

ネットワークの代数的研究

神戸大学大学院理学研究科

JSTさきがけ

春名 太一

ネットワークの代数的研究

Step 1. ネットワークの圏論による記述

ネットワーク=有向グラフ

$$G = (A, O, \partial_0, \partial_1)$$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる函数

$\partial_1: A \rightarrow O$: 終点をとる函数

有向グラフを有限圏 C から集合圏 $Sets$ への関手(前層)と考える。

$$C = \begin{array}{c} \bullet^2 \\ \uparrow m_0 \quad \uparrow m_1 \\ \bullet^1 \end{array}$$

$$G: C^{op} \rightarrow Sets$$

$$G(2) = A, G(1) = O,$$

$$G(m_0) = \partial_0, G(m_1) = \partial_1$$

C^{op} は C の射の向きを逆にした圏

圏(category)のデータ

- (1) 対象(object): A, B, C, Λ
- (2) 射(morphism): f, g, h, Λ
- (3) 関数 dom, cod : 射を対象に対応させる
- (4) 射の合成 $\circ$: $\text{dom}(g) = \text{cod}(f)$ のとき, 射 $g \circ f$ が定義される

$\text{dom}(f) = A, \text{cod}(f) = B$ のとき
 $f: A \rightarrow B$ または $A \xrightarrow{f} B$ と書く。

$$\begin{array}{ccccc} & f & & g & \\ & \searrow & & \searrow & \\ A & \rightarrow & B & \rightarrow & C \\ & \nearrow & & \nearrow & \\ & g \circ f & & & \end{array}$$

$$\text{Hom}(A, B) = \{f \mid f: A \rightarrow B\}$$

圏の公理

- (a) $\text{dom}(g \circ f) = \text{dom}(f), \text{cod}(g \circ f) = \text{cod}(g)$
- (b) $h \circ (g \circ f) = (h \circ g) \circ f$
- (c) すべての対象 A に対して、射 $\text{id}_A: A \rightarrow A$ で、
すべての $\text{dom}(f) = A$ なる射 f に対して $f = f \circ \text{id}_A$,
すべての $\text{cod}(g) = A$ なる射 g に対して $g = \text{id}_A \circ g$
となるものが存在する(恒等射)。

$$\begin{array}{ccccc} C & \xrightarrow{g} & A & \xrightarrow{f} & B \\ & \searrow g & \downarrow \text{id}_A & \nearrow f & \\ & & A & & \end{array}$$

対象 A に対する恒等射 id_A は唯一つ。

圏の例

(1) 集合圏 *Sets*

対象：集合

射：写像

(2) 有向グラフの圏 *Grph*

対象：有向グラフ

射：有向グラフの準同型写像

構造を保存する

有向グラフ $G = (A, O, \partial_0, \partial_1)$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる関数

$\partial_1: A \rightarrow O$: 終点をとる関数

有向グラフの準同型写像

$$\mu = (\mu_A, \mu_O): G \rightarrow G' \quad \begin{array}{c} \mu_A \\ A \rightarrow A' \end{array}$$

$$\partial_i' \mu_A = \mu_O \partial_i \quad (i = 0, 1) \quad \begin{array}{ccc} \partial_i \downarrow & & \downarrow \partial_i' \\ O & \rightarrow & O' \end{array}$$

$$G' = (A', O', \partial_0', \partial_1')$$

$$\mu_O$$

Step 1. ネットワークの圏論による記述

ネットワーク=有向グラフ

$$G = (A, O, \partial_0, \partial_1)$$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる函数

$\partial_1: A \rightarrow O$: 終点をとる函数

有向グラフを有限圏 C から集合圏 $Sets$ への関手(前層)と考える。

$$C = \begin{array}{c} \bullet^2 \\ \uparrow m_0 \quad \uparrow m_1 \\ \bullet^1 \end{array}$$

$$G: C^{op} \rightarrow Sets$$

$$G(2) = A, G(1) = O,$$

$$G(m_0) = \partial_0, G(m_1) = \partial_1$$

C^{op} は C の射の向きを逆にした圏

圏の例

(1) 集合圏 *Sets*

対象：集合

射：写像

(2) 有向グラフの圏 *Grph*

対象：有向グラフ

射：有向グラフの準同型写像

構造を保存する

有向グラフ $G = (A, O, \partial_0, \partial_1)$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる関数

$\partial_1: A \rightarrow O$: 終点をとる関数

有向グラフの準同型写像

$$\mu = (\mu_A, \mu_O): G \rightarrow G' \quad \begin{array}{c} \mu_A \\ A \rightarrow A' \end{array}$$

$$\partial_i' \mu_A = \mu_O \partial_i \quad (i = 0, 1) \quad \begin{array}{ccc} \partial_i \downarrow & & \downarrow \partial_i' \\ O & \rightarrow & O' \end{array}$$

$$G' = (A', O', \partial_0', \partial_1')$$

$$\mu_O$$

Step 1. ネットワークの圏論による記述

ネットワーク=有向グラフ

$$G = (A, O, \partial_0, \partial_1)$$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる函数

