第3章　関係とグラフ

3.1　関係

$A_1, \cdots, A_n$ を集合とするとき，$A_1 \times \cdots \times A_n$ の部分集合を $A_1, \cdots, A_n$ 上の n 項関係 (relation) と呼ぶ．すなわち，R が $A_1, \cdots, A_n$ 上の n 項関係であるとは，

$$R \subset A_1 \times \cdots \times A_n$$

であることである．$A_1 = \cdots = A_n$ のとき，単に「A_1 上の n 項関係」という．また，$n = 2$ のとき,「A_1 上の二項関係 (binary relation)」という．本講義資料では，主として，二項関係を扱う．

例 54　$A = \{a, b\}$, $B = \{1, 2, 3\}$ とする．$R_1 = \{\langle a, 1\rangle, \langle b, 2\rangle\}$ や $R_2 = \{\langle a, 1\rangle, \langle a, 2\rangle, \langle a, 3\rangle\}$ は二項関係である．

R が二項関係のとき，$\langle x, y\rangle \in R$ のことを xRy とも書く．上の例では，$aR_1 1$ や $aR_2 3$ が成立する．

例 55　$A = \{1, 2, 3\}$ とする．以下の R_3, R_4 は A 上の二項関係である．

$$\begin{aligned} R_3 &= \{\langle 1,1\rangle, \langle 2,1\rangle, \langle 2,2\rangle, \langle 3,1\rangle, \langle 3,2\rangle, \langle 3,3\rangle\} \\ R_4 &= \{\langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle\} \end{aligned}$$

R_3 は $\geq$ (以上) という関係を表し，R_4 は $=$ (等しい) という関係を表す．

3.2　二項関係の性質

R を集合 A 上の二項関係とする．すなわち，$R \subset A \times A$ である．
二項関係 R に対して，以下の性質は特に有用なので，名前がついている．

- R が反射的 (reflexive): $\forall x(xRx)$
- R が対称的 (symmetric): $\forall x \forall y(xRy \ \Rightarrow \ yRx)$
- R が推移的 (transitive): $\forall x \forall y \forall z((xRy \wedge yRz) \ \Rightarrow \ xRz)$
- R が反対称的 (antisymmetric): $\forall x \forall y((xRy \wedge yRx) \Rightarrow x = y)$

例 56　自然数の集合 $\mathcal{N}$ 上の二項関係として, $=, \leq, <, \neq, R, S$ を考える．ただし，R, S は以下で定義される関係である．

$$xRy \Leftrightarrow x \text{ と } y \text{ は } 7 \text{ で割った余りが同じ}$$

$$xSy \Leftrightarrow |x - y| \leq 1$$

これらの関係に対する性質を表にすると以下のようになる．

関係	反射的	対称的	推移的	反対称的
$=$	○	○	○	○
$\leq$	○	×	○	○
$<$	×	×	○	○
$\neq$	×	○	×	×
R	○	○	○	×
S	○	○	×	×

たとえば，関係 R が推移的であることの証明は以下のようになる．xRy かつ yRz とする．すると，$(x-y)\ mod\ 7=0$ かつ $(y-z)\ mod\ 7=0$ である．これらより $(x-z)\ mod\ 7=0$ となるので，xRz が成立する．

一方，関係 S は推移的ではない．$0S1$ かつ $1S2$ であるが，$0S2$ でない．

3.3 順序

集合 A 上の二項関係で，反射的かつ推移的かつ反対称的であるものを，A 上の順序 (order) という．なお，後で定義する全順序と区別するため，一般の順序のことを特に半順序 (partial order) ということがある．

例 57 $\mathcal{N}$ 上の二項関係 $\geq$ と $\leq$ と $=$ はどれも順序である．

一方，$\mathcal{N}$ 上の二項関係 $<$ は反射的でないので，順序ではない．

例 58 A を集合とするとき，2^A は A の部分集合からなる集合である．2^A の要素の間の包含関係 $\subset$ は順序である．$\subset$ が推移的であることは，$B\subset C$ かつ $C\subset D$ から $B\subset D$ が導けることからわかる．

最大，最小，極大，極小

集合 A 上の順序 R は一種の大小関係と見なすことができる．すなわち，xRy が成立するとき「x が R に関して y と等しいか小さい」と見なすことができる．このとき，以下の要素が定義される．

