kawai

2022 年 1 月 18 日

前書き (memo)

TeX の環境をいじって色々試しているため, とても奇妙な PDF になってしまっている. 環境としては, エディタとして VScode(LaTeX Workshop,LaTeX Utilities), TeX として TeXLive2021 を使用している.

基本的に, 呼び名が複数あるものは最初に表記してある方で統一する.

. . . .

往々にしてルー大柴さんのようになるが許して欲しい.

macro:-¿“protect “abs

目次

1 Basis

表記 1.0.1

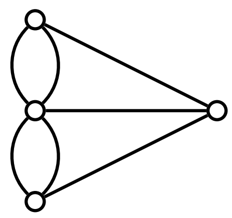
非負整数 $\{0, 1, \ldots\}$ を $\mathbb{N}$, 正の整数 $\mathbb{N} \setminus \{0\} = \{1, 2, \ldots\}$ を $\mathbb{N}^+$ と表す.

2 Graph のモチベーションと定義

2.1 モチベーション

いつか描きたい. いくつか例をあげる.

例 2.1.1 | ⟨ The problem of the bridges of Königsberg(ケーニヒスベルグの橋の問題)⟩



有名な一筆書きに関する問題

2.2 定義

数学的に扱いやすいように, グラフという言葉をきちんと定義していきたい. しかし単にグラフといっても実は様々な区別ができ, それに応じて多くの定義がある. ここではまず, そのようなさまざまなグラフの定義をしておく.

まず初めにこの PDF で通常使用するグラフの定義をしておく.

定義 2.2.1 | ⟨ graph(グラフ)⟩

graph(グラフ) , または **simple(単純)** グラフとは, **vertex(頂点)** [*1] と呼ばれる object の集合 V と, V の異なる 2 元からなる部分集合である **edge(辺)** [*2] の集合 E の組 $G = (V, E)$ のことである. すなわち $V, E : set$ が

$$E \subseteq \binom{V}{2}^{*3}$$

を満たすとき, $G = (V, E)$ は graph であるという.

[*1] **node(ノード)** , **point(点)** ともいう.
[*2] **line(線)** , **link(リンク)** ともいう.
[*3] $\binom{A}{k} = \{X \subset A \mid |X| = k\}$

表記上の曖昧さを回避するために $V \cap E = \emptyset$ とする[*4].

定義 2.2.2

vertex の集合を **vertex set(頂点集合)** , edge の集合を **edge set(辺集合)** とよぶ. すなわち, 定義 2.2.1 の V, E はそれぞれ頂点集合, 辺集合である. 頂点集合 V を持つグラフを, **graph on V(V 上のグラフ)** という.

表記 2.2.3

グラフ G の頂点集合を $\boldsymbol{V(G)}$, 辺集合を $\boldsymbol{E(G)}$ で表す. 厳密に区別せず, 意味が明白な場合は $v \in V(G)$ を $v \in G$ と書いたり, $e \in E(G)$ を $e \in G$ と書いたりする.

定義 2.2.4

グラフ G の頂点の数 (頂点集合の濃度) を **order(位数)** といい, $\boldsymbol{|G|}$ で表す. 辺の数 (辺集合の濃度) を **size(サイズ)** といい, $\boldsymbol{\|G\|}$ で表す.

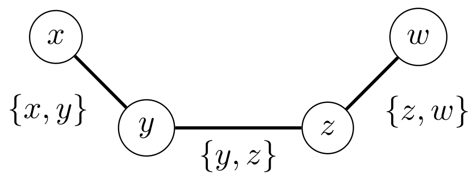
位数とサイズが有限なグラフを **finite graph(有限グラフ)** , そうでないグラフを **infinite graph(無限グラフ)** という. グラフの定義から, 位数が有限ならサイズも有限であるが, この後出てくる多重辺を持つグラフにも対応できるようにこのように定義する.

位数が 0 であるグラフ $(\emptyset, \emptyset)$ を **empty graph(空グラフ)** といい, $\boldsymbol{\emptyset}$ で表す. 位数が 1 であるグラフ $(\{*\}, \emptyset)$ を **trivial(自明)** なグラフという. また, サイズが 0 であるグラフ $(V, \emptyset)$ を **edgeless graph** という[*5].

今後は基本的に有限グラフを扱うことにして, 無限グラフについては § 13 に任せることにする.

通常は, グラフを絵で表すときには頂点を点で, 辺を頂点同士を結ぶ線で表す (絵で表されたグラフを考えるために数学の用語で整備した感じもするが). この時に辺の形や点の位置は重要ではなく, どの頂点が結ばれているかが重要である.

例 2.2.5



上のグラフは $V = \{x, y, z, w\}$ 上のグラフで, 辺集合は $E = \{\{x, y\}, \{y, w\}, \{w, z\}\}$ である.

通常のグラフ理論で考えられるグラフは前述のようなモデルだが, 例えば例 2.1.1 のように, ある 2

[*4] 感覚的には頂点の集合と辺の集合は共通部分を持たないで欲しいので, 明記しておく. 例えば公理的集合論では $2 = \{0, 1\}$ であるから $V = \{0, 1, 2\}, E = \{\{0, 1\}\}$ とすると $V \cap E \neq \emptyset$ となってしまうような事態が発生する.

[*5] empty graph を null graph, edgeless graph を empty graph とする流儀もある.

つの頂点を結ぶ辺が 2 本以上ある場合 (この辺のことを **multiple edges(多重辺)** という) や, 同じ頂点を結んでいる辺がある場合 (この辺のことを **loop(ループ)** という) は, 前述のグラフの定義では扱うことができない. このような多重辺やループを持つグラフを **multigraph(マルチグラフ)** という. グラフとマルチグラフの混同を防ぐために, グラフを単純グラフと言うことが多い.

マルチグラフの定義をしておこう.

定義 2.2.6 | 〈 multigraph(マルチグラフ)〉

multigraph(マルチグラフ) とは, 2 つの非交な集合 V, E(これらの要素をそれぞれ vertex,edge と呼ぶ) と, 辺に対して接続している頂点を対応させる写像 φ の組 $G = (V, E, \varphi)$ のことである. すなわち $V, E : set, \varphi : map$ が

$$V \cap E = \emptyset \ \wedge \ \varphi : E \to V \cup \binom{V}{2}$$

を満たすとき, $G = (V, E, \varphi)$ は multigraph であるという.

このように定義すると, ちゃんと多重辺やループを扱うことができる. 定義 2.2.1 では辺そのものが接続している頂点の情報を持っていたが, 定義 2.2.6 では φ が辺の接続している頂点の情報を持っているからである. ループは φ によって V に行くことに注意する.

マルチグラフについては § 9 で詳しく触れる.

また, グラフの辺に向きをつけたモデルを考えたいときもある. そのような向きのついたグラフのことを **directed graph(有向グラフ)** といい, 省略して **digraph** とよくいう. この PDF でも digraph で統一する.

有向グラフの定義をしておこう.

定義 2.2.7 | 〈 digraph(有向グラフ)〉

digraph(有向グラフ) とは, **vertex(頂点)** と呼ばれる object の集合 V と, V の異なる 2 元の組である **arc(有向辺)** の集合 E の組 $D = (V, E)$ のことである. すなわち $V, E : set$ が

$$E \subseteq V^2 \backslash \{(x, x) \mid x \in V\}$$

を満たすとき, $D = (V, E)$ は digraph であるという.

表記上の曖昧さを回避するために $V \cap E = \emptyset$ とする.

定義 2.2.8

arc の集合を **arc set(有向辺集合)** という.

だいたいは $E \neq \emptyset$ の場合を考える. 図で表す際には, $(u, v) \in E$ を u から v への矢印で表すことが多い. 定義より, 辺 (u, v) と (v, u) は区別され, これらは向きを持っていると考えることが出来る. また, (u, v) と (v, u) のように向きが逆な辺であれば、2 頂点間に辺が 2 本接続することが可能である.

この定義は [8] にのっとったものだが, [10] などではグラフに向きをつけたものとして定義してある. すなわち $\{u, v\} \in E \ \Rightarrow \ \{v, u\} \notin E$ となっている. ここでは必要であればこの条件をつけること

で対応する.

有向グラフについては § 10 で詳しく触れる.

当然向きのついたマルチグラフも考えれるわけで, これを **quiver(箙)** という. 本によってさまざまであり, digraph を quiver と呼んでいたり, digraph を多重辺やループを含めて定義していたりもするが, この PDF ではこの呼び名で統一する[*6].

箙の定義をしておこう.

定義 2.2.9 | ⟨ quiver(箙)⟩

quiver(箙) とは, 2 つの非交な集合 V, E(これらの要素をそれぞれ vertex,arc と呼ぶ) と, 有向辺に対して接続している頂点を対応させる写像 φ_1, φ_2 の組 $G = (V, E, \varphi_1, \varphi_2)$ のことである. すなわち $V, E : set, \varphi_1, \varphi_2 : map$ が

$$V \cap E = \emptyset \ \wedge \ \varphi_1, \varphi_2 : E \to V$$

を満たすとき, $G = (V, E, \varphi_1, \varphi_2)$ は quiver であるという.

ただし, $v_1, v_2 \in V,\ \varphi_1(v_1) = \varphi_2(v_1),\ \varphi_1(v_2) = \varphi_2(v_2)$ のとき $\varphi_1(v_1) = \varphi_1(v_2) \ \Rightarrow \ v_1 = v_2$ と定める.

このように定義すると, ちゃんと多重辺やループを扱うことができる. 定義 2.2.7 では有向辺そのものが接続している頂点の情報を持っていたが, 定義 2.2.9 では φ_1, φ_2 が辺の接続している頂点の情報を持っているからである. 最後の条件は, 同じ頂点でのループが一つに定まることを意味している. まだあまり箙を勉強していないのでわからないが, これはいらないかもしれない.

箙については § 12 で詳しく触れる.

[*6] 物理や環論ではどうやら有向なマルチグラフを quiver と呼ぶことが多いらしいので, こうした.

3 graph

3.1 basis

定義 3.1.1 | ⟨ incident⟩

グラフにおいて, $v \in e$ であるとき, 頂点 v は辺 e に **incident(接続)** しているといい, e を v の **incident edge(接続辺)** という. 一つの辺に接続する 2 つの頂点をその辺の **end(端点)** といい, 辺はその端点を **join(結ぶ)** という.

辺 $\{x, y\}$ をよく省略して $xy(= yx)$ と表す. $x \in X \ \wedge \ y \in Y(X, Y \subseteq V)$ であるとき, 辺 xy を $\boldsymbol{X}$**-**$\boldsymbol{Y}$ **edge(**$\boldsymbol{X}$**-**$\boldsymbol{Y}$ **辺)** という. E に属する X-Y 辺全体の集合を $\boldsymbol{E(X, Y)}$ と表し, $E(\{x\}, Y)$ や $E(X, \{y\})$ のことを単に $E(x, Y)$ や $E(X, y)$ と表す. また, $v \in V$ の E 上の接続辺全体を $\boldsymbol{E(v)}$ と表す. すなわち $E(v) = E(V, v)$ である.

定義 3.1.2 | ⟨ adjacent⟩

2 つの頂点 x, y が $\{x, y\} \in G$ であるとき, x と y は **adjacent(隣接)** しているといい, 互いに他の **neighbour(隣接点)** [*7] であるという. neighbour 関係については定義 3.2.1 にある. また, 2 つの異なる辺 e, f が 1 つの端点を共有しているとき, すなわち $^{\exists}v \in G \ s.t. \ v \in e \ \wedge \ v \in f$ であるときも e と f は **adjacent(隣接)** しているという. [*8]

定義 3.1.3 | ⟨ complete graph⟩

全ての頂点が隣接しているグラフを **complete graph(完全グラフ)** といい, $|G| = n$ のものを $\boldsymbol{K_n}$ で表す[*9]. 特に K_3 は **triangle(三角形)** と呼ばれる.

定義 3.1.4 | ⟨ independent⟩

グラフ G の頂点で, 他のどの頂点とも隣接していない頂点を **independent(独立)** した頂点という. 同じように, グラフ G の辺で, 他のどの辺とも隣接していない辺を **independent(独立)** した辺という. より一般に, $X \subseteq V(G) \veebar E(G)$ のすべての要素が独立しているときに X は **independent(独立)** しているという. $V(G)$ が独立しているとき, **stable set(安定集合)** ということもある.

定義 3.1.5 | ⟨ graph isomorphism⟩

2 つのグラフ $G = (V, E), G' = (V', E')$ に対して, グラフの間の写像 $\varphi : V \to V'$ を $\boldsymbol{\varphi : G \to G'}$ と表す. 2 つのグラフ $G = (V, E), G' = (V', E')$ に対して, $\varphi : G \to G'$ が

$$\{x, y\} \in E \ \Rightarrow \ \{\varphi(x), \varphi(y) \in E'\}$$

を満たすとき, φ を **graph homomorphism(グラフ準同型写像)** であるという. 特にこのとき,

---

*7 イギリス英語は neighbour,neighbourhood. アメリカ英語は neighbor,neighborhood.

*8 x と y を **adjacent vertices(隣接頂点)** , e と f を **adjacent edges(隣接辺)** という.

*9 K^n と表記している本もある.

$x' \in V'$ の φ による逆像 $\varphi^{-1}(x')$ は独立している.

φ が全単射であり φ^{-1} もグラフ準同型写像であるとき, φ を **graph isomorphism(グラフ同型写像)** という. またこのとき, G と G' は **graph isomorphic(グラフ同型)** であるといい, $\boldsymbol{G \simeq G'}$ と書き表す. 同型なグラフは区別せず, $G \simeq G'$ のことを, $G = G'$ と書くことが多い.

G から G へのグラフ同型写像を **automorphism(自己同型写像)** という.

定義 3.1.6

同型写像の下で保存されるような性質を **graph property(グラフの性質)** といい, その中で引数を持つものを **graph invariant(グラフ不変量)** という.

グラフの頂点の数や辺の数などはグラフ不変量である.

3.2 degree

定義 3.2.1 | ⟨ neighbours⟩

グラフ G の頂点集合 U に対して, その頂点の隣接点で U に属さないもの全体, すなわち,

$$\left\{x \in V(G \setminus U) \,\middle|\, {}^{\exists}y \in U \; s.t. \; xy \in E(G)\right\}$$

をグラフ G における U の **neighbourhood(近傍)** [*10] または **open neighbourhood(開近傍)** といい, $\boldsymbol{N_G(U)}$ で表す. $N_G(U) \cup U$ をグラフ G における U の **closed neighbourhood(閉近傍)** といい, $\boldsymbol{N_G[U]}$ で表す.

グラフ G が明らかである場合には $N_G(U)$ を単に $N(U)$ と書き表し, 特に $U = \{u\}$ のとき, $N(\{u\})$ を単に $N(u)$ と表す. すなわち $N(u)$ は u の隣接点全体である. $N_G[U]$ も同様である.