$\partial_1: A \rightarrow O$: 終点をとる函数

有向グラフを有限圏 C から集合圏 $Sets$ への関手(前層)と考える。

$$C = \begin{array}{c} \bullet^2 \\ \uparrow m_0 \quad \uparrow m_1 \\ \bullet_1 \end{array}$$

$$G: C^{op} \rightarrow Sets$$

$$G(2) = A, G(1) = O,$$

$$G(m_0) = \partial_0, G(m_1) = \partial_1$$

C^{op} は C の射の向きを逆にした圏

有向グラフの圏=C上の前層の圏

C上の前層(presheaf)の圏 : $Sets^{C^{op}}$

対象 : 関手 $G : C^{op} \rightarrow Sets$

$$G = (A, O, \partial_0, \partial_1)$$

射 : 自然変換 : $\mu : G \rightarrow G'$

$$G' = (A', O', \partial'_0, \partial'_1)$$

$$\begin{array}{ccc}
 & & \mu_2 \\
 2 & & G(2) = A \rightarrow G'(2) = A' \\
 \downarrow m_i & G(m_i) = \partial_i \downarrow & \downarrow G'(m_i) = \partial'_i \\
 1 \quad \text{in } C^{op} & G(1) = O \xrightarrow{\mu_1} G'(1) = O' & \text{in } Sets
 \end{array}$$

自然変換はちょうど有向グラフの準同型となる。

Step 1. ネットワークの圏論による記述

ネットワーク=有向グラフ

$$G = (A, O, \partial_0, \partial_1)$$

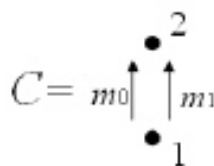
A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる函数

$\partial_1: A \rightarrow O$: 終点をとる函数

有向グラフを有限圏 C から集合圏 $Sets$ への関手(前層)と考える。



$$G: C^{op} \rightarrow Sets$$

$$G(2) = A, G(1) = O,$$

$$G(m_0) = \partial_0, G(m_1) = \partial_1$$

C^{op} は C の射の向きを逆にした圏

有向グラフの圏=C上の前層の圏

C上の前層(presheaf)の圏 : $Sets^{C^{op}}$

対象 : 関手 $G : C^{op} \rightarrow Sets$

$$G = (A, O, \partial_0, \partial_1)$$

射 : 自然変換 : $\mu : G \rightarrow G'$

$$G' = (A', O', \partial'_0, \partial'_1)$$

$$\begin{array}{ccc}
 & & \mu_2 \\
 2 & & G(2) = A \rightarrow G'(2) = A' \\
 \downarrow m_i & G(m_i) = \partial_i \downarrow & \downarrow G'(m_i) = \partial'_i \\
 1 \quad \text{in } C^{op} & G(1) = O \xrightarrow{\mu_1} G'(1) = O' & \text{in } Sets
 \end{array}$$

自然変換はちょうど有向グラフの準同型となる。

圏の例

(1) 集合圏 *Sets*

対象：集合

射：写像

(2) 有向グラフの圏 *Grph*

対象：有向グラフ

射：有向グラフの準同型写像

構造を保存する

有向グラフ $G = (A, O, \partial_0, \partial_1)$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる関数

$\partial_1: A \rightarrow O$: 終点をとる関数

有向グラフの準同型写像

$$\mu = (\mu_A, \mu_O): G \rightarrow G' \quad \begin{array}{c} \mu_A \\ A \rightarrow A' \end{array}$$

$$\partial_i' \mu_A = \mu_O \partial_i \quad (i = 0, 1) \quad \begin{array}{ccc} \partial_i \downarrow & & \downarrow \partial_i' \\ O & \rightarrow & O' \end{array}$$

$$G' = (A', O', \partial_0', \partial_1')$$

$$\mu_O$$

有向グラフの圏=C上の前層の圏

C上の前層(presheaf)の圏 : $Sets^{C^{op}}$

対象 : 関手 $G : C^{op} \rightarrow Sets$

$$G = (A, O, \partial_0, \partial_1)$$

射 : 自然変換 : $\mu : G \rightarrow G'$

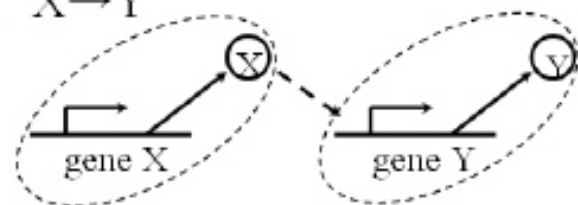
$$G' = (A', O', \partial'_0, \partial'_1)$$

$$\begin{array}{ccc}
 & & \mu_2 \\
 2 & & G(2) = A \rightarrow G'(2) = A' \\
 \downarrow m_i & G(m_i) = \partial_i \downarrow & \downarrow G'(m_i) = \partial'_i \\
 1 \quad \text{in } C^{op} & G(1) = O \xrightarrow{\mu_1} G'(1) = O' & \text{in } Sets
 \end{array}$$