- 最大元 (maximum element) a とは $\forall x\in A(xRa)$ を満たす A の元のことである．
- 最小元 (minimum element) b とは $\forall x\in A(bRx)$ を満たす A の元のことである．
- 極大元 (maximal element) c とは $\forall x\in A(cRx\Rightarrow c=x)$ を満たす A の元のことである．
- 極小元 (minimal element) d とは $\forall x\in A(xRd\Rightarrow x=d)$ を満たす A の元のことである．

ただし，これらの元は存在しないこともある．最大元は極大元の 1 つであり，最小元は極小元の 1 つである．最大元，最小元は，存在すれば唯一だが，極大元，極小元は，1 つとは限らない．

例 59 $\mathcal{N}$ 上の順序 $\leq$ に関する最大元，極大元は存在しない．最小元，極小元は 0 だけである．

例 60 $A=\{1,2,3\}$ として，2^A 上の二項関係として，集合の包含関係を考えると，これは順序となる．

この順序に関する最小元は ϕ である．(2^A のどんな要素 S に対しても $\phi\subset S$ が成立するので．)

この順序に関する最大元は A である．(2^A のどんな要素 S に対しても $S\subset A$ が成立するので．)

例 61 $A = \{1,2,3\}$ として，$2^A - \{A\}$ という集合上の二項関係として，集合の包含関係を考える．

今度は，A がはいっていないので，最大元は存在しない．(2^A のどんな要素 S に対しても $S \subset T$ が成立するような T は存在しない．) 極大元は，$\{1,2\}, \{2,3\}, \{3,1\}$ の 3 つある．

全順序

集合 A 上の順序 R で，以下の条件を満たすものを全順序 (total order) という．

$$\forall x \in A \forall y \in A \ (xRy \ \lor \ yRx)$$

すなわち，任意の 2 つの要素が，R に関して比較可能 (どちらかが小さい) のとき，R を全順序という．先の例では，$\mathcal{N}$ 上の $\geq$ や $\leq$ は全順序であるが，$2^{\mathcal{N}}$ 上の包含関係 $\subset$ は全順序でない．たとえば，$\{1,2\}$ と $\{2,3\}$ は，$\subset$ では関係付けられない．

直積集合上の順序 (†)

R_1, R_2 をそれぞれ集合 A_1, A_2 上の順序とする．このとき，直積集合 $A_1 \times A_2$ 上の順序として以下の 2 つのものが考えられる [1]．

- 直積順序 P:

$$\langle x_1, y_1 \rangle \ P \ \langle x_2, y_2 \rangle \Leftrightarrow (x_1 R_1 x_2 \land y_1 R_2 y_2)$$

- 辞書式順序 L:

$$\langle x_1, y_1 \rangle \ L \ \langle x_2, y_2 \rangle \Leftrightarrow ((x_1 \neq x_2 \land x_1 R_1 x_2) \lor (x_1 = x_2 \land y_1 R_2 y_2))$$

R_1, R_2 がいずれも全順序であるとき，辞書式順序も全順序である．一方，R_1, R_2 が全順序であっても，直積順序は全順序とは限らない．国語辞典などの辞書では，全ての単語を一列に並べる必要があるため，辞書式順序を (任意の長さの文字列に対して拡張した) 全順序を用いている．

3.4 同値関係

集合 A 上の二項関係で，反射的, 対称的, 推移的であるものを A 上の同値関係 (equivalence relation) という. 同値関係は, 要素が等しいことを表す関係を一般化したものになっている.

例 62 自然数の集合 $\mathcal{N}$ 上の以下の二項関係は同値関係である．

$$\{\langle n, n \rangle \mid n \in \mathcal{N}\}$$

実数の集合 $\mathcal{R}$ 上の以下の二項関係は同値関係である．

$$\{\langle a, b \rangle \mid a, b \in \mathcal{R} \text{ かつ } |a| = |b|\}$$

一方，$\mathcal{N}$ 上の $\geq$ や $<$ は同値関係ではない．

[1] ここでは，見やすさのために，一番外側の $\forall x_1$ 等を省略した．正確に書くならば，$\forall x_1 \forall x_2 \forall y_1 \forall y_2$ を一番外に補う必要がある

同値類 (†)

R を集合 A 上の同値関係とする．$x \in A$ なる x に対して，x の同値類 $[x]$ とは，以下の集合のことである．

$$[x] = \{y \mid \ y \in A \ \wedge \ xRy\}$$

同値関係 R を「等しい」という概念と見なしたとき，R に関する同値類は x と等しい要素を全て集めた集合である．

商集合 (†)