定義 3.2.2 | ⟨ degree⟩

グラフ G の頂点 v に対して, その頂点の接続辺の数, すなわち,

$$|E(v)| = \left|\left\{vx \in E(G) \,\middle|\, {}^{\exists}x \in V(G)\right\}\right|$$

をグラフ G における x の **degree(次数)** といい, $\boldsymbol{\deg_G(v)}$ [*11] や, G が明らかである場合は単に $\deg(v)$ で表す. (単純) グラフの場合, 頂点 v に対して, 接続する辺の数と隣接する頂点の数は等しいため, $\deg(v) = |N(v)|$ が成り立つ.

次数が 0 の頂点を **isolated vertex(孤立点)** , 次数が 1 の頂点を **leaf(葉)** または **end vertex(端点)** という. 葉に接続する辺を **pendant edge** という.

$\sup\{\deg(v) \mid v \in V(G)\}$ を G の **maximum degree(最大次数)** , $\inf\{\deg(v) \mid v \in V(G)\}$ を G の **minimum degree(最小次数)** といい, それぞれ $\boldsymbol{\Delta(G)}$,$\boldsymbol{\delta(G)}$ で表す.

定義より次のことが直ちにわかる.

[*10] neighbours of U ともいう.

[*11] $d_G(v)$ で表している本もある. [10] とか.

系 3.2.3

$|G| = n,\ {}^{\forall}v \in V(G)$ のとき, 以下が成り立つ.

$$0 \leq \delta(G) \leq \deg_G(v) \leq \Delta(G) \leq n-1$$

定理 3.2.4 | ⟨ The First Theorem of Graph Theory(グラフ理論の第一定理)⟩

サイズ m のグラフ G に対して, 以下が成り立つ.

$$\sum_{v \in V(G)} \deg(v) = 2m$$

Proof. 1 つの辺に対して 2 つの頂点が接続していることからわかる. □

系 3.2.5

任意のグラフにおいて, 次数が奇数の頂点の個数は偶数である.

Proof. 定理 3.2.4 よりわかる. □

3.3 operation

定義 3.3.1 | ⟨ subgraph⟩

2 つのグラフ $G = (V, E), G' = (V', E')$ が, $V' \subseteq V \ \wedge \ E' \subseteq E$ であるとき, G' を G の **subgraph(部分グラフ)** であるといい, G は G' の **supergraph(スーパーグラフ)** であるという. G' が G の部分グラフであるとき, $\boldsymbol{G \subseteq G'}$ と書き表す.

よく G は G' を **contain(含む)** ともいう.

$G' \subseteq G \wedge G' \neq G$ であるとき, すなわち $G' \subsetneq G$ であるとき, G' を G の **proper subgraph(真部分グラフ)** という. また, $G' \subseteq G \wedge V(G') = V(G)$ であるとき, G' を G の **spanning subgraph(全域部分グラフ)** という.

定義 3.3.2 | ⟨ induced⟩

グラフ G の空でない頂点部分集合 $S \subset V(G)$ に対して, 頂点集合が S であり, S の 2 点 v, u が G 上で隣接しているときにのみ S 上で隣接しているグラフを **subgraph of $\boldsymbol{G}$ induced by $\boldsymbol{S}$($\boldsymbol{S}$ によって誘導される $\boldsymbol{G}$ の部分グラフ)** といい, $\boldsymbol{G[S]}$ と表す. すなわち,

$$G[S] \coloneqq (S, \{xy \mid x, y \in S \ \wedge \ xy \in E(G)\})$$

である. G の部分グラフ H が $H = G[S]$ となる空でない頂点集合 S をもつとき, H を G の **induced subgraph(誘導部分グラフ)** という.

同じように, グラフ G の空でない辺部分集合 $X \subset E(G)$ に対して, 辺集合が X であり, G 上で X の元 (辺) に接続している頂点全体を頂点集合としてもつグラフを **subgraph induced by $\boldsymbol{X}$($\boldsymbol{X}$ に**

よって誘導される部分グラフ) といい, $\boldsymbol{G[X]}$ と表す. すなわち,

$$G[X] \coloneqq (\{v \mid {}^{\exists}e \in E(G)\ s.t.\ v \in e\}, X)$$

である. G の部分グラフ H が $H = G[X]$ となる空でない辺集合 X をもつとき, H を G の **edge induced subgraph(辺誘導部分グラフ)** という.

任意のグラフ G において $G = G[V(G)]$ であり, 孤立点を持たない任意のグラフ H において $H = H[E(H)]$ である.

定義 3.3.3 | ⟨ complement⟩

グラフ $G = (V, E)$ に対して, すべての頂点の隣接関係を反転させたもの, すなわち V の 2 頂点 v, u に対して $vu \in E \Leftrightarrow vu \notin E'$ を満たすグラフ (V, E') をグラフ G の **complement graph(補グラフ)** といい, $\boldsymbol{\overline{G}}$ で表す. 定義より明らかに $|G| = n, \|G\| = m$ ならば $\left|\overline{G}\right| = n, \left\|\overline{G}\right\| = \binom{n}{2} - m$ であり, $G \simeq H \Leftrightarrow \overline{G} \simeq \overline{H}$ である. $\overline{K_n}$ は位数 n の edgeless graph である.

G が $G \simeq \overline{G}$ であるとき, G は **self complementary** であるという. 明らかに self complementary なグラフは位数 n が $n \equiv 0 \pmod 4 \ \vee\ n \equiv 1 \pmod 4$ である.

表記 3.3.4

グラフ $G = (V, E)$ とその頂点 $v \in V$ と辺 $e \in E$ に対して, G から頂点 v とその接続辺を除いたグラフを $\boldsymbol{G - v}$, G から辺 e を除いたグラフを $\boldsymbol{G - e}$ で表す. より一般に, G の真の頂点部分集合 $U \subsetneq V$ と辺部分集合 $X \subset E$ に対して, G から U に含まれる頂点とその接続辺を全て除いたグラフを $\boldsymbol{G - U}$, G から X に含まれる辺を全て除いたグラフを $\boldsymbol{G - X}$ で表す. また, G の隣接していない 2 頂点 u, v を結ぶ辺を加えたグラフを $\boldsymbol{G + uv}$ で表す. すなわち,

$$\begin{aligned} G - v &\coloneqq G[V \setminus \{v\}] & \qquad G - e &\coloneqq (V, E \setminus \{e\}) \\ G - U &\coloneqq G[V \setminus U] & G - X &\coloneqq (V, E \setminus X) \\ & & G + uv &\coloneqq (V, E \cup \{uv\}) \end{aligned}$$

である.

辺集合を取り除く時は頂点は取り除かないため, $G - X$ は一般には $G[E \setminus X]$ と同じにはならない. 例えば G として leaf v があるグラフ, X として pendant edge を取ると, $v \in G - X$ だが $v \notin G[E(G) \setminus X]$ である.

U や X を G の中に限らなくてもよさそうだが, 結局差集合の定義より含まれていない部分は無視される. X については限る必要がなく, U は $V \setminus U \neq \emptyset$ であれば限る必要がない. 頂点集合のところに「真の」とついているのは, $G[\emptyset]$ が定義されていないからである.

定義 3.3.5

2 つのグラフ G_1, G_2 に対して, 2 つのグラフの **union(和)** を, 頂点集合と辺集合を集合として union

したもので定め, $\boldsymbol{G_1 \cup G_2}$ で表す. **intersection(共通部分)** も同様で, $\boldsymbol{G_1 \cap G_2}$ で表す. すなわち,

$$G_1 \cup G_2 \coloneqq (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$$
$$G_1 \cap G_2 \coloneqq (V(G_1) \cap V(G_2), E(G_1) \cap E(G_2))$$

である. これらは確かめるとグラフになっている.

$G_1 \cap G_2 = \emptyset$ であるとき, G_1 と G_2 は **disjoint(非交)** であるという. G_1, G_2 が disjoint であるとき, $G_1 \cup G_2$ を $\boldsymbol{G_1 + G_2}$ で表す. 逆に, $G_1 + G_2$ と書いてあるときは G_1 と G_2 は disjoint であることを仮定するものとする. グラフ G に対して, $G+G$ と書いたときは, G と disjoint な G のコピーとの union であると考える. すなわち, G と同型で disjoint なグラフ G' と G との union $G+G'$ のことを $\boldsymbol{G+G}$ であるとし, 単に $\boldsymbol{2G}$ で表す. より一般に, G がグラフ H と同型で互いに disjoint なグラフ $H_1, H_2, \ldots, H_k$ $(k \geq 2)$ によって $G = H_1 + H_2 + \cdots + H_k$ とかけるとき, $G = \boldsymbol{kH}$ と表す.

G_1, G_2 が disjoint であるとき, その 2 つのグラフの頂点間を結んで得られるグラフを G_1 と G_2 の **join(結合)** といい, $\boldsymbol{G_1 \vee G_2}$ で表す. すなわち,

$$G_1 \vee G_2 \coloneqq (V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{uv \mid u \in V(G_1), v \in V(G_2)\})$$

である.

定義 3.3.6

2 つのグラフ G_1, G_2 に対して, その直積ともいえるグラフ G を次のように考えることができる. G の頂点集合を, G_1, G_2 の頂点集合の集合としての直積で定める. G の 2 つの頂点 $(u_1, u_2), (v_1, v_2)$ の隣接関係を, $u_1 = v_1 \ \wedge \ u_2 v_2 \in E(G_2)$ または $u_2 = v_2 \ \wedge \ u_1 v_1 \in E(G_1)$ であるとき, そのときに限り隣接していると定める. すなわち,

$$V(G) \coloneqq V(G_1) \times V(G_2)$$
$$E(G) \coloneqq \{\{(u_1, u_2), (v_1, v_2)\} \mid (u_1 = v_1 \ \wedge \ u_2 v_2 \in E(G_2)) \ \vee \ (u_2 = v_2 \ \wedge \ u_1 v_1 \in E(G_1))\}$$

であるとき, G を G_1 と G_2 の **cartesian product(直積, デカルト積)** といい, $\boldsymbol{G_1 \times G_2}$ で表す.

定義 3.3.7 | ⟨ contraction⟩

グラフ $G = (V, E)$ とその辺 $xy \in E$ に対して, $v_{xy} \notin V$ として

$$(V \backslash \{x,y\} \cup \{v_{xy}\}, \{vw \in E \mid \{x,y\} \cap \{v,w\} = \emptyset\} \cup \{v_{xy} w \mid w \in V \setminus \{x,y\} \ s.t.\ xw \in E \ \vee \ yw \in E\})$$

すなわち

$$G - \{x,y\} \cup \{v_{xy}\} + \{v_{xy} w \mid w \in V \setminus \{x,y\} \ s.t.\ xw \in E \ \vee \ yw \in E\}$$

で与えられるグラフを $\boldsymbol{G/xy}$ で表し, G から G/xy を作ることを **contraction(縮約)** という.

定義より次のことが直ちにわかる.

系 3.3.8

定義 3.3.7 の $G, v_{xy}, G/xy$ において,

$$N(\{x,y\}) = N(v_{xy}) \ , \ G - \{x,y\} = G/xy - v_{xy}$$

3.4 regular

定義 3.4.1

グラフ G の各頂点の次数が同じであるとき, G は **regular(正則)** であるといい, 特に各頂点の次数が r であるときに ***r*-regular(*r*-正則)** であるという. 特に 3-正則なグラフは **cubic graph** と呼ばれる.

定理 3.4.2

$n, r \in \mathbb{N}^+$ とする. 位数 n の r-正則グラフが存在する必要十分条件は, $0 \leq r \leq n-1 \land \neg(n, r$: 奇数$)$ である.

Proof. wait □

定義 3.4.3 | ⟨ factor⟩

グラフ G の全域部分グラフを G の **factor(因子)** といい, 特に k-正則なものを ***k*-factor(*k*-因子)** という.

定理 3.4.4 | ⟨ Petersen's Theorem⟩

橋を持たない cubic graph は 1-factor をもつ.

Proof. G を橋を持たない cubic graph, S を $V(G)$ の真の部分集合とし $k = |S|$ とする.

□

3.5 path and cycle

定義 3.5.1 | ⟨ path⟩

空でないグラフ $P = (V, E)$ が

$$V = \{x_0, x_1, \ldots, x_k\}, E = \{x_0x_1, x_1x_2, \ldots, x_{k-1}x_k\} \ (x_0, x_1, \ldots, x_k \text{はすべて異なる})$$

とかけるとき, P を **path(道)** といい, $\|P\| = k-1$ を path P の **length(長さ)** という. 位数 n の path を $\boldsymbol{P_n}$ で表す. すなわち P_n の長さは $n-1$ である. またこの P について, x_0 と x_k は P で **link(結ばれている)** という. x_0 と x_k を P の **end(端点)** といい, $x_1, \ldots, x_{k-1}$ を P の **inner vertex(内点)** という. 複数の道が互いに内点を含まないとき, それらを **independent(独立)** な path といい, それぞれの path は **independent(独立)** であるという.

表記 3.5.2

定義 3.5.1 の P を簡単に $P = x_0x_1 \cdots x_k$ と書いて, x_0 から $_k$ までの path という. また、$0 \leq i \leq$

$j \le k$ に対して,

$$Px_i \coloneqq x_0 \cdots x_i$$
$$x_iP \coloneqq x_i \cdots x_k$$
$$x_iPx_j \coloneqq x_i \cdots x_j$$

のように書き表す. 他にも, 直観的にわかりやすいため, path $P(\ni x), Q(\ni x, y), R(\ni y)$ に対して $Px \cup xQy \cup yR$ を $PxQyR$ と書き表す.

定義 3.5.3 | ⟨ *A*-*B* path⟩

頂点集合 A, B に対して, path $P = x_0x_1 \cdots x_k$ が

$$V(P) \cap A = \{x_0\} \ \wedge \ V(P) \cap B = \{x_k\}$$

であるとき, P を ***A*-*B* path(*A*-*B* 道)** という.

表記 3.5.4

上の $A = \{a\}$ のときは, $\{a\}$-B path の意味で単に a-B path と書く. また, A, B がグラフであるとき, $V(A)$-$V(B)$ path を単に A-B path と書く.

定義 3.5.5 | ⟨ *H*-path⟩

グラフ H に対して, その端点のみで H と接しているような自明でない path のことを **H-path(*H*-道)** という. すなわち, path $P = x_0x_1 \cdots x_k$ が

$$P \cap H = (\{x_0, x_k\}, \emptyset) \ \wedge \ |P| > 1$$

であるとき, P は H-path であるという.

定義より, 長さが 1 の H-path x_0x_1 の辺は H の辺にはならない.

定義 3.5.6 | ⟨ cycle⟩

空でないグラフ $C = (V, E)$ が

$$V = \{x_0, x_1, \ldots, x_{k-1}\}, E = \{x_0x_1, x_1x_2, \ldots, x_{k-2}x_{k-1}, x_{k-1}x_0\} \ (x_0, x_1, \ldots, x_k \text{はすべて異なる}, k \ge 3)$$

とかけるとき, C を **cycle(閉路)** といい, 位数 n のものを $\boldsymbol{C_n}$ で表す.

いいかえれば, path $P = x_0x_1 \cdots x_{k-1} (k \ge 3)$ に対して $C_k \coloneqq P + x_0x_{k-1}$ を cycle という.