自然変換はちょうど有向グラフの準同型となる。

Step 2. 機能の表現: 情報処理パターン

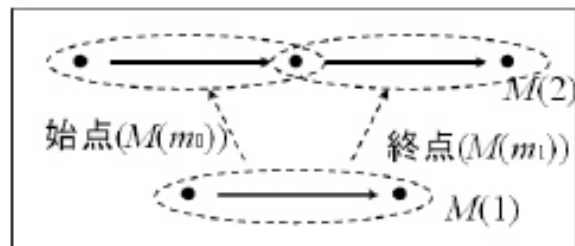
例. $X \rightarrow Y$



情報処理パターン

$$M : C \rightarrow Grph = Sets^{C^{op}}$$

C から有向グラフの圏 $Grph$ への関手



$M(1)$: 頂点の機能を有向グラフで表す

$M(2)$: 頂点の機能間の関わり方を有向グラフで表す

有向グラフの圏=C上の前層の圏

C上の前層(presheaf)の圏 : $\text{Sets}^{C^{op}}$

対象 : 関手 $G : C^{op} \rightarrow \text{Sets}$

$$G = (A, O, \partial_0, \partial_1)$$

射 : 自然変換 : $\mu : G \rightarrow G'$

$$G' = (A', O', \partial'_0, \partial'_1)$$

$$\begin{array}{ccc}
 & & \mu_2 \\
 2 & & G(2) = A \rightarrow G'(2) = A' \\
 \downarrow m_i & G(m_i) = \partial_i \downarrow & \downarrow G'(m_i) = \partial'_i \\
 1 \quad \text{in } C^{op} & G(1) = O \xrightarrow{\mu_1} G'(1) = O' & \text{in } \text{Sets}
 \end{array}$$

自然変換はちょうど有向グラフの準同型となる。

Step 1. ネットワークの圏論による記述

ネットワーク=有向グラフ

$$G = (A, O, \partial_0, \partial_1)$$

A : 矢印の集合

O : 頂点の集合

$\partial_0: A \rightarrow O$: 始点をとる函数

$\partial_1: A \rightarrow O$: 終点をとる函数

有向グラフを有限圏 C から集合圏 $Sets$ への関手(前層)と考える。

$$C = \begin{array}{c} \bullet^2 \\ \uparrow m_0 \quad \uparrow m_1 \\ \bullet_1 \end{array}$$

$$G: C^{op} \rightarrow Sets$$

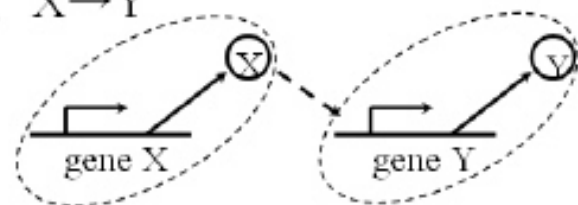
$$G(2) = A, G(1) = O,$$

$$G(m_0) = \partial_0, G(m_1) = \partial_1$$

C^{op} は C の射の向きを逆にした圏

Step 2. 機能の表現: 情報処理パターン

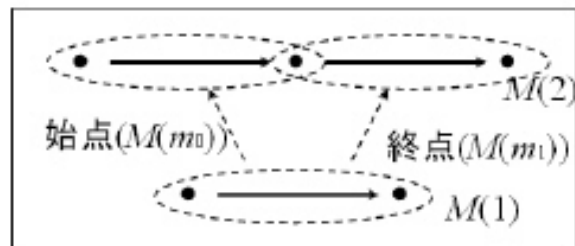
例. $X \rightarrow Y$



情報処理パターン

$$M : C \rightarrow Grph = Sets^{C^{op}}$$

C から有向グラフの圏 $Grph$ への関手



$M(1)$: 頂点の機能を有向グラフで表す

$M(2)$: 頂点の機能間の関わり方を有向グラフで表す

Step 3. 情報処理パターンが誘導する、「機能を忘れる」関手

情報処理パターン $M : C \rightarrow Grph$

G : 有向グラフ

関手 $R : Grph \rightarrow Sets^{C^{op}} = Grph$ を以下で定義。

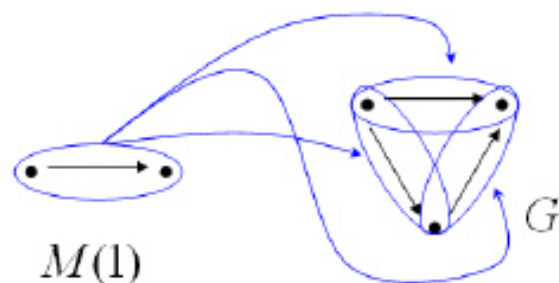
$$G \propto \text{Hom}(M(-), G)$$

$R(G) = \text{Hom}(M(-), G) = (A, O, \partial_0, \partial_1)$ とすると、

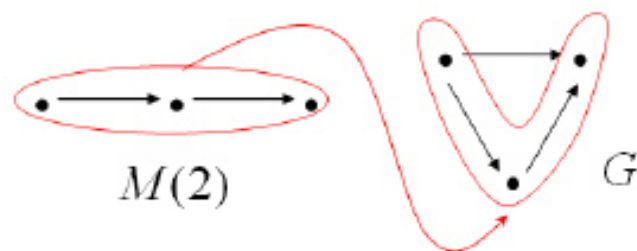
$$A = \text{Hom}(M(2), G), O = \text{Hom}(M(1), G)$$

$f \in \text{Hom}(M(2), G)$ に対して、 $\partial_i(f) = f \circ M(m_i) \quad (i = 0, 1)$