集合 A 上の同値関係 R に対して，商集合 A/R は，同値類をすべて集めた集合である．

$$A/R = \{[x] \mid x \in A\}$$

例 63 自然数上の二項関係 R を次のように定義する．

$$a\,R\,b \Leftrightarrow (a \bmod 5 = b \bmod 5)$$

この時，同値類は以下のようになる．

$$[0] = \{0, 5, 10, 15, \ldots\}$$
$$[1] = \{1, 6, 11, 16, \ldots\}$$
$$[2] = \{2, 7, 12, 17, \ldots\}$$
$$[3] = \{3, 8, 13, 18, \ldots\}$$
$$[4] = \{4, 5, 14, 19, \ldots\}$$

$\mathcal{N}/R$ は, 以下の集合になる．

$$\mathcal{N}/R = \{[0], [1], [2], [3], [4]\}$$

3.5 関係の合成

R, S をそれぞれ A, B 上の二項関係，B, C 上の二項関係とする．すなわち，$R \subset A\times B, S \subset B\times C$ とする．R と S を合成した関係 $R \circ S$ は次のように定義される [2]．

$$x(R \circ S)y \ \Leftrightarrow \exists z \in B(xRz \wedge zSy)$$

例 64 以下の自然数上の関係 R, S を考える．

$$\begin{aligned} R &= \{\langle x, x+1\rangle \mid x \in \mathcal{N}\} \\ S &= \{\langle x, 2x\rangle \mid x \in \mathcal{N}\} \end{aligned}$$

このとき, 関係の合成, $R \circ R$, $R \circ S$, $S \circ R$ は以下のようになる．

$$\begin{aligned} R \circ R &= \{\langle x, x+2\rangle \mid x \in \mathcal{N}\} \\ R \circ S &= \{\langle x, 2(x+1)\rangle \mid x \in \mathcal{N}\} \\ S \circ R &= \{\langle x, 2x+1\rangle \mid x \in \mathcal{N}\} \end{aligned}$$

[2]なお，一部の教科書では，この定義の R と S を逆に置いた定義を採用しているものがあり，大変混乱させられる．参考書の [4] は良くまとまった本であるが,「逆」の定義を採用しているから注意されたい．本書では，国際的に標準的な定義を採用した．

例 **65** $\mathcal{R}$ 上の二項関係 $>$ と $<$ に対して，

$$(<) \circ (>) = \mathcal{R} \times \mathcal{R}$$

なぜならば, 任意の x, y に対して, ある z が存在して, $x < z$ かつ $z > y$ となるように z を取ることができるから．($z = x + y + 1$ と取ればよい．)

関係の合成に関して以下の性質 (結合律) が成り立つ.

$$R \circ (S \circ T) = (R \circ S) \circ T$$

定義 2 R が A 上の二項関係で n が自然数の時,「R の (関係としての)n 乗」という関係 R^n が以下のように定義できる [3].

$$\begin{aligned} R^0 &= \{\langle a, a\rangle \mid a \in A\} \\ R^{n+1} &= R^n \circ R = R \circ R^n \end{aligned}$$

例 **66** $R = \{\langle a, b\rangle, \langle b, c\rangle, \langle c, d\rangle\}$

R :

R^2 :

R^3 :

3.6 閉包 (closure) (†)

閉包というのは,「何らかの性質に関して閉じたもの」という意味である．p を関係に関する性質とする. 関係 R の p に関する閉包とは，R を含み, p の性質を満たす最小の関係のことである. (ただし，p によっては，そのような最小の関係が存在しないこともある．)
p として,「反射的である」「対称的である」「推移的である」などの性質を取ることが多い.

- R の反射閉包 $r(R)$ とは R を含み，反射的である最小の関係.
- R の対称閉包 $s(R)$ とは R を含み，対称的である最小の関係.
- R の推移閉包 $t(R)$ とは，R を含み，推移的である最小の関係.

また,「最小の関係」というのは，関係は集合の一種なので，集合として一番小さいもの (集合の包含関係に関する最小元) という意味である.

例 **67** $A = \{a, b, c\}$, $R = \{\langle a, a\rangle, \langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle\}$ とする.