3.6 connectivity

この subsection にはまずい部分が多い

定義 3.6.1 | ⟨ connected⟩

グラフ G が **connected(連結)** であるとは, G の任意の 2 頂点 x, y に対してその 2 点を結ぶ G 上の path が存在することである. すなわち,

$$^{\forall}x, y \in G, ^{\exists}P \subset G \ : \ \text{path } s.t. \ P = x \cdots y$$

であるとき, G は connected であるという.

また, グラフ G が連結でないとき, グラフ G は **disconnected(非連結)** であるという.

定義 3.6.2

グラフ G の空でない極大な連結部分グラフを G の **component(連結成分)** という. 言い換えれば, H が G の連結成分であるとは, G の連結な部分グラフで H を真部分グラフとして持つグラフが存在しないということである. 各連結成分は共通部分を持たない. そのため, $G_1, G_2, \ldots, G_n$ がグラフ G の連結成分であるとき, $G = G_1 + G_2 + \cdots + G_n$ である. またこのときグラフ連結成分の個数 n を $\boldsymbol{k(G)}$ で表す. すなわちグラフ G が連結であるということは $k(G) = 1$ と同値である. 空グラフは連結成分を持たないことに注意する.

次の定理は, 次数がとても高いグラフは連結であるということを示している.

定理 3.6.3

G を位数 n の自明でないグラフとする. G の任意の隣接していない頂点 u, v が $\deg(u) + \deg(v) \geq n - 1$ を満たすとき, G は連結である.

Proof. G の異なる 2 頂点 x, y に対して, その 2 頂点を結ぶ path があることを示す. x と y が隣接しているときは明らかにその接続辺が path となるため, x と y は隣接していないとする. 仮定より, $\deg(x) + \deg(y) \geq n - 1$ であるが, $N(x) \cap N(y) = \emptyset$ とすると x と y が隣接していないことから, $|G| \geq n + 1$ となり矛盾する. よって $N(x) \cap N(y) \neq \emptyset$ であり, $a \in N(x) \cap N(y)$ とすると xay は x-y path になる. よって任意の 2 頂点を結ぶ path が存在するので G は連結である. □

系 3.6.4

位数 n のグラフ G が $\delta(G) \geq (n-1)/2$ であるとき, G は連結である.

Proof. G の任意の隣接していない 2 頂点 u, v について,

$$\deg(u) + \deg(v) \geq \frac{n-1}{2} + \frac{n-1}{2} = n - 1$$

であるため, 定理 3.6.3 より G は連結なグラフである. □

定義 3.6.5 | ⟨ $\boldsymbol{k}$-connected⟩

$k \in \mathbb{N}, |G| > k$ で, $|X| < k$ である任意の頂点集合 (グラフ)X に対して $G - X$ が連結であるとき, グラフ G は $\boldsymbol{k}$**-connected(**$\boldsymbol{k}$**-連結)** または $\boldsymbol{k}$**-vertex connected(**$\boldsymbol{k}$**-点連結)** であるという.

X は任意だが, 仮に G の部分グラフではない X を取ったとしても $|G| > k, |X| < k$ より $G \cap X \subset X' \subset G, |X'| < k$ となる X' が取れ, $G - X \supset G - X'$ である. そのため, $X \subset G$ としても問題はない. 定義より, 全ての空でないグラフは 0-連結であり, 全ての連結なグラフは 1-連結である. また定義より $n, m \in \mathbb{N}, n < m$ のとき, グラフ G が m-連結ならば G は n-連結である.

定義 3.6.6

定義 3.6.5 より, グラフ G は有限なので $\{x \in \mathbb{N} \mid G \text{ は } x\text{-連結}\}$ は最大値をもつ. その値をグラフ G の **connectivity(連結度)** または **vertex connectivity(点連結度)** といい, $\boldsymbol{\kappa(G)}$ で表す. すなわち, グラフ G に対して k-連結であるが $(k+1)$-連結でない $k \in \mathbb{N}$ が存在し, $\kappa(G) = k$ を G の connectivity という.

当然点の場合があれば辺の場合もある.

定義 3.6.7 | ⟨ *k*-edge connected⟩

$k \in \mathbb{N}, |G| > k$ で, $|X| < k$ である任意の辺集合 X に対して $G - X$ が連結であるとき, グラフ G は ***k*-edge connected(*k*-辺連結)** であるという.

点連結のときと同様の議論で, $X \subset G$ としても問題はない. 定義より, 全ての空でないグラフは 0-辺連結であり, 全ての連結なグラフは 1-辺連結である. また定義より $n, m \in \mathbb{N}, n < m$ のとき, グラフ G が m-辺連結ならば G は n-辺連結である.

定義 3.6.8

定義 3.6.7 より, グラフ G は有限なので $\{x \in \mathbb{N} : G \text{ は } x\text{-辺連結}\}$ は最大値をもつ. その値をグラフ G の **edge connectivity(辺連結度)** といい, $\boldsymbol{\lambda(G)}$ で表す. すなわち, グラフ G に対して k-連結であるが $(k+1)$-辺連結でない $k \in \mathbb{N}$ が存在し, $\lambda(G) = k$ を G の edge connectivity という.

定義より次のことが直ちにわかる.

系 3.6.9

位数 n のグラフ G に対して

$$0 \leq \kappa(G) \leq n-1 \ , \ 0 \leq \lambda(G) \leq \delta(G) \leq n-1$$

また, (点) 連結度と辺連結度の間には綺麗な関係がある.

定理 3.6.10 | ⟨ Whitney's Inequalities⟩

任意のグラフ G に対して, 以下が成り立つ.

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

Proof. $\kappa(G) \leq \lambda(G)$ を示す. wait □

3.7 distance

定義 3.7.1 | ⟨ distance⟩

連結なグラフ G においては, その 2 頂点 u と v を結ぶ path が存在する. それらの path の中で最も短いものの長さを G における u と v の **distance(距離)** といい, $\boldsymbol{d_G(u,v)}$ や, G が明らかである場合は単に $d(u,v)$ で表す. すなわち,

$$d_G(u,v) = \inf\{\|P\| \mid P : u\text{-}v\text{path}\}$$

である. u-v path の中で長さが $d(u,v)$ であるものを ***u-v* geodesic** という. d は明らかに以下を満たす.

1. $u, v \in V(G), d(u,v) \geq 0$
2. $d(u,v) = 0 \Leftrightarrow u = v$
3. $d(u,v) = d(v,u)$ (**symmetric proprety(反対称律)**)
4. $u, v, w \in V(G), d(u,v) \leq d(u,w) + d(w,v)$ (**triangle inequality(三角不等式)**)

よって d は G 上の **metric(距離)** となり, $(V(G), d)$ を **metric space(距離空間)** として考えることができる.

定理 3.7.2

自明でないグラフ G が二部グラフである必要十分条件は, 奇数の長さの閉路を含まないことである.

Proof. wait(2.6) □

定義 3.7.3 | ⟨ eccentricity⟩

連結なグラフ G の頂点 v において, v から一番遠い頂点との距離を v の **eccentricity(離心率)** といい, $\boldsymbol{e(v)}$ で表す. すなわち,

$$e(v) \coloneqq \sup\{d(u,v) \mid u \in V(G)\}$$

である.

定理 3.7.4

自明でない連結なグラフ G の隣接している 2 頂点 u, v において, 以下が成り立つ.

$$|e(u) - e(v)| \leq 1$$

Proof. $e(u) \geq e(v)$ として問題ない. 定義より $e(u) = d(u,w)$ となる $w \in V(G)$ が存在する. すると, 三角不等式より $d(u,w) \leq d(u,v) + d(v,w)$ であり, 仮定より $d(u,v) = 1, d(v,w) \leq e(v)$ であるから,

$$|e(u) - e(v)| = e(u) - e(v) \leq d(u,w) - d(v,w) \leq d(u,v) = 1$$

□

定義 3.7.5 | ⟨ diameter,radius⟩

連結なグラフ G の各頂点の離心率に対して, その最大値と最小値をそれぞれ **diameter(直径)** , **radius(半径)** といい, **diam(G)** , **rad(G)** で表す. すなわち,

$$\mathrm{diam}(G) \coloneqq \sup\{e(v) \mid v \in V(G)\},\ \mathrm{rad}(G) \coloneqq \inf\{e(v) \mid v \in V(G)\}$$

である. 定義より, G の任意の 2 点 u, v に対して $d(u, v)$ の最大値が diameter である. $e(v) = \mathrm{rad}(G)$ となる $v \in V(G)$ を **central vertex(中心点)** , $e(v) = \mathrm{diam}(G)$ となる $v \in V(G)$ を **peripheral vertex(末端)** , $d(u, v) = \mathrm{diam}(G)$ となる $u, v \in V(G)$ を **antipodal vertices(対蹠点)** という. 定義より対蹠点は 2 つ以上あり, $u, v \in V(G)$ が対蹠点であれば末端ではあるが, 逆は成り立つとは限らない. また, 直径が半径の 2 倍になるとも限らない.

定理 3.7.6

自明でない連結なグラフ G において, 以下が成り立つ.

$$\mathrm{rad}(G) \leq \mathrm{diam}(G) \leq 2\,\mathrm{rad}(G)$$

Proof. $\mathrm{rad}(G) \leq \mathrm{diam}(G)$ は定義より明らかに従う. また, G の任意の対蹠点 u, w, 任意の中心点 v をとると, 定義より $d(u, w) = \mathrm{diam}(G)$, $e(v) = \mathrm{rad}(G)$ である. よって三角不等式より,

$$\mathrm{diam}(G) = d(u, w) \leq d(u, v) + d(v, W) \leq 2e(v) = 2\,\mathrm{rad}(G)$$

であるので $\mathrm{rad}(G) \leq \mathrm{diam}(G) \leq 2\,\mathrm{rad}(G)$ である. □

定義 3.7.7

位数 $n(\geq 2)$ 以上の連結なグラフ G に対して, 各頂点の離心率を小さいものから順に並べた順列を **eccentricity sequence** という. すなわち, G の頂点を適切に選び $v_1, v_2, \ldots, v_n$ としたときに, $e(v_i) = e_i(1 \leq i \leq n)$ かつ $i \leq j \Rightarrow e_i \leq e_j$ となる正の整数の単調増加 (減少しない) 順列 $e_1, e_2, \ldots, e_n$ を eccentricity sequence という. 明らかに $e_1 = \mathrm{rad}(G), e_n = \mathrm{diam}(G)$ であり, 対蹠点が 2 つ以上あることから $e_{n-1} = \mathrm{diam}(G)$ である. 定理 3.7.6 より $e_i \leq 2e_1$ であり, 定理 3.7.4 より $e_{i+1} - e_i \leq 1$ である. また, v_1-v_n path を考えれば, e_1 から e_n の整数を離心率に持つ G の頂点が必ず存在することがわかる.

私たちの通常の感覚に習って言えば, 直径は半径の 2 倍であってほしいが, 一般にそうはならないことが容易にわかる. 例えば直径と半径が一致する例として完全グラフがある. 逆に直径が半径の 2 倍となるようなグラフとして P_{2k+1} がある. 直径が半径の 2 倍となるようなグラフのクラスの決定はできているのだろうか.

定義 3.7.8

連結なグラフ G の中心点の集合によって誘導されるグラフを G の **center(中心)** , 末端の集合に

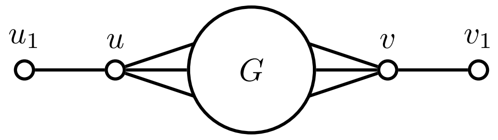
よって誘導されるグラフを G の **periphery(周囲)** といい, それぞれ **Cen($\boldsymbol{G}$)** , **Per($\boldsymbol{G}$)** で表す.

path では $\mathrm{Cen}(P_{2k+1}) = K_1, \mathrm{Cen}(P_{2k}) = K_2, \mathrm{Per}(P_{k+2}) = 2K_1 \ (k \geq 1)$ であり, cycle では $\mathrm{Cen}(C_n) = \mathrm{Per}(C_n) = C_n \ (n \geq 3)$ である. 次の定理は Stephen Hedetniemi が最初に発見したとされている.

定理 3.7.9

任意のグラフに対して, そのグラフを中心に持つようなグラフが存在する.

Proof. G をグラフとする. $H \coloneqq G \vee (u, v, \emptyset) + u_1, v_1 + u_1u + v_1v$ とする. このとき, $e(u_1) = e(v_1) = 4, e(u) = e(v) = 3, e(x) = 2(x \in V(G))$ であるため, $\mathrm{Cen}(H) = H[V(G)] = G$ である. □



3.8 graph example

定義 3.8.1

$n(\in \mathbb{N}^+)$ 個の K_2 の直積であるグラフを **$\boldsymbol{n}$-cube** といい,$\boldsymbol{Q_n}$ で表す. すなわち,

$$Q_n \coloneqq \begin{cases} K_2 & (n = 1) \\ Q_{n-1} \times K_2 & (n \geq 2) \end{cases}$$

である. Q_n は位数 2^n の n-regular グラフである. またこれら Q_n を **hypercubes** ともいう.

K_2 を $V(K_2) = \{0, 1\}$ とすれば, $V(Q_n) = \{(a_1, a_2, \ldots, a_n) \mid a_i \in \{0, 1\} \ (1 \leq i \leq n)\}$ と表すことができ, これを単に $a_1a_2 \cdots a_n$ と表すことにすれば, これは **$\boldsymbol{n}$-bit strings($\boldsymbol{n}$-ビット文字列)** と見ることができる. この表記では, 2 つの頂点が隣接していることと, ビット文字列が 1 つの箇所だけ異なることが対応する.

定義 3.8.2 | ⟨ wheel graph⟩

位数 $n-1(n \geq 4)$ の閉路と 1 つの頂点を結合したグラフを **wheel graph(車輪グラフ)** といい, $\boldsymbol{W_n}$ で表す. すなわち, $W_n \coloneqq C_{n-1} \vee K_1$ である.

定義 3.8.3 | ⟨ gear graph⟩

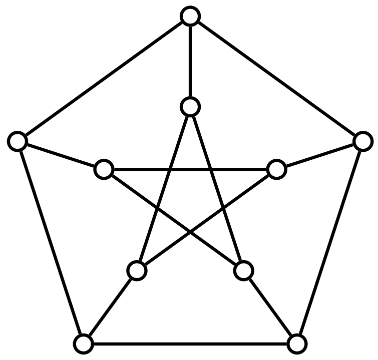
閉路と 1 つの頂点を一つ飛ばしに結合したグラフを **gear graph(ギアグラフ)** という. すなわち,

$$\begin{aligned} V(G) &= \{v\} \cup \{c_1, c_2, \ldots, c_{2n}\} \\ E(G) &= \{vc_{2i} \mid 1 \leq i \leq n\} \cup \{c_ic_{i+1} \mid 1 \leq i \leq 2n-1\} \cup \{c_{2n}c_1\} \end{aligned}$$

であるグラフ G を gear graph という.

定義 3.8.4

以下のグラフを **Petersen Graph(パターソングラフ)** という.



3.9 graph class

定義 3.9.1

A を集合とし, S をその部分集合の集まりとする. このとき, グラフ $(S, \{vw \mid v, w \in S, v \cap w \neq \emptyset\})$ を S による **intersection graph(交点グラフ)** という. 任意のグラフに対して, 同型な交点グラフが存在する.