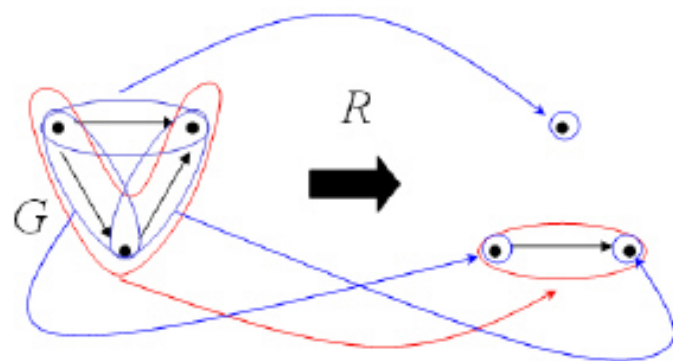
$$M(1) \xrightarrow{M(m_1)} M(2) \xrightarrow{f} G$$



準同型 $M(1) \rightarrow G$ は3つ。
 $R(G)$ の頂点は3つ。

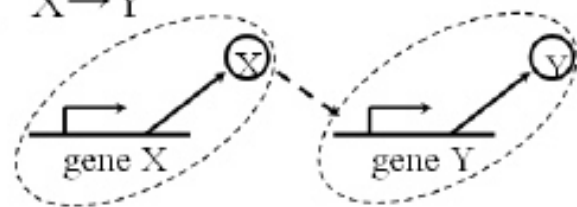


準同型 $M(2) \rightarrow G$ は1つ。
 $R(G)$ の矢印は1つ。



Step 2. 機能の表現: 情報処理パターン

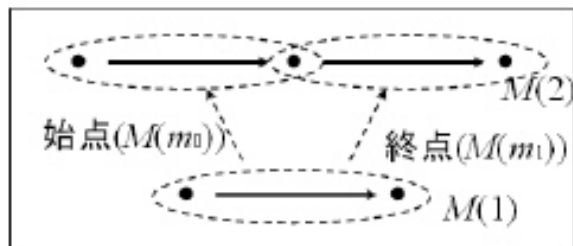
例. $X \rightarrow Y$



情報処理パターン

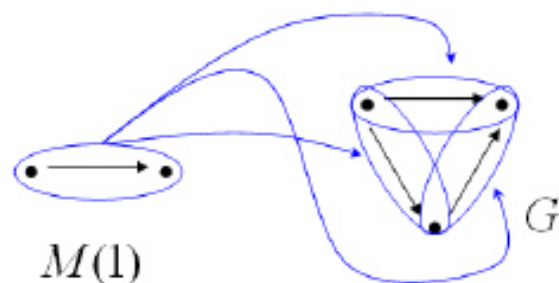
$$M : C \rightarrow Grph = Sets^{C^{op}}$$

C から有向グラフの圏 $Grph$ への関手

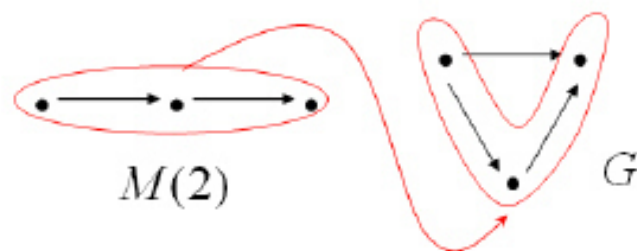


$M(1)$: 頂点の機能を有向グラフで表す

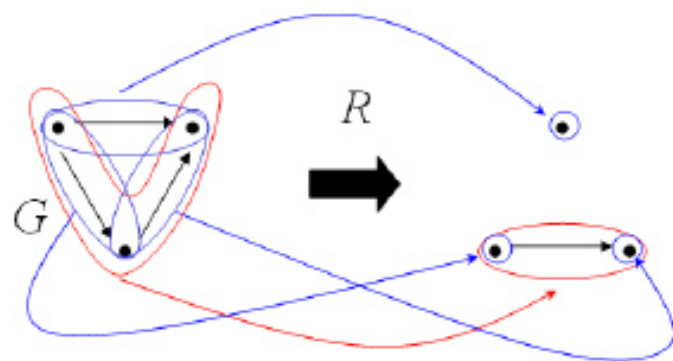
$M(2)$: 頂点の機能間の関わり方を有向グラフで表す



準同型 $M(1) \rightarrow G$ は3つ。
 $R(G)$ の頂点は3つ。



準同型 $M(2) \rightarrow G$ は1つ。
 $R(G)$ の矢印は1つ。



Step 4. 機能から構造へ: 安定条件

機能を忘れる関手 R は左随伴関手 L を持つ

$$\text{Hom}(L(F), G) \cong \text{Hom}(F, R(G))$$

機能を埋め込む

$$F \longmapsto F \otimes M$$

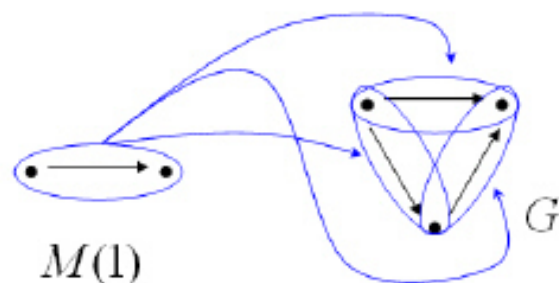
$$\text{Grph} = \text{Sets}^C \begin{array}{c} \xrightarrow{L} \\ \xleftarrow{R} \end{array} \text{Grph}$$