[3] ここで定義したものは，集合の章で定義した「集合としての n 乗」とは異なるものである．異なる概念に同じ記号を使うのは混乱の元であるので避けるべきであるが，ここでは伝統的表記に従った.

$$
\begin{aligned}
r(R) &= \{\langle a,a\rangle,\langle a,b\rangle,\langle b,a\rangle,\langle b,c\rangle,\langle b,b\rangle,\langle c,c\rangle\} \\
s(R) &= \{\langle a,a\rangle,\langle a,b\rangle,\langle b,a\rangle,\langle b,c\rangle,\langle c,b\rangle\} \\
t(R) &= \{\langle a,a\rangle,\langle a,b\rangle,\langle b,a\rangle,\langle b,c\rangle,\langle a,c\rangle,\langle b,b\rangle\}
\end{aligned}
$$

また， $S=\{\langle a,b\rangle,\langle b,c\rangle,\langle c,d\rangle\}$ とすると，

$$
\begin{aligned}
r(S) &= \{\langle a,b\rangle,\langle b,c\rangle,\langle c,d\rangle,\langle a,a\rangle,\langle b,b\rangle,\langle c,c\rangle,\langle d,d\rangle\} \\
s(S) &= \{\langle a,b\rangle,\langle b,c\rangle,\langle c,d\rangle,\langle b,a\rangle,\langle c,b\rangle,\langle d,c\rangle\} \\
t(S) &= \{\langle a,b\rangle,\langle b,c\rangle,\langle c,d\rangle,\langle a,c\rangle,\langle b,d\rangle,\langle a,d\rangle\}
\end{aligned}
$$

閉包の構成法

R を A 上の二項関係とすると，反射閉包，対称閉包，推移閉包は以下のようにして計算できる．

(a) $r(R)=R\cup R^0$

(b) $s(R)=R\cup R^c$
ただし R^c は，R の左右を逆にした関係であり R の逆関係と呼ばれる．

$$R^c=\{\langle x,y\rangle \mid \langle y,x\rangle\in R\}$$

(c) A が n 個の要素を持つ有限集合の時
$t(R)=R\cup R^2\cup\cdots\cup R^n$

(d) A が無限集合のとき，R の推移閉包 $t(R)$ は以下のように，無限個の集合の和集合 (極限) として表される．
$t(R)=R\cup R^2\cup R^3\cup\cdots$

例 **68** $A=\{a,b,c,d,e\}$
$R=\{\langle a,b\rangle,\langle b,c\rangle,\langle b,d\rangle,\langle d,e\rangle\}$

この場合，$R^4=R^5=\phi$ である．

例 **69** $R=\{\langle x,x+1\rangle \mid x\in\mathcal{N}\}$ とすると，

$$
\begin{aligned}
R &= \{\langle x,x+1\rangle \mid x\in\mathcal{N}\} \\
R^2 &= \{\langle x,x+2\rangle \mid x\in\mathcal{N}\} \\
&\vdots \\
R^k &= \{\langle x,x+k\rangle \mid x\in\mathcal{N}\}
\end{aligned}
$$

従って，$t(R)$ は関係 $<$ (「より小さい」, “less-than”) となる．

例 70 $\mathcal{N}$ 上の二項関係 `less-than` (より小さい), `not-equal` (等しくない) を考える．それらの閉包は以下の通りである．

$$\begin{array}{rcl}
r(\texttt{less-than}) & = & \{\langle x, y\rangle \mid x \leq y\} \\
s(\texttt{less-than}) & = & \{\langle x, y\rangle \mid x \neq y\} \ (= \texttt{not-equal}) \\
t(\texttt{less-than}) & = & \{\langle x, y\rangle \mid x < y\} \ (= \texttt{less-than}) \\
r(\texttt{not-equal}) & = & \mathcal{N} \times \mathcal{N} \\
s(\texttt{not-equal}) & = & \{\langle x, y\rangle \mid x \neq y\} \ (= \texttt{not-equal}) \\
t(\texttt{not-equal}) & = & \mathcal{N} \times \mathcal{N}
\end{array}$$

例 71 インターネットの Web page 全体の集合を W とし，その上の二項関係 R として，以下のものを採用する．

$$aRb \ \Leftrightarrow \ \text{a から b へのリンクが 1 本以上ある．}$$

R に関する推移閉包 $r(R)$ を考えると，$\langle a, b\rangle \in r(R)$ は「a からリンクをたどって b に到達可能である」ことを表す．