定義 3.9.2 | ⟨ chordal graph⟩

グラフ G の閉路 $C \subset G$ に対して, $xy \in E(G)$ が $x, y \in V(C) \ \wedge \ xy \notin E(C)$ であるとき, xy を C の **chord(弦)** という. 明らかに triangle は chord を含まない.

グラフ G に含まれる長さ 4 以上の全ての閉路が弦を持つとき, グラフ G を **chordal graph(弦グラフ)** という.

明らかに完全グラフは弦グラフである. 弦グラフは 1970 年代にさまざまな分野で独立に考え出されたグラフのクラスであるため, **triangulated graph** や **rigid circuit graph** と呼ばれていたりもする (呼び名で出身や年代がわかるらしい).

弦グラフには, 前述したような cycle を用いた定義以外にも同値な定義がいくつか知られており, それにより特徴づけられている. 代表的なのは以下である.

1. 任意の relatively minimal cut-set(minimal separating set) がクリークであるグラフ.[7]
2. ある tree T 上の部分木の集まり S に対して, その交点グラフと同型なグラフ.(Gavril,1974)
3. perfect elimination ordering をもつグラフ.

chordal graph では minimal separating set の数は order より小さくなる (一般には O(黄金比^n) ぐらいらしい [6]).

定義 3.9.3 | ⟨ AT-free⟩

グラフ G の異なる 3 頂点に対して, どの 2 頂点に対しても, その 2 頂点を結び他のもう一つの頂点の

近傍を通らないような path が存在するとき, 3 頂点は **asteroidal triple** という. すなわち異なる 3 頂点 $x, y, z \in V(G)$ に対して,

$$^{\forall}u, v \in \{x, y, z\}, w \coloneqq \{x, y, z\} \setminus u, v, ^{\exists} P\text{:path } s.t.\ u, v \in V(P) \ \wedge \ P \cap N(w) = \emptyset$$

であるとき, x, y, z は asteroidal triple であるという. グラフ G が asteroidal triple を持たないとき, すなわち $^{\forall}x, y, z \in V(G)$ が asteroidal triple でないとき, グラフ G は **AT-free** であるという.

定義 3.9.4 | ⟨ interval⟩

$\mathbb{R}$ 上の閉区間の集まり S に対して, その交点グラフと同型なグラフを **interval graph(区間グラフ)** という. 逆に, グラフ G に対して, 交点グラフが G と同型になるような $\mathbb{R}$ 上の閉区間の集まり S が存在するとき, G は **interval** であるという.

グラフ G が interval graph であることと, G が AT-free な弦グラフであることは同値らしい. また, G が C_4-free かつ補グラフが comparability graph であることとも同値らしい. この概念の拡張として k-DIR graph が (多分) ある (その用語でいうと, interval graph は 1-DIR graph である).

note

参考にしたのは [10][8].

4 factorization and decomposition

4.1 factorization

定義 4.1.1 | ⟨ factorable⟩

グラフ G の factor$F_1, F_2, \ldots, F_t$ ($t \in \mathbb{N}^+$) で

$$\bigcup_{i=1}^{t} E(F_i) = E(G) \ \wedge \ E(F_i) \cap E(F_j) = \emptyset \ (i \neq j)$$

を満たすものが存在するとき, グラフ G は $F_1, F_2, \ldots, F_t$ によって **factorable** であるといい, $\mathcal{F} = \{F_1, F_2, \ldots, F_t\}$ を G の **factorization** という. 特に各 factor が k-factor ($k \geq 1$) であるような factorization を ***k*-factorization** といい, それが存在するときに G は ***k*-factorable** であるという. グラフ G が k-factorable であるとき, G は r-正則であり $k|r$ である.

また, 各 factor がグラフ H と同型すなわち $F_i \simeq H$ であるとき, G は ***H*-factorable** であるといい, G は H の **isomorphic factorization** をもつという.

この分野でよく調べられているのは, どのようなグラフが 1-factorable であるである. もちろん定理 3.4.2, 定義 4.1.1 より明らかに偶数次の正則グラフであることが必要条件である. 当然 1-正則グラフは 1-factorable であり, 2-正則グラフ, すなわち cycle も 1-factorable である (交互に辺をとればよい). r-正則 ($r \geq 3$) のときは複雑である. 例えば $r = 3$ のときは, 定理 3.4.4 より橋を持たない cubic graph は 1-factor と 2-factor に分解できるが, Petersen graph などは 1-factorable ではない.

1-factorable については以下の定理がある.

定理 4.1.2

正の整数 k に対して, K_{2k} は 1-factorable である.

Proof. $k = 1$ のときは明らか. $k \geq 2$ のときを考える. 今 K_{2k} の各頂点を $\{v_0, v_1, \ldots, v_{2k-1}\}$ とする. このとき, $v_1, v_2, \ldots, v_{2k-1}$ を $2k-1$ 角形の頂点上に並べ, 真ん中に頂点 v_0 をおいて v_0 と v_i を結びその辺の含む直線と線対称になる頂点を結んでできるグラフを F_i とする. すなわち,

$$F_i = (\{v_0, v_1, \ldots, v_{2k-1}\}, \{v_n v_m \mid n + m \equiv 2k - 1 + 2i \pmod{2k-1}\} + \{v_0 v_i\}) \ (1 \leq i \leq 2k-1)$$

とすると,

$$\bigcup_{i=1}^{2k-1} E(F_i) = E(K_{2k}) \ \wedge \ E(F_i) \cap E(F_j) = \emptyset \ (i \neq j)$$

となる. よって $\{F_1, F_2, \ldots, F_{2k-1}\}$ は K_{2k} の 1-factorization となる. □

上の証明では各 F_i は F_1 を $2\pi(i-1)/(2k-1)$ 回転したものになっている. このような factorization を **cyclic factorization** という.

予想 4.1.3 | ⟨ The 1-Factorization Conjecture⟩

グラフ G が r-正則かつ位数が偶数であり,

$$\begin{aligned} r &\geq n/2 && n \equiv 2 \pmod 4 \\ r &\geq (n-2)/2 && n \equiv 0 \pmod 4 \end{aligned}$$

であるとき, G は 1-factorable である.

とても大きな n 以上の位数のグラフでは, この予想が成り立つことが示された [5].

2-factorable なグラフについては次の定理が知られている.

定理 4.1.4

グラフ G が 2-factorable である必要十分条件は G が $2k$-正則 $(k \in \mathbb{N}^+)$ であることである.

Proof. オイラー周遊を使えば示せるっぽい. wait □

4.2 decomposition

decomposition は factorization をゆるくした概念である.

定義 4.2.1 | ⟨ decomposition⟩

グラフ G の空でない部分グラフの集合 $\{H_1, H_2, \ldots, H_t\}$ $(t \in \mathbb{N}^+)$ で, 各 H_i について $H_i = G[E_i]$ となる辺集合 E_i が存在し,

$$\bigcup_{i=1}^{t} E_i = E(G) \ \wedge \ E_i \cap E_j = \emptyset \ (i \neq j)$$

を満たす $E_1, E_2, \ldots, E_t$ が存在するとき, $\mathcal{D} = \{H_1, H_2, \ldots, H_t\}$ を G の **decomposition** という.

また, 各 H_i がグラフ H と同型すなわち $H_i \simeq H$ であるとき, G は ***H*-decomposition** であるといい, それが存在するときに G は ***H*-decomposable** であるという.

各 H_i が全域部分グラフの時には decomposition は factorization になる.

5 tree

定義 5.0.1 | ⟨ tree⟩

グラフが cycle をもたないとき, そのグラフを **tree(木)** という. 木はよく T と表記する.

6 labeling

定義 6.0.1 | ⟨ labeling⟩

グラフ G の頂点集合もしくは辺集合 (もしくはその両方) に対して, 各要素に値を割り当てる写像のことを **labeling(ラベリング)** という. 通常は値として整数や自然数を割り当てることがおおい.

6.1 graceful labeling

定義 6.1.1 | ⟨ graceful⟩

空でないサイズ m のグラフ G に対して, 各頂点に $0, 1, \ldots, m$ を, 各辺 xy に $|x - y|$ を割り当てるラベリングを考える. このとき, 頂点のラベリングが単射であり, 辺のラベリングの像が $\{1, 2, \ldots, m\}$ であるようなラベリングが存在するとき, グラフ G は **graceful(優美)** であるといい, このラベリングを **graceful labeling(優美ラベリング)** という.

位数 4 までの連結なグラフは優美である. 位数 5 では 3 つの連結なグラフ $C_5, K_5, K_1 \vee 2K_2$ が優美ではない. 位数 6 では以下の 6 つの連結なグラフが優美ではない.

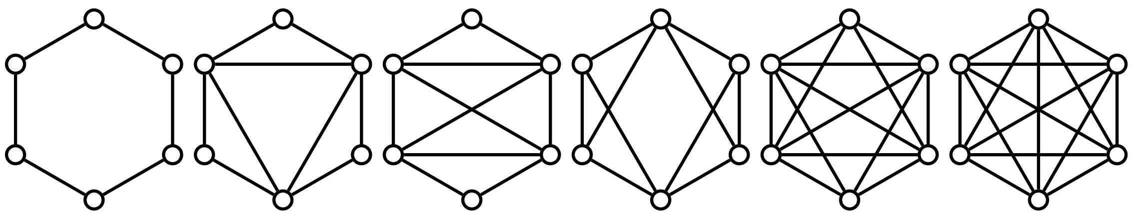


図 1 位数 6 の連結な優美でないグラフ

優美ラベリングに関しては, 以下の有名な予想がある.

予想 6.1.2 | ⟨ Graceful Tree Conjecture⟩

任意の木は優美である.

予想 6.1.3 | ⟨ Ringel's Conjecture⟩

サイズ m の任意の木は K_{2m+1} を分解することができる.

おおきな m では解かれたと [11] にある.

予想 6.1.4 | ⟨ Ringel-Kotzig Conjecture⟩

サイズ m の任意の木は K_{2m+1} を周期的に分解することができる.

実はこの予想は予想 6.1.2 と同値であるらしい. 予想 6.1.2 が解けると予想 6.1.4 が解けるのはすぐにわかる. 反対はちょっとわからない.

6.2 coloring

定義 6.2.1

グラフ G の **vertex coloring** とは, $V(G)$ から **color(色)** と呼ばれる集合 C へのラベリングのことである. その中で, 辺の端点が異なる色に対応している (塗り分けられている) もののことを **vertex proper coloring** という. 基本的に **coloring(彩色)** といった場合には vertex proper coloring をさす. すなわち,

$$c : V(G) \to C \ s.t. \ {}^{\forall}xy \in E(G), f(x) \neq f(y)$$

となる写像のことをグラフ G の coloring という. C としては $\mathbb{N}^+$ やその部分集合をとることがおおく, ここでも基本的に $\mathbb{N}^+$ の部分集合をとる. $|C| = k$ であるとき, c を $\boldsymbol{k}$**-coloring(**$\boldsymbol{k}$**-彩色)** という. c を k-彩色として, $c : V(G) \to \{1, 2, \ldots, k\}$ としたとき, i $(1 \leq i \leq k)$ で彩色されている頂点の集合 (i.e. $c^{-1}(i)$) を V_i を **color class** という. 明らかに $\bigcup_{i=1}^{k} V_i = V(G)$ であり, 各 V_i は partite set となっている.

グラフ G に対して, k-彩色が存在するとき, G は $\boldsymbol{k}$**-colorable(**$\boldsymbol{k}$**-彩色可能)** であるという. また, $\min\{k \mid G{:}k\text{-colorable}\}$ を **chromatic number(彩色数)** といい, $\boldsymbol{\chi(G)}$ で表す. 彩色数が k であるグラフを $\boldsymbol{k}$**-chromatic graph** ともいう. 明らかに $G{:}k\text{-colorable} \Leftrightarrow \chi(G) \leq k \ \wedge \ k \leq |G|$ である.

7 category of graph

7.1 category

定義 7.1.1 | ⟨ category⟩

C が **category(圏)** であるとは, **object(対象)** の集まり **Ob(C)** と, **morphism(射)** の集まり **Mor(C)** の組であって, 以下を満たすものをいう. なお, $\mathrm{Ob}(\mathbf{C}), \mathrm{Mor}(\mathbf{C})$ は集合とは限らないが, a が $\mathbf{C}$ の対象であることを $a \in \mathrm{Ob}(\mathbf{C})$, f が $\mathbf{C}$ の射であることを $f \in \mathrm{Mor}(\mathbf{C})$ と書き表す.

1. 各 $f \in \mathrm{Mor}(\mathbf{C})$ に対して, **domain(ドメイン)** と呼ばれる対象 $\mathrm{dom}(f) \in \mathrm{Ob}(\mathbf{C})$ と **codomain(コドメイン)** と呼ばれる対象 $\mathrm{cod}(f) \in \mathrm{Ob}(\mathbf{C})$ が定められている. $\mathrm{dom}(f) = a, \mathrm{cod}(f) = b$ であることを $f : a \to b$ や $a \xrightarrow{f} b$ で表し, f を a から b への射という. また, 対象 a から対象 b への射の集まりを $\mathrm{Hom}_{\mathbf{C}}(a, b)$ で表す (もちろんこれも集合とは限らない).
2. 2 つの射 $f, g \in \mathrm{Mor}(\mathbf{C})$ が $\mathrm{cod}(f) = \mathrm{dom}(g)$ を満たすとき, f, g の **composition morphism(合成射)** と呼ばれる射 $g \circ f$ が定められていて, $\mathrm{dom}(g \circ f) = \mathrm{dom}(f), \mathrm{cod}(g \circ f) = \mathrm{cod}(g)$ を満たす. すなわち, $f : a \to b, g : b \to c \Rightarrow g \circ f : a \to c$ である.
3. 射の合成は結合律を満たす. すなわち, 射 $f : a \to b$, $g : b \to c$, $h : c \to d$ に対して $h \circ (g \circ f) = (h \circ g) \circ f$ である.
4. 各対象 $a \in \mathrm{Ob}(\mathbf{C})$ に対して, **identity(恒等射)** と呼ばれる射 $\mathrm{id}_a : a \to a$ が存在し, 射の合成に関する単位元となる. すなわち, 射 $f : a \to b$ に対して, $f \circ \mathrm{id}_a = f$, $\mathrm{id}_b \circ f = f$ である.

文脈から明らかな場合は $a \in \mathrm{Ob}(\mathbf{C}), f \in \mathrm{Mor}(\mathbf{C})$ を $a \in \mathbf{C}, f \in \mathbf{C}$ と書き表す. id_a を 1_a と書いたりもする.

例 7.1.2 | ⟨ Set⟩

対象を集合, 射を集合から集合への写像とすれば, これは圏となる. この圏を **Set** で表す. 詳しく書くと以下のようになる.

- $\mathrm{Ob}(\mathbf{Set}) :=$ 全ての集合の集まり
- $\mathrm{Mor}(\mathbf{Set}) :=$ 集合から集合への全ての写像の集まり
- 集合 X から集合 Y への写像 f に対して, $\mathrm{dom}(f) = X, \mathrm{cod}(f) = Y$ と定める.
- 射の合成を写像の合成で定める. もちろんこれは結合律を満たす.
- 恒等射を恒等写像で定める. すなわち $X \in \mathrm{Ob}(\mathbf{Set}), \mathrm{id}_X : X \to X := x \mapsto x$

例 7.1.3 | ⟨ Grph⟩

対象をグラフ, 射をグラフからグラフへのグラフ準同型写像とすれば, これは圏となる. この圏を **Grph** で表す. 詳しく書くと以下のようになる.