$$\text{Hom}(M(-), G) \longleftarrow G$$

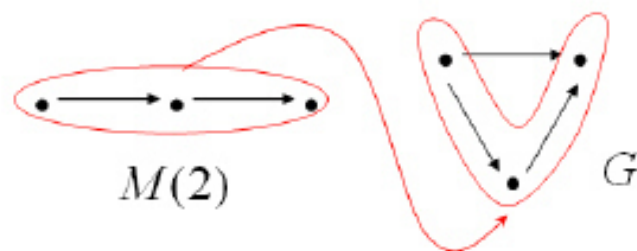
機能を忘れる

情報処理パターンの安定条件: $RL(F) = F$

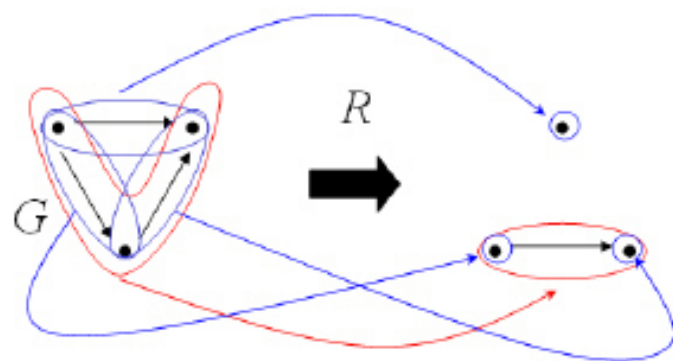
M が F に埋め込まれることによって新たに印加されるべき情報がすでに F の構造に全て反映されている



準同型 $M(1) \rightarrow G$ は3つ。
 $R(G)$ の頂点は3つ。



準同型 $M(2) \rightarrow G$ は1つ。
 $R(G)$ の矢印は1つ。



Step 4. 機能から構造へ: 安定条件

機能を忘れる関手 R は左随伴関手 L を持つ

$$\text{Hom}(L(F), G) \cong \text{Hom}(F, R(G))$$

機能を埋め込む

$$F \longmapsto F \otimes M$$

$$\text{Grph} = \text{Sets}^C \begin{array}{c} \xrightarrow{L} \\ \xleftarrow{R} \end{array} \text{Grph}$$

$$\text{Hom}(M(-), G) \longleftarrow G$$

機能を忘れる

情報処理パターンの安定条件: $RL(F) = F$

M が F に埋め込まれることによって新たに印加されるべき情報がすでに F の構造に全て反映されている

函手(functor)

$C, C': \text{圏}$

函手 $F: C \rightarrow C'$

$$\begin{array}{ccc} A & & F(A) \\ \downarrow f & & \downarrow F(f) \\ B \text{ in } C & & F(B) \text{ in } C' \end{array}$$

$$(1) \quad F(\text{id}_A) = \text{id}_{F(A)}$$

$$(2) \quad F(g \circ f) = F(g) \circ F(f)$$

自然变换(natural transformation)

C, C' : 圏

函手 $F, G: C \rightarrow C'$

自然变换 $\mu: F \rightarrow G$

$$\begin{array}{ccc} A & & F(A) \xrightarrow{\mu_A} G(A) \\ \downarrow f & & F(f) \downarrow \quad \quad \downarrow G(f) \\ B \text{ in } C & & F(B) \xrightarrow{\mu_B} G(B) \text{ in } C' \end{array}$$

$$G(f) \circ \mu_A = \mu_B \circ F(f)$$

随伴関手(adjoint functor)

C, C' : 圏 関手 $F: C \rightarrow C'$ $A: C$ の対象、 $B: C'$ の対象
 $G: C' \rightarrow C$

$\text{Hom}(F(A), B) \cong \text{Hom}(A, G(B))$
が成り立つとき、 F は G の左随伴関手、
 G は F の右随伴関手という。

例. X : 集合

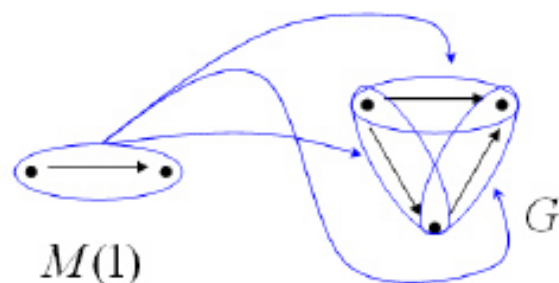
関手 $(-) \times X: \text{Sets} \rightarrow \text{Sets}: Y \mapsto Y \times X$

$(-)^X: \text{Sets} \rightarrow \text{Sets}: Z \mapsto Z^X$ Z^X は X から Z への関数の集合

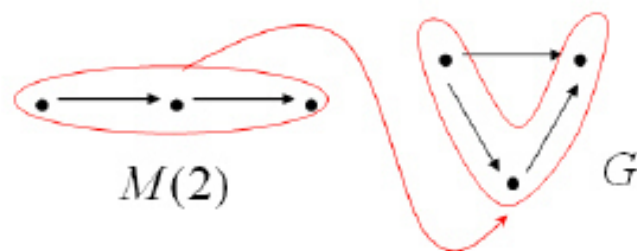
$$\text{Hom}(Y \times X, Z) \cong \text{Hom}(Y, Z^X)$$