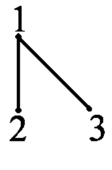
3.7 グラフ

グラフ (graph) は，いくつかの頂点 (点，vertex) と，それらを結ぶいくつかの辺 (edge) から構成された図形である．

グラフは，無向グラフ (辺が向きを持たないグラフ) と有向グラフ (辺が向きを持つグラフ) に大別される．

3.7.1 無向グラフ

無向グラフでは，各々の辺は 2 つの頂点を結びつける．たとえば，以下のグラフは 3 点 1,2,3 と，それらを結ぶ 2 本の辺からなる無向グラフである．



無向グラフ G は，頂点の集合 V と辺の集合 E の対として表現できる．[4] たとえば，上のグラフは以下の G として表現される．

$$\begin{array}{l}
G = \langle V, E\rangle \\
V = \{1, 2, 3\} \\
E = \{\{1, 2\}, \{1, 3\}\}
\end{array}$$

例 72 町と道路を想像せよ．町が頂点で道路が辺である．

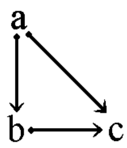
無向グラフにおいて，頂点 a の次数 (degree) とは，a と他の頂点を結ぶ辺の本数のことである．上の例では，頂点 1 の次数は 2, 頂点 2 および頂点 3 の次数は 1 である．

[4] この場合，同じ 2 点を結ぶ辺が複数あることはないと仮定している．用途によっては，同じ 2 点の間を結ぶ辺が 2 本以上あるグラフを考えることもある．

3.7.2　有向グラフ

辺に向きがついているグラフを，有向グラフ (directed graph) という．

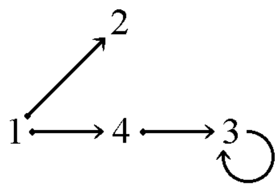
例 **73**　3 つの頂点 a,b,c からなる有向グラフの例．



このグラフの場合，a から b には辺があるが，b から a への辺はない．a から b への辺に対して，頂点 a を始点といい，頂点 b を終点という．

有向グラフ G も，頂点の集合と辺の集合の対として表現できる．ただし，辺の表現において，始点と終点は区別しなければいけないので，始点と終点からなる集合ではなく，始点と終点の対 (つい) として表現する．

例 **74**　有向グラフとその表現．



$$
\begin{aligned}
G &= \langle V, E\rangle \\
V &= \{1,2,3,4\} \\
E &= \{\langle 1,2\rangle, \langle 1,4\rangle, \langle 4,3\rangle, \langle 3,3\rangle\}
\end{aligned}
$$

E は $V \times V$ の部分集合であるので，V 上の二項関係である [5]．逆に，集合とその上の二項関係が与えられれば，それに対応する有向グラフが決まる．

有向グラフにおける頂点の次数は,「入次数」と「出次数」の 2 通りある．頂点 a の入次数とは，a を終点とする (a にはいってくる) 辺の本数であり，出次数とは，a を始点とする (a から出ていく) 辺の本数である．

以下では，無向グラフと有向グラフを総称して「グラフ」という．

3.7.3　位数とサイズ

グラフ $G = \langle V, E\rangle$ に対して，頂点の数 (集合 V の要素数) を G の位数といい，辺の数 (集合 E の要素数) を G のサイズという．位数やサイズが無限大であるグラフを扱うこともある．

3.7.4　道 (パス) と閉路 (サイクル)

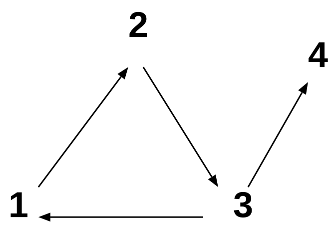
[2013/12/01 修正] グラフ G において，すべての $0 \le i < n$ に対して $\langle v_i, v_{i+1}\rangle$ が G の辺であるとき，G の頂点の列 $\langle v_0, v_1, v_2 \cdots, v_n\rangle$ を,「頂点 v_0 から頂点 v_n への道 (パス, path)」と言う．v_0 をこの道の始点，v_n を終点，n をこの道の長さという．道の定義は $n = 0$ の場合を許しており，

[5] この場合，同じ 2 点を結ぶ同一の方向の辺が複数あることはないと仮定している．

これは一点 v_0 だけからなる長さ 0 の道 $\langle v_0\rangle$ である．また， 長さ 1 以上の道 $\langle v_0, v_1, v_2 \cdots, v_n\rangle$ を，辺の列 $\langle v_0, v_1\rangle, \langle v_1, v_2\rangle, \cdots, \langle v_{n-1}, v_n\rangle$ で表すこともある．