- $\mathrm{Ob}(\mathbf{Grph}) \coloneqq$ 全てのグラフの集まり
- $\mathrm{Mor}(\mathbf{Grph}) \coloneqq$ グラフからグラフへの全てのグラフ準同型写像の集まり
- グラフ G からグラフ H へのグラフ準同型写像 f に対して, $\mathrm{dom}(f) = G, \mathrm{cod}(f) = H$ と定める.
- 射の合成をグラフ準同型写像の合成で定める. グラフ準同型写像の合成写像はグラフ準同型写像である. もちろんこれは結合律を満たす.
- 恒等射を恒等グラフ写像で定める. すなわち $G \in \mathrm{Ob}(\mathbf{Grph}), \mathrm{id}_G : G \to G \coloneqq v \in V(G), v \mapsto v$

7.2 note

参考にしたのは [12]

8 connectivity

8.1 2-connected graph

最も単純な 2-連結グラフは cycle である. 他のすべての 2-連結グラフも, cycle に path を加えることで作ることが出来る.

定理 8.1.1

cycle は 2-連結であり, cycle 上の任意の 2 点 x, y に対して, x と y を結ぶ C 上の path が 2 本存在しそれらは独立である.

Proof. 任意の cycle$C$ は $C = P + x_0x_{k-1}, P = x_0x_1 \cdots x_{k-1}$: path ($k \geq 3$) とする. 今, C 上の任意の点 x_i に対して $C - x_i = x_{i+1} \cdots x_{k-1}x_0 \cdots x_{i-1}$ は path となり, 連結である. また, C 上の任意の 2 点 $x_i, x_j (i \leq j)$ に対して $x_ix_{i+1} \cdots x_{j-1}x_j$ と $x_jx_{j+1} \cdots x_{k-1}x_0 \cdots x_{i-1}x_i$ は x_i と x_j を結ぶ C 上の独立な path となる.

□

定理 8.1.2

(有限) グラフ G に対して, 以下は同値である.

1. G は 2-連結.
2. G は閉路から始めて, すでに構築したグラフ H に H-path を加えることで構築できる.
3. G は連結で, G の任意の 2 点に対してそれを含む G 上の cycle が存在する.

Proof. (3.)⇒(1.):G が 2-連結でないとすると, 仮定より 1-連結であるため, $z \in G$ $s.t.$ $G - z$: 非連結 が存在する. $G - z$ が非連結より, $G - z$ 上で結ばれていないような 2 点 x, y が存在する. すなわち $^\exists x, y$ $s.t.$ $^\forall P : x - y$path, P $\notin$ G. 今, $x, y \in G - z \subset G$ と仮定より, x, y を含むような G 上の cycle C が存在する. $z \notin C$ とすると, $C \subset G - z$ より x, y を結ぶ $G - z$ 上の path が存在し, 矛盾する. $z \in C$ とすると, cycle は 2-連結より $C - z$ は連結で $x, y \neq z$ であるため, x, y を結ぶ $G - z$ 上の path が存在し, 矛盾する. したがって G は 2-連結であり, (3.)⇒(1.) が示せた.

(1.)⇒(2.):G を 2-連結グラフとすると, G は cycle を含む[*12]. したがって, (2.) のように構築される極大な部分グラフ H を含む. 仮に $xy \in E(G) \backslash E(H),\ x, y \in H$ が存在するとすると xy は H-path となり, H の極大性に反する. そのため $x, y \in H \Rightarrow xy \in H$ であるから H は G の誘導部分グラフである. 今 $G \neq H$ とすると, H が G の誘導部分グラフであることから $|G| > |H|$ がわかる. よって $G - H \neq \emptyset$ であり, G の連結性より vw $s.t.$ $v \in G - H \wedge w \in H$ が存在する. また G は 2-連結であるため, v-H path $P \subset G - w$ が存在する. このとき, wvP は H-path であり, G に含まれている. したがって $H \cup wvP$ は H より大きい (H を含む)(2.) のように構築されるグラフであり, これは H の

[*12] (3.)⇒(1.) より

極大性に反する. したがって $G=H$ であり, (1.)⇒(2.) が示せた.

(2.)⇒(3.):G を cycle とすると, 明らかに (3.) を満たす. 今, H を (3.) を満たすグラフ, $P=x\cdots y$ を H-path , すなわち $H\cap P=(\{x,y\},\emptyset)$ とする. H が (3.) を満たすことから, $^\forall z\in H$ に対して x と y を結ぶ path $Q=x\cdots z\cdots y$ $s.t.$ $Q\subset H$ が存在する*13. よって $P\cap Q=(\{x,y\},\emptyset)$ より $P\cup Q$ は cycle となり*14. この cycle は P 上の任意の点と H 上の任意の点を含む cycle となる. また, この cycle は P 上の任意の 2 点を含む cycle にもなっている. したがって, $H\cup P$ は (3.) を満たすため, (2.) で構築されるグラフは (3.) を満たすことがわかり, (2.)⇒(3.) が示せた.

(1.)⇒(2.)⇒(3.)⇒(1.) より, (1.),((2.),(3.) は同値である. □

note

(1.)⇔(3.) が [1] にある.

8.2 3-connected graph

補題 8.2.1

G: グラフ, $e=xy\in G$,

G: 連結 ⇔ G/e: 連結

Proof. G: 連結とすると, $^\forall a,b\in G, ^\exists P$:$a$ と b を結ぶ G 上の path . ここで $\{a,b\}\cap\{x,y\}=\emptyset$, $P'=G/e[P\cup\{v_{xy}\}]$ とすると,

1. $P\cap\{x,y\}=\emptyset$ のとき, $P\subset G-\{x,y\}=G/e-v_e\subset G/e$.
2. $P\cap\{x,y\}=\{x\}(\text{or }\{\text{y}\})$ のとき, P: 連結と $P\ni x(\text{or y})$ より P' は連結であり, $a,b\in P'\subset G/e$.
3. $P\cap\{x,y\}=\{x,y\}$ のとき, P: 連結と $P\ni x,y$ より P' は連結であり, $a,b\in P'\subset G/e$.

であるから, a と b を結ぶ G/e 上の path が存在することがわかる. また, $\{a,b\}\cap\{x,y\}=\{x(\text{or y})\}$ の場合は (2.) の最後を $v_{xy},b(\text{or a},\text{v}_{\text{xy}})\in \text{P}'\subset \text{G/e}$ とすればよい. $\{a,b\}\cap\{x,y\}=\{x,y\}$ の場合は a,b は G/e 上で一点 $v_{x,y}$ になる. よって G/e: 連結である. 逆も同様に示せる. □

補題 8.2.2

G: グラフ, $e=xy\in G$, $x,y,v_{xy}\notin S$:vertices set,

$(G-S)/e=G/e-S$

*13 $x,z\in H$ より, x,z を含むような cycle C_1 が存在し, ゆえに x と z を結ぶような path P_1 が存在する

*14 $Q=x\cdots z\cdots y'y$ として, $yPxQy'+y'y$ は cycle であり, $yPxQy'+y'y=P\cup Q$ である.

Proof.

$$
\begin{aligned}
V((G-S)/e) &= (V(G)\backslash S)\backslash\{x,y\}\cup\{v_{xy}\}\\
&= (V(G)\backslash\{x,y\}\cup\{v_{xy}\})\backslash S(x,y,v_{xy}\notin S \text{ より})\\
&= V(G/e-S)
\end{aligned}
$$

$$
\begin{aligned}
E((G-S)/e) = &\{vw\in E(G-S)|\{x,y\}\cap\{v,w\}=\emptyset\}\cup\\
&\ \{v_{xy}w|w\in V(G-S)\backslash\{x,y\}\ s.t\ xw\in E(G-S)\vee yw\in E(G-S)\}\\
= &\{vw\in E(G-S)|\{x,y\}\cap\{v,w\}=\emptyset\}\cup\\
&\ \{v_{xy}w|w\in V(G-S)\backslash\{x,y\}\ s.t\ xw\in E(G-S)\vee yw\in E(G-S)\}\\
= &\{vw\in E|(\{x,y\}\cup S)\cap\{v,w\}=\emptyset\}\cup\\
&\ \{v_{xy}w|w\in V(G-S)\backslash\{x,y\}\ s.t\ xw\in E\vee yw\in E\}\\
&\ \because V(G-S)=V-S,\\
&\quad w\in V(G-S)\wedge x,y\notin S\Rightarrow(xw(yw)\in E\Leftrightarrow xw(yw)\in E(G-S))\\
= &\{vw\in E|(\{x,y\}\cup S)\cap\{v,w\}=\emptyset\}\cup\\
&\ \{v_{xy}w|w\in V\backslash\{x,y\}\ s.t\ (xw\in E\vee yw\in E)\wedge w\notin S\}\\
= &\{vw\in E(G/e)|\{v,w\}\cap S=\emptyset\}\\
= &E(G/e-S)
\end{aligned}
$$

□

補題 8.2.3

G: グラフ, $e=xy\in G$, $x,y,v_{xy}\notin S$:vertices set,
$(G-S)$: 連結 $\Leftrightarrow$ $G/e-S$: 連結

Proof. 補題 8.2.1, 補題 8.2.2 よりわかる. □

定理 8.2.4

G:3-連結グラフ,
$|G|>4\Rightarrow^{\exists}e\in E(G)\ s.t.\ G/e:3\text{−}$連結

Proof. そのような辺が存在しない, つまり $^{\forall}e=xy\in G\ s.t.\ \kappa(G/e)\leq 2$ とする. すなわち $^{\exists}S\subset G/e$: vertices set s.t. |S| $\leq 2\wedge$ G/e − S: 非連結. また, $e=xy$ によって G を縮約した際に得られる頂点を v_{xy}と書き表すこととする. 今, $v_{xy}\notin S$ とすると, $S\subset G/e-v_{xy}=G-\{x,y\}$ より $x,y\notin S$ である. よって補題 8.2.3 より $G-S$: 非連結となり, $\kappa(G)\leq 2$ となり G:3-連結グラフに矛盾する. よって $v_{xy}\in S$ である. また $|S|=1$ すなわち $S=\{v_{xy}\}$ とすると, これも $G/e{-}S=G{-}\{x,y\}$ より $\kappa(G)\leq 2$ となり矛盾する. したがって $|S|=2\wedge v_{xy}\in S$ がわかる. S の元で v_{xy} ではないほうの頂点を z とすると, $z\notin\{x,y\}$ より $G/e-S=G/e-\{v_{xy}\}-\{z\}=G-\{x,y,z\}$ である.

以上をまとめると $^{\forall}x,y\in G\ s.t.\ xy\in E(G)$, $^{\exists}z$, $G-\{x,y,z\}$: 非連結 が導ける. ここで, $G-\{x,y,z\}$ は非連結より 2 つ以上の連結成分を持ち, G が 3-連結であることから x,y,z はすべての連結

成分と隣接していることに注意する. ここで $G-\{x,y,z\}$ の中で一番位数が小さい連結成分を C とし, $|C|$ が最小になるように x,y を取り直す. $v\in V(C)\ s.t.\ vz\in E(G)$ とすると v の存在性は明らか. また, 仮定より G/vz は 3-連結ではないため, $^{\exists}w\in G\ s.t.\ G-\{z,v,w\}$: 非連結. x と y が隣接していることから, $G-\{z,v,w\}$ は $D\cap\{x,y\}=\emptyset$ となる連結成分 D をもつ. $u\in V(D)\ s.t.\ vu\in E(G)$ とすると u の存在性は明らかであり, $v\in V(C)$ より $u\in V(C)$ $i.e.$ $C\cap D\neq\emptyset$ がわかる. $x,y,z\notin D$ より D は $G-\{x,y,z\}$ の連結成分の部分グラフであるため, $D\subseteq C$. また $v\notin D$, $v\in C$ であるため $D\subsetneq C$ である. ゆえに C の最小性に反する. □

定理 8.2.5

G が 3-連結グラフ (ただし K_3 は除く) であるための必要十分条件は, 以下を満たすようなグラフの列 $G_0,\cdots,G_n$ が存在することである.

1. $G_0=K_4\wedge G_n=G$.
2. $^{\forall}i<n,^{\exists}xy\in E(G_{i+1}),(d(x),d(y)\geq 3\wedge G_i=G_{i+1}/xy)$.

Proof. 必要性 $(\Rightarrow)$:G:3-連結 $\Rightarrow$ グラフの列 $G_0,\cdots,G_n$ が存在.

位数が 4 である 3-連結なグラフは K_4 のみであるから, 定理 8.2.4 より, 各 G_i:3-連結となるようなグラフの列 $G=G_n,\cdots,G_0=K_4(^{\exists}xy\in E(G_{i+1}),G_i=G_{i+1}/xy)$ が構成することができる, また, 任意のグラフ H に対し, $\kappa(H)\leq\lambda(H)\leq\delta(H)=\min\{d(x)|x\in H\}$ であるから, この列は (2.) を満たす.

十分性 $(\Leftarrow)$:G:3-連結 $\Leftarrow$ グラフの列 $G_0,\cdots,G_n$ が存在.

(2.) のときに, G_i:3-連結 $\Rightarrow$ G_{i+1}:3-連結を示す. これが示せると K_4 は 3-連結であることとグラフの列が有限であることから帰納的に $G_n=G$:3-連結が導ける.