二変数関数 $f: Y \times X \rightarrow Z$ と

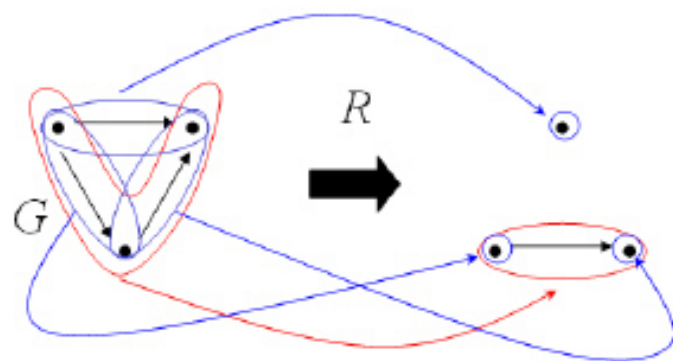
関数値関数 $\hat{f}: Y \rightarrow Z^X$ は一対一対応



準同型 $M(1) \rightarrow G$ は3つ。
 $R(G)$ の頂点は3つ。



準同型 $M(2) \rightarrow G$ は1つ。
 $R(G)$ の矢印は1つ。



Step 4. 機能から構造へ: 安定条件

機能を忘れる関手 R は左随伴関手 L を持つ

$$\text{Hom}(L(F), G) \cong \text{Hom}(F, R(G))$$

機能を埋め込む

$$F \longmapsto F \otimes M$$

$$\text{Grph} = \text{Sets}^C \begin{array}{c} \xrightarrow{L} \\ \xleftarrow{R} \end{array} \text{Grph}$$

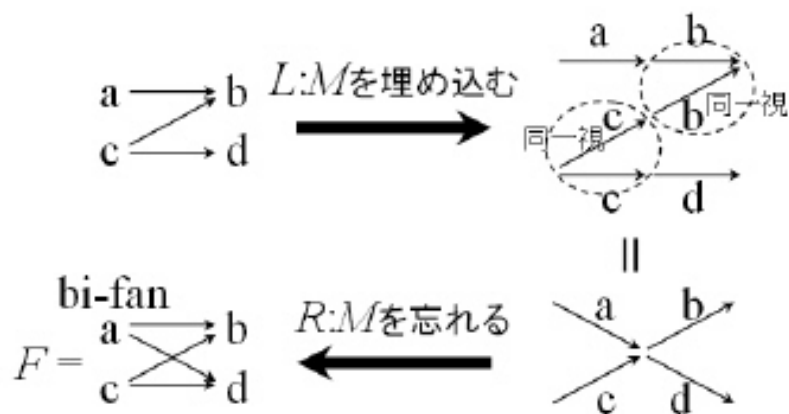
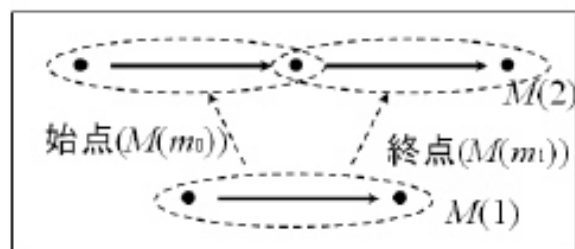
$$\text{Hom}(M(-), G) \longleftarrow G$$

機能を忘れる

情報処理パターンの安定条件: $RL(F) = F$

M が F に埋め込まれることによって新たに印加されるべき情報がすでに F の構造に全て反映されている

例.



$F = RL(F)$ iff $a \rightarrow b \leftarrow c \rightarrow d$ in F then $a \rightarrow d$ in F

Step 4. 機能から構造へ: 安定条件

機能を忘れる関手 R は左随伴関手 L を持つ

$$\text{Hom}(L(F), G) \cong \text{Hom}(F, R(G))$$

機能を埋め込む

$$F \longmapsto F \otimes M$$

$$\text{Grph} = \text{Sets}^C \begin{array}{c} \xrightarrow{L} \\ \xleftarrow{R} \end{array} \text{Grph}$$

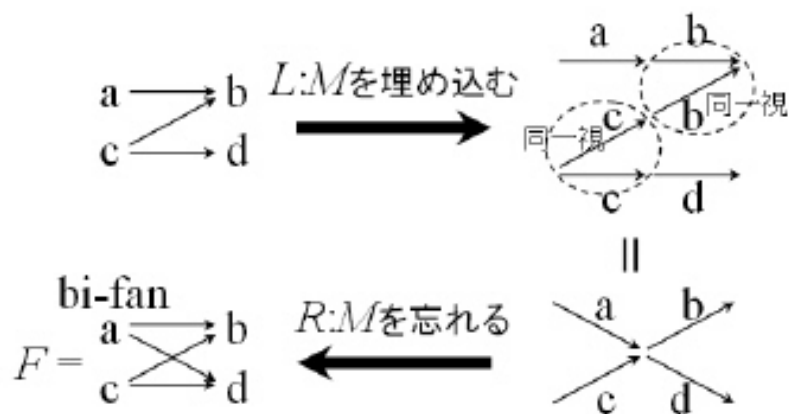
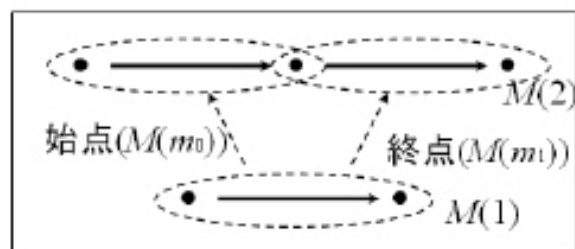
$$\text{Hom}(M(-), G) \longleftarrow G$$

機能を忘れる

情報処理パターンの安定条件: $RL(F) = F$

M が F に埋め込まれることによって新たに印加されるべき情報がすでに F の構造に全て反映されている

例.