修正個所についての補足; 2013/11/30 までの道の定義では，辺の列として定義していたため，「長さ 0 の道」が許されていなかった．変更後は頂点の列として定義し直したため，長さ 0 の道も含まれるようになった．なお，長さ 1 以上の道については，従来の定義と同一である．[2013/12/01 修正; 終わり]

例 **75** 次のグラフについて考える．



$\langle 1, 2, 3, 4\rangle$ は頂点 1 から頂点 4 への長さ 3 の道である．

一方，頂点 4 から頂点 1 への道は存在しない．

同じ辺が 2 回以上現れない道を単純道という．なお，グラフ理論の専門書によっては，ここで定義した「道」を「歩道」と呼び,「単純道」を「道」と呼ぶものもある．そのほか，細かな定義・用語が違うことがあるので注意されたい．

閉路 (サイクル, cycle) とは，始点と終点が同じ道のことである．先ほどのグラフでは，$\langle 1, 2, 3, 1\rangle$ という道があり，これは閉路になっている．

閉路を持たない有向グラフを非循環グラフ (directed acyclic graph, 略して dag) という．dag は応用上重要なグラフである．

3.7.5 グラフの同型

グラフ $G_1 = \langle V_1, E_1\rangle$ と $G_2 = \langle V_2, E_2\rangle$ が同型 (isomorphic) であるとは，頂点の名前を除いて 2 つのグラフが一致することである．

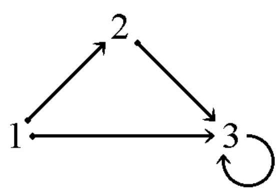
グラフの同型は,「関数」の言葉を使えば，以下のように厳密に定義できる (詳細は，第 2.7 章を参照のこと)．すなわち，関数 $f : V_1 \to V_2$ が存在して，以下の 2 つの条件が成立することである．

- f は全単射．
- $\forall x \in V_1.\forall y \in V_1.((\langle x, y\rangle$ が E_1の辺$) \Leftrightarrow (\langle f(x), f(y)\rangle$ が E_2の辺$))$

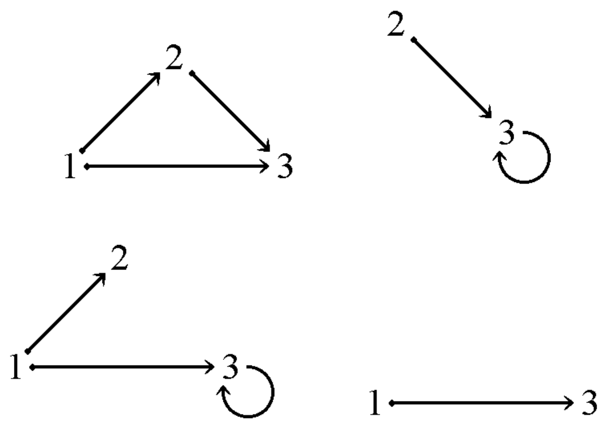
3.7.6 部分グラフ

グラフ $G = \langle V, E\rangle$ の部分グラフとは，$G' = \langle V', E'\rangle$ で，$V' \subset V$ かつ $E' \subset E$ となるグラフである．

例 **76** 次のグラフの部分グラフを考える．



このグラフの部分グラフの例としては下のようなものがある．(この他にもある．)



3.7.7　連結，連結成分 (†)

グラフが連結 (connected) であるとは，どの 2 つの頂点 a,b を取っても，a から b への道が存在することである．

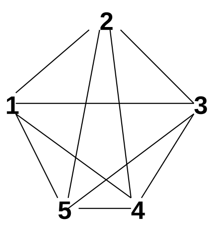
グラフ G の連結成分とは，G の極大な連結部分グラフのことである．すなわち，G の連結部分グラフ G' であって，G' を真に含む G の部分グラフで，連結なものが存在しないもののことである．

3.7.8　完全グラフ，完全 2 部グラフ (†)

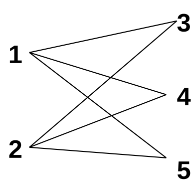
無向グラフで，全ての頂点の間に辺があるグラフを完全グラフという．頂点数が n の完全グラフを K_n と書く．たとえば，K_5 は，頂点が 5 個, 辺が 10 本あるグラフである．

無向グラフにおいて，頂点の集合が A と B に分割され，A に属する全ての頂点と B に属する全ての頂点の間に辺があり，それ以外には辺がないグラフを完全 2 部グラフという．A の頂点の数が n 個で，B の頂点の数が m 個であるような完全 2 部グラフを $K_{n,m}$ と書く．これは，頂点が $n + m$ 個で，辺が nm 本あるグラフである．