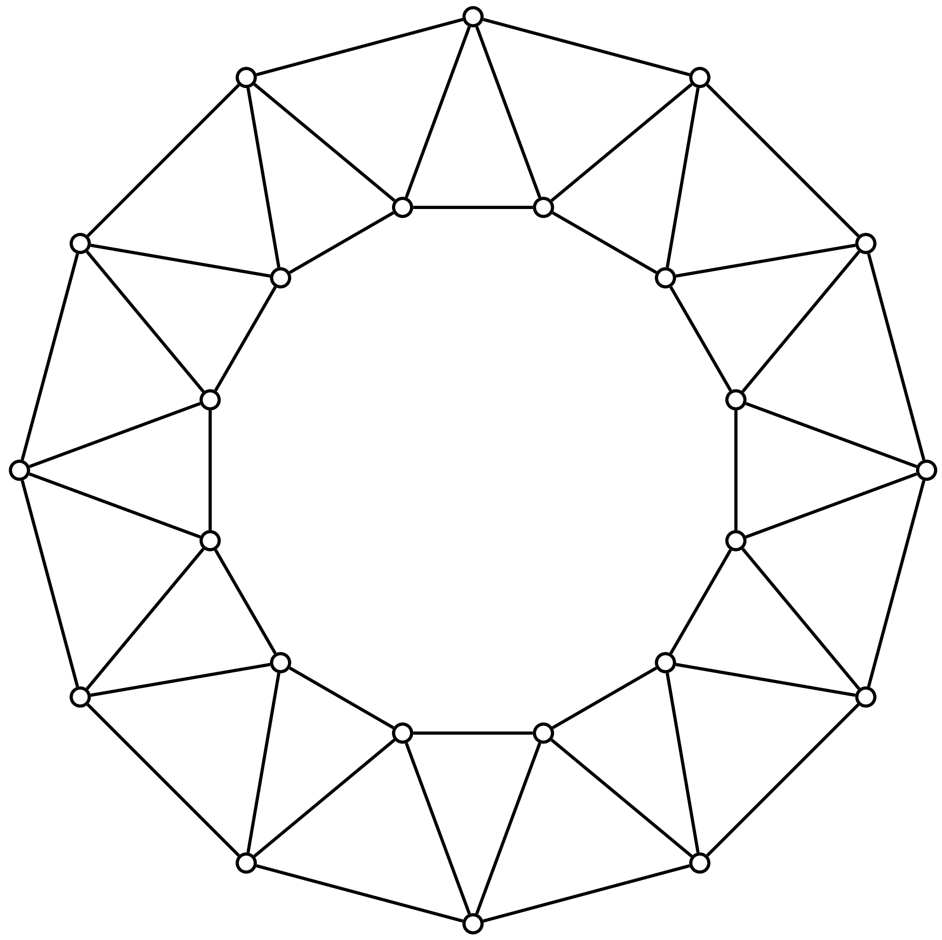
G_i:3-連結であり G_{i+1} が 3-連結でない, つまり $\kappa(G_{i+1})\leq 2$ とする. すなわち $^{\exists}S\subset G_{i+1}$: vertices set s.t. $|\mathrm{S}|\leq 2\wedge \mathrm{G_{i+1}}-\mathrm{S}$: 非連結. また, xy によって G_{i+1} を縮約した際に得られる頂点を v_{xy} と書き表すこととする. 今, $x,y\notin S$ とすると, $S\subset G-\{x,y\}=G/xy-v_{xy}$ より $v_{xy}\notin S$ である. よって補題 8.2.3 より $G_{i+1}/xy-S$: 非連結となり, $\kappa(G_i)\leq 2$ となり G_i:3-連結グラフに矛盾する. よって $\{x,y\}\cap S\neq\emptyset$ である. また $S\subseteq\{x,y\}$ とすると, これも $G_{i+1}/xy-\{v_{xy}\}=G_{i+1}-\{x,y\}\subset G_{i+1}-S$ より $\kappa(G_i)\leq 2$ となり矛盾する. したがって $|S|=2\wedge x$ (resp y, 以降はxの場合で証明する) $\in S$ がわかる. S の元で x ではないほうの頂点を z とすると, $z\notin\{x,y\}$ より $G_{i+1}-S=G_{i+1}-\{x,z\}$ である.

ここで, $G_{i+1}-\{x,z\}$ は非連結より, 各連結成分 $C_k(k\in\mathbb{N})$ に分離することが出来, 特に $y\in C_1$ とすることが出来る. このとき, C_1 に y 以外の元 v が存在するとすると, $G_i-\{v_{xy},z\}=G_{i+1}-\{x,y,z\}$ であり G_i:3-連結であるため, $G_{i+1}-\{x,y,z\}$: 連結である. よって C_2-v path P が存在し, $P\subset G_{i+1}-\{x,y,z\}\subset G_{i+1}-\{x,z\}$ であるが, これは $v\in C_1$ であるため C_1 が連結成分であることに矛盾する. よって $C_1=y$ である. y は $G_{i+1}-S$ における連結成分であるから, $N(y)\subset S\cup V(C_1\backslash y)=S$ より $d(y)=|N(y)|\leq 2$ であるため, 仮定に反する.

□

これは 3-連結だけの場合で, 4-連結だと反例がある

例 8.2.6



どの頂点を縮約しても 3-連結になる [2].

9 multigraph(マルチグラフ)

10 digraph(有向グラフ)

定義 10.0.1 | ⟨ 隣接 (adjacent)⟩

グラフ D の有向辺 (u, v) に対して, u は v へ隣接している (u is adjacent to v) といい, 逆に v は u から隣接している (v is adjacent from u) という.

定義 10.0.2 | ⟨ 近傍 (out-neighborhood,in-neighborhood)⟩

グラフ D の頂点 v に対して,

$$N^{+}(v) = \{x \in V | (v, x) \in E\}$$

$$N^{-}(v) = \{x \in V | (x, v) \in E\}$$

をそれぞれグラフ D における v の外近傍 (out-neighborhood), 内近傍 (in-neighborhood) という.

定義 10.0.3 | ⟨ 次数 (outdegree,indegree)⟩

グラフ D の頂点 v に対して,

$$od(v) = |\{(v,x) \in E|^{\exists}x \in V\}|$$
$$id(v) = |\{(x,v) \in E|^{\exists}x \in V\}|$$

をそれぞれグラフ D における v の外次数 (outdegree), 内次数 (indegree) という. また, v の次数 $d(v)$ を $d(v) = od(v) + id(v)$ と定める.

つまり, 次数とは v に接続している辺の本数である. また, 明らかに $od(v) = |N^+|, id(v) = |N^-|$ である.

10.1 ネットワーク (Network)

定義 10.1.1 | ⟨ ネットワーク (Network)⟩

有向グラフ $D = (V,E)$ が, source と sink という 2 つの異なる頂点 u, v をもち, また $c : E \to \mathbb{R}^+(\mathbb{R}^+ = \{|x||x \in \mathbb{R}\})$ が存在するとき, $N = (D,v,u,c)$ をネットワーク (Network) という.

D を N の underlying digraph , c を N の容量関数 (capacity function) , $e = (x,y) \in E$ に対する $c(e) = c(x,y)$ の値を e の capacity , v, u 以外の $N(D)$ の頂点を N の intermediate vertex という.

表記 10.1.2

有向グラフ D, $g : E(D) \to \mathbb{R}$, $X, Y \subset V(D)$ に対して,

$$[X,Y] = \{(x,y)|x \in X, y \in Y\}$$
$$g(X,Y) = \sum_{(x,y)\in[X,Y]} g(x,y) \ ([X,Y] = \emptyset \Rightarrow g(X,Y) = 0)$$

また, $x \in V(D)$ のとき,

$$g^+(x) = \sum_{y\in N^+(x)} g(x,y) \ , \ g^-(x) = \sum_{y\in N^-(x)} g(y,x)$$

とし, より一般に $X \subseteq V(D)$ のとき,

$$g^+(X) = \sum_{x\in X} g^+(x) \ , \ g^-(X) = \sum_{x\in X} g^-(x)$$

10.2 フローネットワーク (Flow Network)

定義 10.2.1 | ⟨ Flow⟩

ネットワーク $N = (D,u,v,c)$ に対して, $f : E(D) \to \mathbb{R}$ が以下を満たしているとき, f は N のフロー (flow) であるという.

1. ${}^{\forall}a \in E(D), 0 \leq f(a) \leq c(a)$
2. $\forall x \in V(D) \setminus \{u, v\}, f^{+}(x) = f^{-}(x)$

$a = (x, y) \in E(D)$ のとき, $f(a) = f(x, y)$ を a に沿ったフローという. また, (2.) をフローの保存則 (conservation equation) という.

他にも, $f : E(D) \to 0$ の場合, f はフローになる. これをゼロフローという. $X \subset V(D)$ に対して, $f^{+}(X) - f^{-}(X)$ を X から出ていくネットフロー, $f^{-}(X) - f^{+}(X)$ を X に入っていくネットフローという. とくに $x \in V(D)$ に対して, $f^{+}(x) - f^{-}(x)$ を x から出ていくネットフロー, $f^{-}(x) - f^{+}(x)$ を x に入っていくネットフローという. x が intermediate vertex であるとき, これらは (2.) より 0 となる.
$a \in E(D)$ に対して $f(a) = c(a)$ であるとき, a は f について飽和している (saturated) といい, そうでないときには不飽和 (unsaturated) であるという.

定理 10.2.2

グラフ $N = (D, u, v, c)$, f を N 上のフローとするとき,

$$f^{+}(u) - f^{-}(u) = f^{-}(v) - f^{+}(v)$$

が成り立つ.

Proof.

$$\sum_{x \in V(D)} f^{+}(x) = \sum_{x \in V(D)} f^{-}(x) \; i.e. \; f^{+}(V(D)) = f^{-}(V(D))$$

であるから, 定義 10.2.1 の (2.) より

$$f^{+}(u) - f^{-}(u) = f^{-}(v) - f^{+}(v)$$

が導ける. □

10.3 最大フロー (Maximum Flow)

定義 10.3.1 | ⟨ value⟩

$N = (D, u, v, c)$ において, source u から出ていくネットフローをフロー f の value といい, $\mathrm{val}(f)$ で表す. すなわち $\mathrm{val}(f) = f^{+}(u) - f^{-}(u)$ である.

定義 10.3.2 | ⟨ 最大フロー (Maximum Flow)⟩

$N = (D, u, v, c)$ に対して, value が最大となるフロー f のことを N の最大フロー (Maximum flow) という. すなわち ${}^{\forall}f'$:N 上のフロー, $\mathrm{val}(f) \geq \mathrm{val}(f')$ である.

一意には定まらないが存在する. これは定義 10.2.1 の (1.) より従う.

定義 10.3.3 | ⟨ カット (Cut)⟩

$N=(D,u,v,c)$. $X\subset V(D)$ に対して $\overline{X}=V(D)-X$ と定める． $u\in X\wedge v\in\overline{X}$ であるとき, $A=[X,\overline{X}]\subset E(D)$ を N のカット (cut) という．

u から v への path は必ず K を通らなければならない,

補題 10.3.4

$N=(D,u,v,c)$: ネットワーク, f: フロー, $X\subset V(D)$,

$$f^+(X)-f^-(X)=f(X,\overline{X})-f(\overline{X},X)$$

Proof.

$$\begin{aligned}
f^+(X)-f^-(X) &= \sum_{x\in X} f^+(x)-\sum_{x\in X} f^-(x)\\
&= \sum_{x\in X}\left(\sum_{y\in N^+(x)} f(x,y)\right)-\sum_{x\in X}\left(\sum_{y\in N^-(x)} f(y,x)\right)\\
&= \sum_{x\in X}\left(\sum_{y\in N^+(x)\cap X} f(x,y)\right)+\sum_{x\in X}\left(\sum_{y\in N^+(x)\cap\overline{X}} f(x,y)\right)\\
&\quad -\sum_{x\in X}\left(\sum_{y\in N^-(x)\cap X} f(y,x)\right)+\sum_{x\in X}\left(\sum_{y\in N^-(x)\cap\overline{X}} f(y,x)\right)\\
&= \sum_{a\in[X,X]} f(a)+\sum_{a\in[X,\overline{X}]} f(a)-\sum_{a\in[X,X]} f(a)-\sum_{a\in[\overline{X},X]} f(a)\\
&\quad ((\because)[A,B]=\{(x,y)\in E(D)|x\in A,y\in B\}\\
&\qquad =\{(x,y)|x\in A,y\in B\cap N^+(x)\}(\{(x,y)|x\in A\cap N^-(x),y\in B\}))\\
&= \sum_{a\in[X,\overline{X}]} f(a)-\sum_{a\in[\overline{X},X]} f(a)\\
&= f(X,\overline{X})-f(\overline{X},X)
\end{aligned}$$

□

定義 10.3.5 | ⟨ 容量 (Capacity)⟩

$N=(D,u,v,c)$. $K=[X,\overline{X}]$ を N のカットとする. このとき, カットに含まれる arc の容量の合計値をカット K の容量 (capacity) といい,$\mathrm{cap}(K)$ と表す． すなわち

$$\mathrm{cap}(K)=c(X,\overline{X})=\sum_{(x,y)\in[X,\overline{X}]} c(x,y)$$

である.

定理 10.3.6

$N = (D, u, v, c)$: ネットワーク, f:N のフロー, $K = [X, \overline{X}]$:N のカット, $\mathrm{val}(f) = f^+(X) - f^-(X) \leq \mathrm{cap}(K)$

Proof. 仮定より $v \notin X$ であり, 定義 10.2.1 の (2.) より, ${}^\forall x \in X - \{u\}, f^+(x) - f^-(x) = 0$ である. よって

$$\begin{aligned} f^+(X) - f^-(X) &= \sum_{x \in X} f^+(X) - \sum_{x \in X} f^-(X) \\ &= \sum_{x \in X} (f^+(x) - f^-(x)) \\ &= (f^+(u) - f^-(u)) \\ &= \mathrm{val}(f) \end{aligned}$$

が成り立つ. また, ${}^\forall a \in E(D), 0 \leq f(a) \leq c(a)$ であるから, 補題 10.3.4 より,

$$\begin{aligned} f^+(X) - f^-(X) &= f(X, \overline{X}) - f(\overline{X}, X) \\ &\leq f(X, \overline{X}) \\ &\leq c(X, \overline{X}) \\ &= \mathrm{cap}(K) \end{aligned}$$

となり, 示せた. □

10.4 最小カット (Minimum Cut)

定義 10.4.1 | ⟨ 最小カット (Minimum Cut)⟩

$N = (D, u, v, c)$ に対して, capacity が最小となるカット K のことを N の最小カット (Minimum cut) という. すなわち ${}^\forall K'$:N 上のカット, $\mathrm{cap}(K) \geq \mathrm{cap}(K')$ である.

一意には定まらないが, 存在する. これはネットワークの定義より従う.

系 10.4.2

$N = (D, u, v, c)$: ネットワーク, f:N のフロー, K:N のカット. このとき, $\mathrm{val}(f) = \mathrm{cap}(K)$ ならば, f は N の最大フローであり, K は N の最小カットである.

Proof. $\mathrm{val}(f) = \mathrm{cap}(K)$ とすると, 任意の N のフロー f', 任意の N のカット K' に対して, 定理 10.3.6 より, $\mathrm{val}(f') \leq \mathrm{cap}(K) = \mathrm{val}(f) \leq \mathrm{cap}(K')$ である. よって, f は N の最大フローであり, K は N の最小カットである. □

系 10.4.3

$N = (D, u, v, c)$: ネットワーク, f:N のフロー, $K = [X, \overline{X}]$:N のカット. このとき,

$$({}^\forall a \in [X, \overline{X}], f(a) = c(a)) \wedge ({}^\forall a \in [\overline{X}, X], f(a) = 0)$$

ならば, f は N の最大フローであり, K は N の最小カットである.

Proof. 定理 10.3.6 より,

$$\begin{aligned}\mathrm{val}(f) &= f^+(X) - f^-(X)\\ &= f(X, \overline{X}) - f(\overline{X}, X)\\ &= c(X, \overline{X}) - 0\\ &= \mathrm{cap}(K)\end{aligned}$$

よって系 10.4.2 より f は N の最大フローであり, K は N の最小カットである. □

最大フローと最小カットであるための条件を与えている. 特に系 10.4.2 は定理 11.1.1 の十分条件を与えている. 必要条件を与えるために, ここで一つ便利な概念を導入する.

定義 10.4.4 | ⟨ semipath⟩

有向グラフ D に対して semipath とは, 以下を満たすような空でない有向グラフ $P = (V, E)$ のことである.