$F = RL(F)$ iff $a \rightarrow b \leftarrow c \rightarrow d$ in F then $a \rightarrow d$ in F

Step 4. 機能から構造へ: 安定条件

機能を忘れる関手 R は左随伴関手 L を持つ

$$\text{Hom}(L(F), G) \cong \text{Hom}(F, R(G))$$

機能を埋め込む

$$F \longmapsto F \otimes M$$

$$\text{Grph} = \text{Sets}^C \begin{array}{c} \xrightarrow{L} \\ \xleftarrow{R} \end{array} \text{Grph}$$

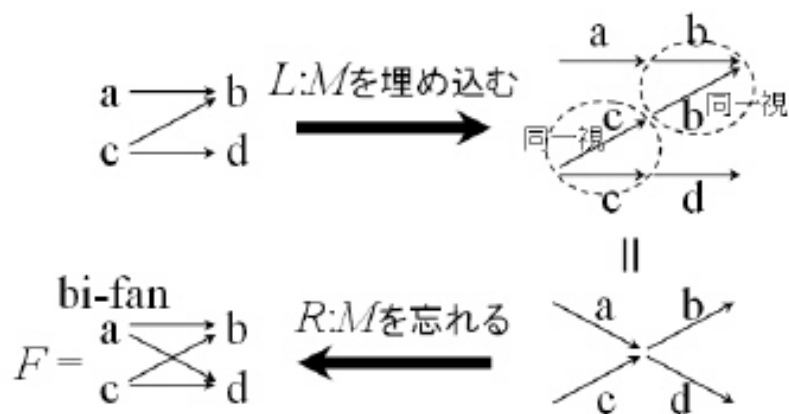
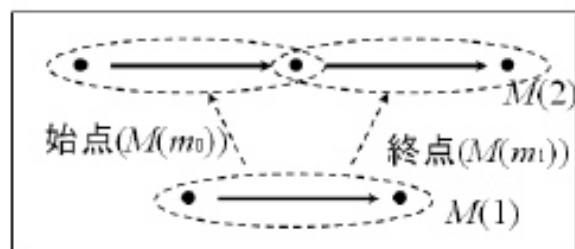
$$\text{Hom}(M(-), G) \longleftarrow G$$

機能を忘れる

情報処理パターンの安定条件: $RL(F) = F$

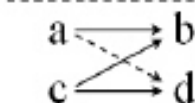
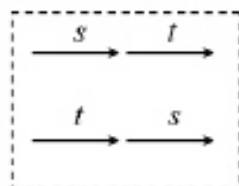
M が F に埋め込まれることによって新たに印加されるべき情報がすでに F の構造に全て反映されている

例.

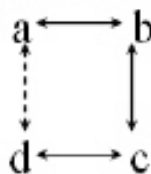
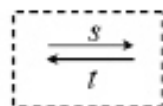


$F = RL(F)$ iff $a \rightarrow b \leftarrow c \rightarrow d$ in F then $a \rightarrow d$ in F

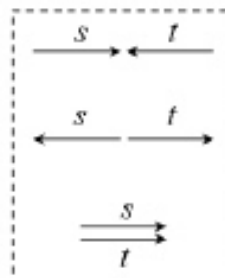
簡単な場合の計算例.



bi-fan



タンパク質間相互作用ネットワーク
のモチーフの一つ

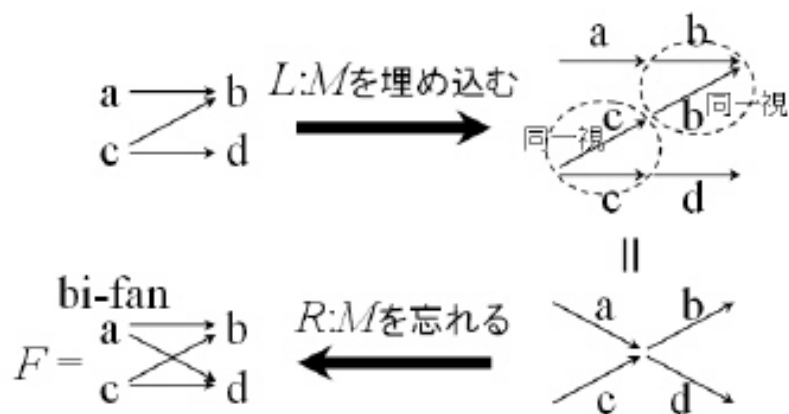
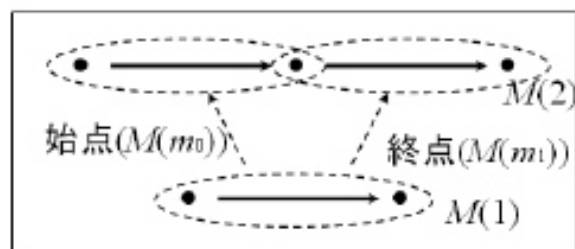


同値関係のグラフ

$$M(1) = \bullet \rightarrow \bullet$$

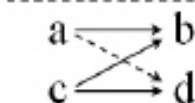
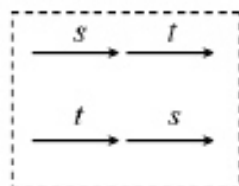
点線の中の一つ一つのグラフは $M(2)$ 。
 s は始点、 t は終点に対応。

例.