例 77



完全グラフ



完全2部グラフ

3.7.9　グラフの応用 (†)

グラフは，コンピュータ科学で頻繁に使われる構造であり，グラフに対する効率のよいアルゴリズムの開発は重要な課題である．ここでは，無向グラフの一筆書きの例をあげる．

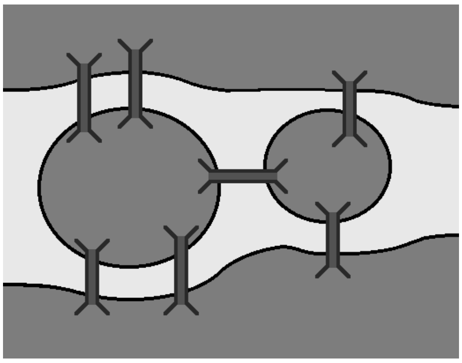
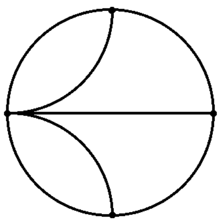


図 3.1: Königsberg の橋

例 78 [Königsberg の橋 (図 3.1)] 7 つの橋全部を一度ずつ渡って町の見物ができるか．Euler は，このような道が存在しないことを示した．この研究がグラフ理論誕生の契機である．

Königsberg の橋の問題は，グラフの一筆書き問題の一例である．与えられたグラフに対して，そのグラフの全ての辺をちょうど 1 回だけ含む道があるとき，そのグラフを一筆書き可能という．

連結な無向グラフが一筆書き可能かどうかは，次数が奇数となる頂点の個数を調べることでわかり，そのような頂点が 2 個以下であるときに一筆書き可能，そうでないとき一筆書き不能である．Königsberg の橋をグラフとして表現すると，次のようになる．



次数が奇数の頂点が 4 個あるため一筆書き不能である．

3.8 木

木 (tree) はいろいろな定式化が可能である．ここでは無向グラフの一種としての定義を与える．

無向グラフで，以下の条件を満たすものを木という．

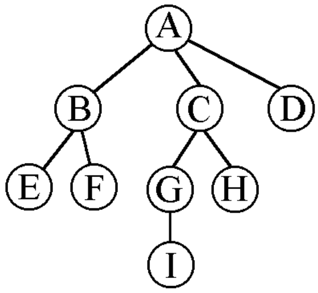
- 根 (root) と呼ばれる特別な頂点が 1 つだけある．
- 全ての頂点から根に至る単純道が唯一つ存在する．

この定義は以下の定義と同値である．

- 根 (root) と呼ばれる特別な頂点が 1 つだけある．
- 連結で閉路がない．

木を図形として表現するときは，通常，根を一番上に書く．

例 79 A を根とする木



木の定義より，根以外の頂点に対して，根に至る単純道があり，その長さは 1 以上である．この道の長さを，その頂点の深さ (depth) という．この道の上で，頂点 v の次の頂点を v の親という．根以外の頂点に対してその親は一意的に定まる．上の例では E の深さは 2 で親は B, B の深さは 1 で親は A, A の深さは 0 で親は存在しない．

頂点 A が 頂点 B の親であるとき，B は A の子という．子のない頂点を葉 (leaf) という．上の例では，E, F, I, H, D が葉である．葉以外の頂点のことを節 (せつ，node)，あるいは，節点という．

また，ある頂点に対して，それと根を結ぶ道の上にある頂点をその頂点の祖先という．つまり，祖先とは，親，親の親，親の親の親，等の総称である．(『関係』の章で学習した言葉を使えば,「親である」という関係の逆関係が「子である」という関係であり,「親である」という関係の推移的閉包が「祖先である」という関係である．) 同様に子孫も定義される．同じ親をもつ頂点同士を兄弟 (あるいは姉妹) という．

木の高さ (height) とは，木に含まれる頂点の深さの最大値である．上の例の木では，その高さは 3 である．

木の中の全ての節点において，その子が n 個以下である場合，n 分木 (エヌぶんぎ) という．上の例は，各節点の子の数は 3 以下であるので，3 分木という．

木をいくつか集めたグラフを森 (forest) という．すなわち，連結成分が全て木になっているグラフを森という．

3.8.1 順序木

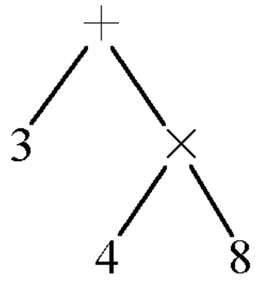
前節の木の定義では，1 つの節点に対する子の間に順序はなく，左右をひっくり返した木も同じであった．(左右をひっくり返したグラフも，グラフとしては同型である．)