$V = \{\omega_i | i = 0, \cdots, k\}, E = \{a_i \in E(D) | a_i = (\omega_{i-1}, \omega_i) \vee a_i = (\omega_i, \omega_{i_1}), i = 1, \cdots k\}$(各$\omega_i$は異なる)

またこのとき, P を ω_0 から ω_k への semipath (ω_0-ω_k semipath) という. またこのとき E の元について, $a_i = (\omega_{i-1}, \omega_i)$ を forward arc , $a_i = (\omega_i, \omega_{i_1})$ を backward arc という.

表記 10.4.5

上の semipath を $P = (\omega_0, a_1, \omega_1, \cdots, \omega_{k-1}, a_k, \omega_k)$ と書き表す.

10.5 f-Augmenting Semipaths

定義 10.5.1

$N = (D, u, v, c)$: ネットワーク, f:N のフロー, $P = (\omega_0, a_1, \omega_1, \cdots, \omega_{k-1}, a_k, \omega_k)$:$D$ の semipath とする. P が以下の条件を満たしているとき, P は f-unsaturated な semipath であるという.

1. $f(a_i) < c(a_i)$ (a_i:forward arc)
2. $f(a_i) > 0$ (a_i:backward arc)

自明な semipath ($P = (\omega_0)$) は f-unsaturated とする. P が f-unsaturated な u-v semipath であるとき, f-augmenting な semipath であるという.

定理 10.5.2

$N = (D, u, v, c)$: ネットワークとする. このとき, f が最大フローであることと D 上に f-augmenting な semipath が存在しないことは同値である.

Proof. $\Rightarrow$)f を最大フローとし, D 上に f-augmenting な semipath P が存在するとする. $P = (\omega_0, a_1, \omega_1, \cdots, \omega_{k-1}, a_k, \omega_k)$ とすると, $\omega_0 = u, \omega_k = v$ である. P の forward arc a_{i_n} について, $c(a_{i_n}) - f(a_{i_n}) > 0$ であり $n \leq k < \infty$ であるため, $p_1 = \min\{c(a_{i_n}) - f(a_{i_n}) | a_{i_n}\text{:forward arc}\} > 0$ が存在する. 同様に P の backward arc a_{i_m} についても, $f(a_{i_n}) > 0$ であるため $p_2 = \min\{f(a_{i_m}) | a_{i_n}\text{:backward arc}\} > 0$ が存在する. $p = \min\{p_1, p_2\}$ とすれば,

$$f'(a) = \begin{cases} f(a) + p & \text{if } a \text{ is a forward arc on } P \\ f(a) - p & \text{if } a \text{ is a backward arc on } P \\ f(a) & \text{if } a \notin E(P) \end{cases}$$

は N のフローとなり*15,

1. $f^+(u) + p = f'^+(u)$ (a_1:forward arc)
2. $f^-(u) - p = f'^-(u)$ (a_1:backward arc)

より $f^+(u) - f^-(u) < f'^+(u) - f'^-(u)$ *i.e.* $\mathrm{val}(f) < \mathrm{val}(f')$ であるから f が最大フローであることに矛盾する. よって f-augmenting な semipath P は存在しないことがわかる.

$\Leftarrow$)f が D 上 f-augmenting な semipath が存在しないような flow とする. このとき f が最大フローであることを示す. 今 $X = \{x \in V(D) | {}^{\exists}P\text{:}f\text{-unsaturated } u\text{-}x \text{ semipath}\}$ とすると, $u \in X, v \notin X$ であるため, $K = [X, \overline{X}]$ は cut となる. ${}^{\forall}a \in [X, \overline{X}], {}^{\forall}b \in [\overline{X}, X], f(a) = c(a), f(b) = 0$ であるから*16, 系 10.4.3 より f は最大フローである. また, K は最小カットとなっている. □

11 最大フロー最小カット定理

11.1 maximum flow minimum cut theorem

定理 11.1.1 | ⟨ maximum flow minimum cut theorem⟩

ネットワーク $N = (D, u, v, c)$ に対して, 最大フローと最小カットの値は一致する. すなわち, f:N のフロー, K:N のカットに対して,

$$f\text{: 最大フロー} \wedge K\text{: 最小カット} \Leftrightarrow \mathrm{val}(f) = \mathrm{cap}(K)$$

---

*15 ω_i $(0 < i < k)$ に対して

1. $f^+(\omega_i) + p = f'^+(\omega_i), f^-(\omega_i) + p = f'^-(\omega_i)$ (a_i, a_{i+1}:forward arc)
2. $f^-(\omega_i) + p - p = f'^-(\omega_i)$ (a_i:forward arc, a_{i+1}:backward arc)
3. $f^+(\omega_i) - p + p = f'^+(\omega_i)$ (a_i:backward arc, a_{i+1}:forward arc)
4. $f^+(\omega_i) - p = f'^+(\omega_i), f^-(\omega_i) - p = f'^-(\omega_i)$ (a_i, a_{i+1}:backward arc)

であるため $f^+(\omega_i) - f^-(\omega_i) = f'^+(\omega_i) - f'^-(\omega_i)$ であり, $x \in V(D) \backslash V(P)$ については明らかに $f^+(x) - f^-(x) = f'^+(x) - f'^-(x)$ である. また, f' は p のとり方から各 arc の容量を超えない. ゆえに f' は flow である.

*16 任意の $(y, z) \in [X, \overline{X}]$ について, $y \in X$ より f-unsaturated u-y semipath が存在し, $z \in \overline{X}$ より f-unsaturated u-z semipath が存在しない. $f(y, z) < c(y, z)$ とすると, f-unsaturated u-y semipath に $f(y, z), z$ を加えたものは f-unsaturated u-z semipath になり矛盾する. よって $f(y, z) = c(y, z)$ である. 同様に, 任意の $(w, x) \in [\overline{X}, X]$ についても $f(w, x) = 0$ が言える.

Proof. $\Leftarrow$) 系 10.4.2 より従う.

$\Rightarrow$)f: 最大フロー, K: 最小カットとする. 定理 10.5.2 より, D 上に f-augmenting な semipath は存在せず, $X = \{x \in V(D) | {}^{\exists}P{:}f\text{-unsaturated } u\text{-}x \text{ semipath}\}$ とすると $K' = [X, \overline{X}]$ は最小カットとなり,

$$f(a) = \begin{cases} c(a) & \text{if } a \in K' \\ 0 & \text{if } a \in [\overline{X}, X] \end{cases}$$

である. よって系 10.4.3 より $\mathrm{val}(f) = \mathrm{cap}(K') = \mathrm{cap}(K)$ となり示せた. □

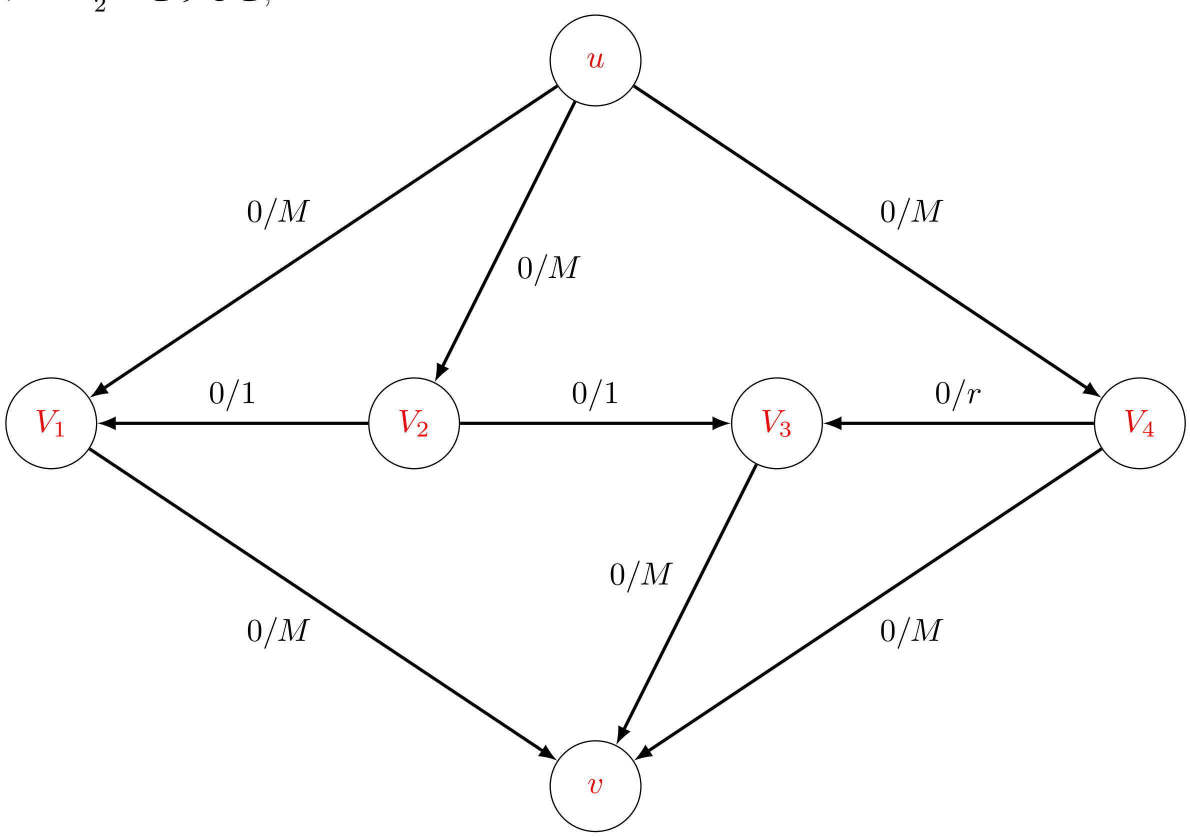
11.2 The Ford-Fulkerson Algorithm

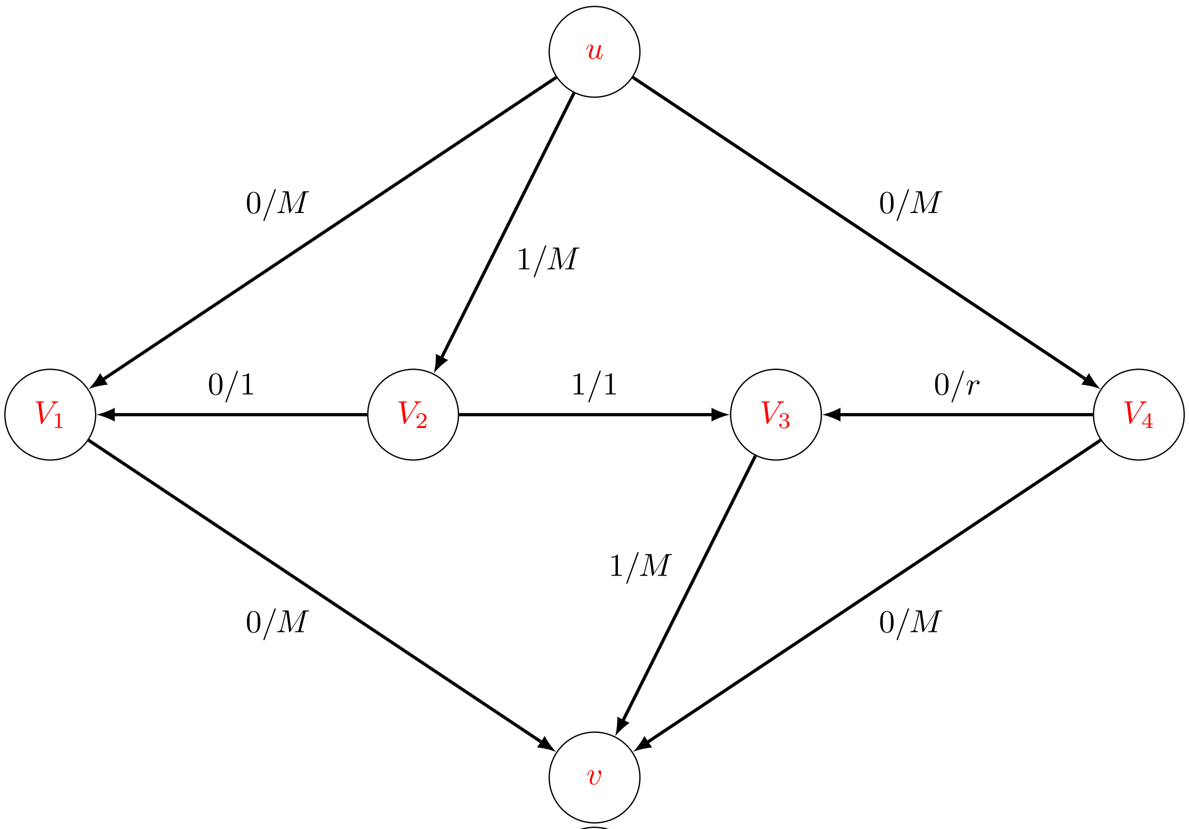
$N = (D, u, v, c)$: ネットワークとする.

1. flow f を一つとる.
2. f-augmenting な semipath を見つける. 見つけれなかった場合は終了する.
3. 定理 10.5.2 の $\Rightarrow$ で作ったように f' を構成する.
4. $f = f'$ として Step 2. に戻る.

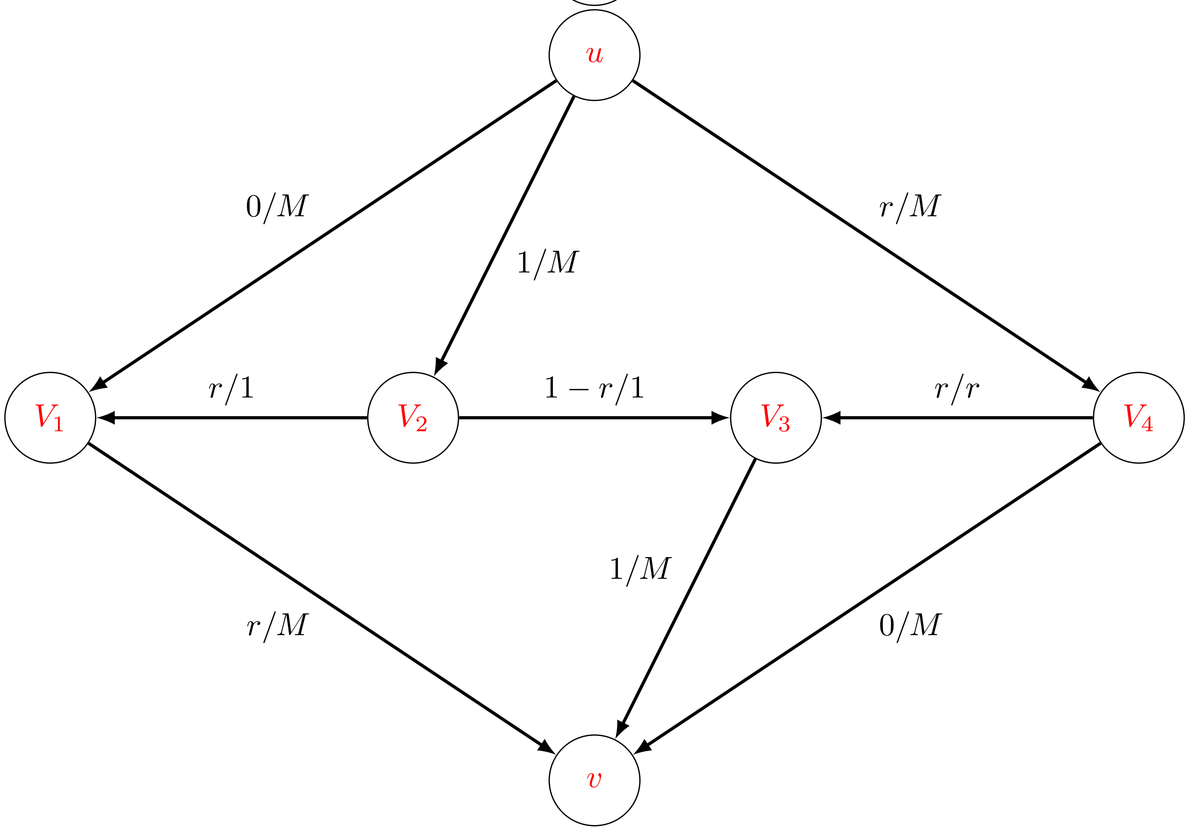
しかしこのアルゴリズムは欠点が多い. 一つはグラフと f-augment semipath の選び方によって, 計算量がとても大きくなるという点. もう一つは容量が無理数だとアルゴリズムが止まらなくなるという点だ [3].

$r = \frac{\sqrt{5}-1}{2}$ とすると,

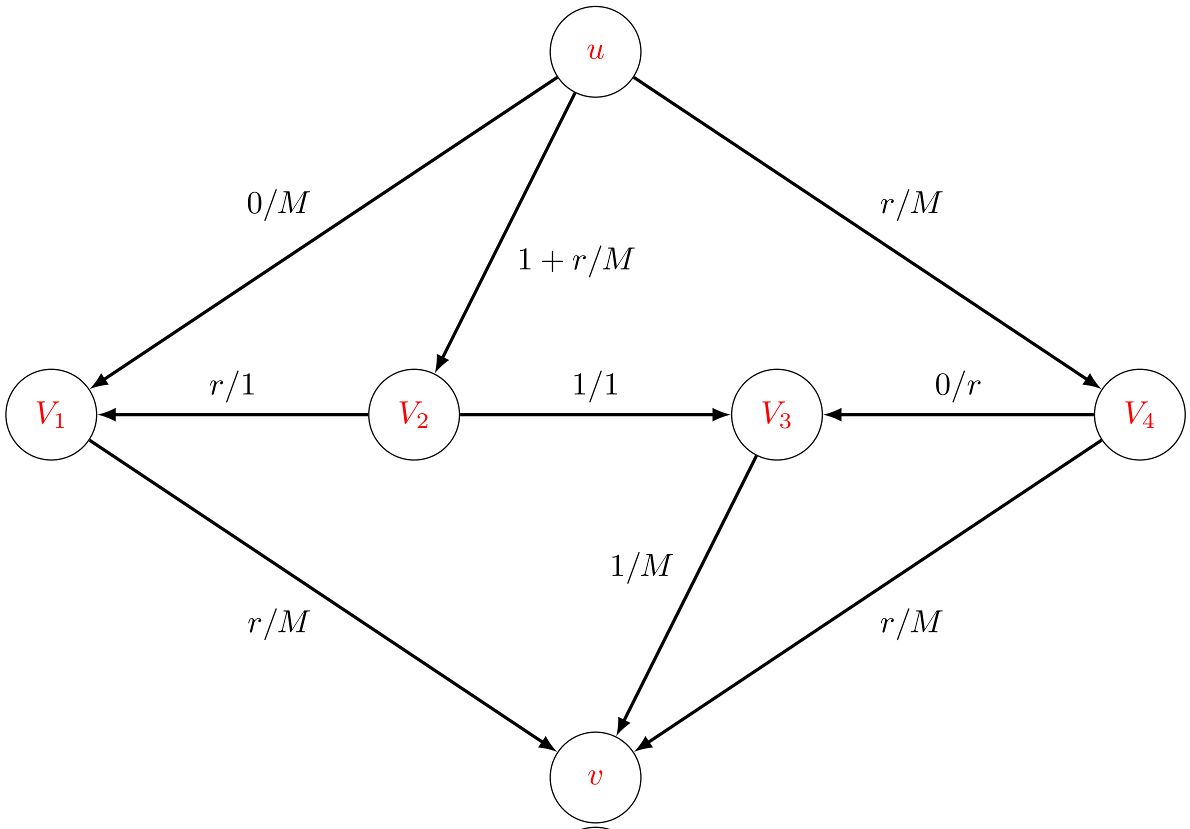


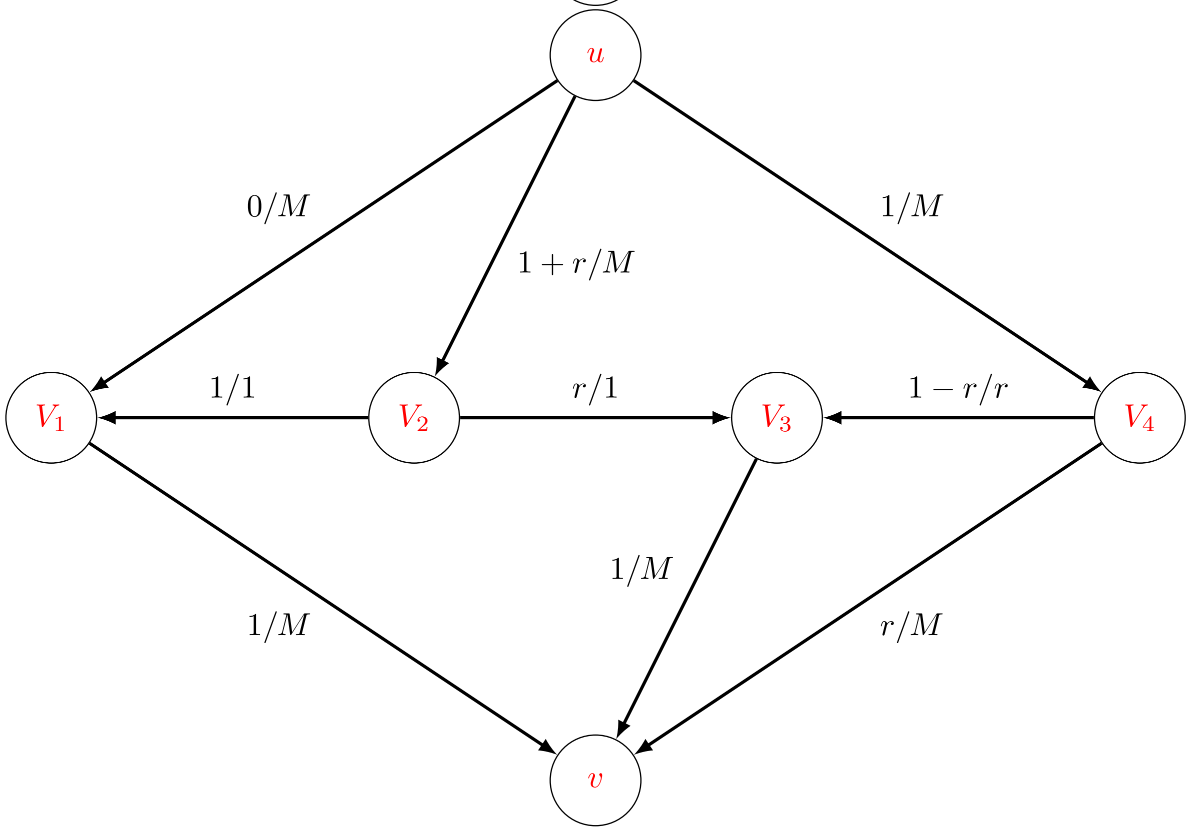


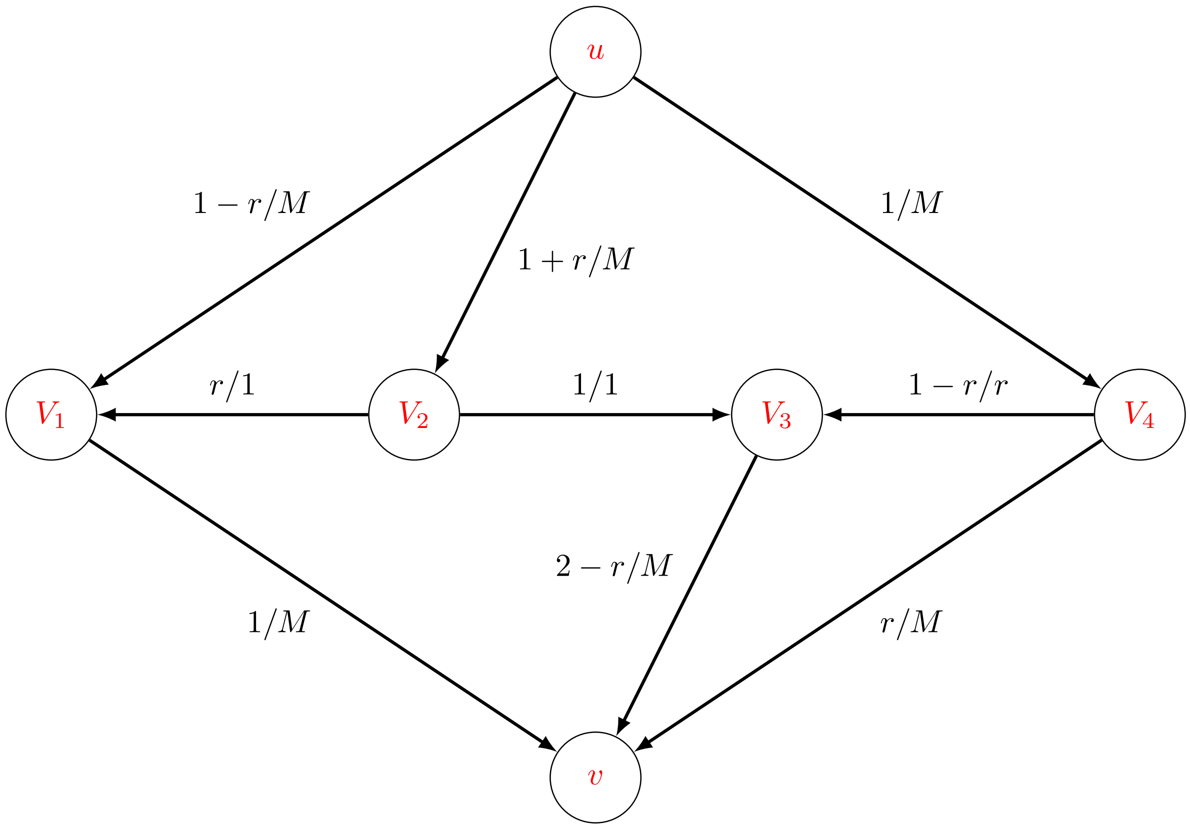
u
$0/M$
$1/M$
$0/M$
V_1
$0/1$
V_2
$1/1$
V_3
$0/r$
V_4
$1/M$
$0/M$
$0/M$
v



u
$0/M$
$1/M$
r/M
V_1
$r/1$
V_2
$1-r/1$
V_3
r/r
V_4
$1/M$
r/M
$0/M$
v


u
0/M
1 + r/M
r/M
V1
r/1
V2
1/1
V3
0/r
V4
1/M
r/M
r/M
v


u
0/M
1 + r/M
1/M
V1
1/1
V2
r/1
V3
1 − r/r
V4
1/M
1/M
r/M
v



11.3 The Edmonds-Karp Algorithm

$N=(D,u,v,c)$: ネットワークとする. 各点にラベルを付け, ラベル付けされているがスキャンされていない頂点のリスト L を用意する. ラベルは 2 つの値のペアである.

1. flow f を一つとる. N の intermediate vertex w において f-unsaturated な u-w semi-path P が存在するときに, P の直前の頂点 x について $(x,w)\in E(P)$ であればラベルは $(x+,\epsilon(w))$, $(w,x)\in E(P)$ であればラベルは $(x-,\epsilon(w))$ とする.
2. u のラベルは $(-,\infty)$ とし, u をリスト L に加える.
3. $L=\emptyset$ ならばとめ. そうでなければ L の最初の元 x(ラベル $(z+,\epsilon(x))$ or $(z+,\epsilon(x))$ を持つ) について,
   3.1. flow f を一つとる.
   3.2. ラベルを付ける. u は $(-,\infty)$ とし, リスト L に加える.
4. v がラベルを持っている場合, Step 5. へ, そうでなければ 3. へ行く.
5. v がラベルを持っている場合, Step 5. へ, そうでなければ 3. へ行く.
6. ラベルを削除し, L から頂点を全て削除し, Step 2. に戻る.

12 quiver(箙)

13 infinite graph(無限グラフ)

[4]

14　わからない問題集

例 **14.0.1** | ⟨ **[9]** の **1J**⟩

G を位数が $n(n \geq 3)$ で $\Delta(G) < n-1$ なグラフとする. グラフ G の任意の 2 頂点 u, v に対して, u, v に隣接している頂点がただ一つ存在するとき, 以下が成り立つことを示せ.

1. G の 2 頂点 x, y が隣接していないとき, $\deg(x) = \deg(y)$
2. G は正則なグラフである.

例 **14.0.2** | ⟨ **[9]** の **3C**⟩

の式の右辺がどちらも偶数だった場合, 等式が成り立たないことを示せ.

索引

参考文献

[1] 2-Connected Graphs. http://www.cs.rpi.edu/~goldberg/14-GT/08-block.pdf.

[2] Connectivity. http://www-sop.inria.fr/members/Frederic.Havet/Cours/connectivity.pdf.

[3] Ford Fulkerson algorithm. https://en.wikipedia.org/wiki/Ford\OT1\textendashFulkerson_algorithm#Non-terminating_example.

[4] Infinite graph. https://www.math.uni-hamburg.de/home/schacht/lehre/SS13/GT/Ch8prelims.pdf.

[5] B. Csaba, D. Kühn, A. Lo, D. Osthus, A. Treglown. PROOF OF THE 1-FACTORIZATION AND HAMILTON DECOMPOSITION CONJECTURES III: APPROXIMATE DECOMPOSITIONS . https://arxiv.org/pdf/1401.4178.pdf.

[6] Fedor V. Fomin, Dieter Kratsch. *Exact Exponential Algorithms*. Springer, 2010.

[7] G. A. Dirac. On rigid circuit graphs. pp. 71–76, 1960.

[8] G. Chartrand, L. Lesniak, P. Zhang. *GRAPH & DIGRAPH, Sixth Edition*. CRC Press?, 2016.

[9] J. H. van Lint, R. M. Wilson. *A COURSE IN Combinatorics, Second Edition*. CAMBRIDGE UNIVERSITY PRESS, 2001.

[10] R. Diestel. *Graph Theory, Fifth Edition*. Springer, 2017.

[11] R. Montgomery, A. Pokrovskiy, B. Sudakov. A proof of Ringel's Conjecture. https://arxiv.org/abs/2001.02665.

[12] University of Montana. The Categories of Graphs. https://scholarworks.umt.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1986&context=etd.