しかし，コンピュータ科学における多くの応用では，子の間に左右の順序関係があった方が便利である．すなわち，先の例では，A の子である B, C, D の間には,「B が一番左の子であり，C が次に左の子であり，D が一番右の子である」というように順序付けられている方が都合がよい．この場合，C, B, D の順番に並べられている木とは異なる木であると考える．

このように，木において左右の順序関係を区別する場合を順序木という．コンピュータ科学では，ほとんどの場合，順序木を使うので，順序木のことを単に木と呼ぶことが多い．

順序木の応用として，コンピュータによる記号処理を考える．記号表現は，論理式や数式など，記号の列で表現されたものであり，木と見なすことができる．ここでは，数式が木で表わせることを見る．

例 80 $3 + (4 \times 8)$ に対応する木.



木として表現することにより，$3 + (4 \times 8)$ と $(3 + 4) \times 8$ が異なる構造を持つことが明確になる．また，$3 + (4 \times 8)$ と $3 + (8 \times 4)$ は異なる表現として扱いたいので，単純な木ではなく，順序木として表現するのが普通である．

記号表現以外にも，コンピュータ科学では木の概念を使うことが多い．たとえば，1 つのコンピュータの中のファイルシステム (ファイルとディレクトリ全体) は木の構造を持つ．また，インターネットのドメイン名 (メールアドレスの @より後の部分) の全体も一種の木を構成している [6].

3.8.2 走査

木の走査 (traversal) とは，すべての頂点をある順番にしたがって 1 回ずつ処理することである．この節では，順序 2 分木の走査について説明する．

木の走査において，頂点を順番にたどる系統的な方法は複数考えられる．特によく使われる方法は，幅優先 (breadth-first) で頂点をたどるものと，深さ優先 (depth-first) でたどるものである．

幅優先の走査では，根に近い順に左から右に処理を行う．たとえば，例 80 の数式を表す木を幅優先で走査すると，$+, 3, \times, 4, 8$ の順に処理される．

一方，深さ優先の走査では，左部分木を深さ優先で走査した後，右部分木を同様に走査する．根の処理を行うタイミングとしては，左部分木を走査する直前，右部分木を走査する直前，右部分木を走査した直後の 3 つが考えられるので，同じ深さ優先の走査法の中でも，それぞれに対応した以下の 3 種類がある．

- 行きがけ順 (pre-order)
  1. 根を処理する
  2. 左部分木を行きがけ順で走査する
  3. 右部分木を行きがけ順で走査する
- 通りがかり順 (in-order)
  1. 左部分木を通りがかり順で走査する
  2. 根を処理する
  3. 右部分木を通りがかり順で走査する
- 帰りがけ順 (post-order)
  1. 左部分木を帰りがけ順で走査する
  2. 右部分木を帰りがけ順で走査する

[6] トップレベルのドメインが 1 つではないので，木ではなく森である，という見方もできる．

3. 根を処理する

たとえば，例 80 の木を 3 つの方法で走査すると，頂点が処理される順番はそれぞれ以下のようになる．

- 行きがけ順：$+, 3, \times, 4, 8$

  頂点を行きがけ順に表示すると「$+\ 3 \times 4\ 8$」が出力される．このように，$+$ や $\times$ といった演算子を，演算対象の前に置いて数式を記述する方法を前置記法 (ポーランド記法) という．前置記法は LISP というプログラミング言語で採用されている．

- 通りがかり順：$3, +, 4, \times, 8$

  頂点を通りがかり順に (括弧を補いながら) 表示すると「$(3 + (4 \times 8))$」が出力される．このように，演算子を 2 つの演算対象の間に置いて数式を記述する方法を中置記法という．中置記法は算数・数学の教育や，ほとんどの電卓・プログラミング言語で採用されている最も標準的な記法である．

- 帰りがけ順：$3, 4, 8, \times, +$

  頂点を帰りがけ順に表示すると「$3\ 4\ 8 \times +$」が出力される．このように，演算子を演算対象の後に置いて数式を記述する方法を後置記法 (逆ポーランド記法) という．後置記法は一部の電卓に加え，Forth や PostScript といったプログラミング言語で採用されている．