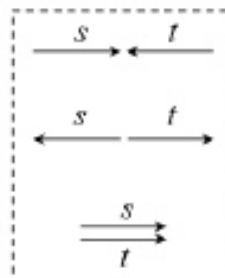
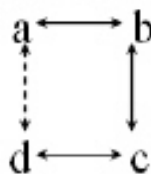
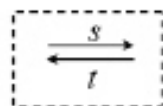


$F = RL(F)$ iff $a \rightarrow b \leftarrow c \rightarrow d$ in F then $a \rightarrow d$ in F

簡単な場合の計算例.



bi-fan



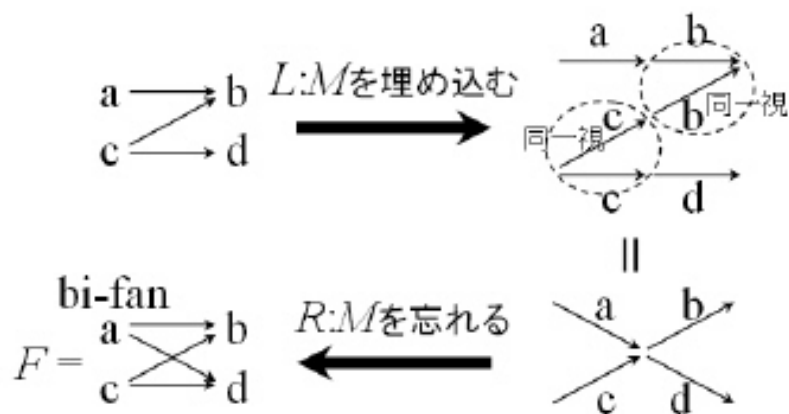
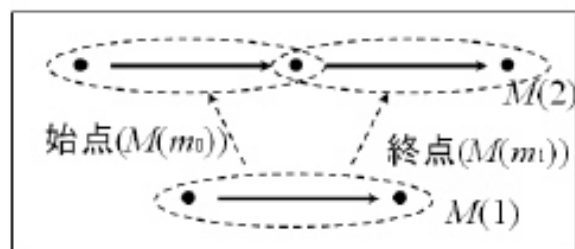
同値関係のグラフ

タンパク質間相互作用ネットワーク
のモチーフの一つ

$$M(1) = \bullet \rightarrow \bullet$$

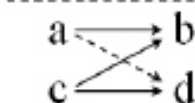
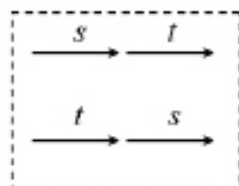
点線の中の一つ一つのグラフは $M(2)$ 。
 s は始点、 t は終点に対応。

例.

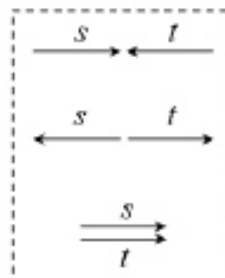
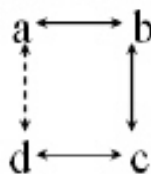
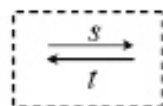


$F = RL(F)$ iff $a \rightarrow b \leftarrow c \rightarrow d$ in F then $a \rightarrow d$ in F

簡単な場合の計算例.



bi-fan



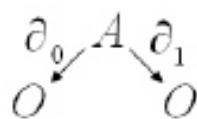
同値関係のグラフ

タンパク質間相互作用ネットワーク
のモチーフの一つ

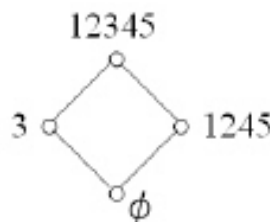
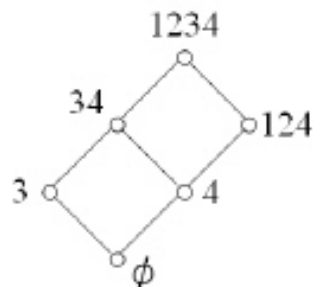
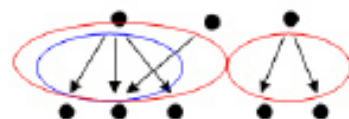
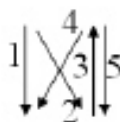
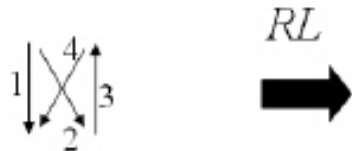
$$M(1) = \bullet \rightarrow \bullet$$

点線の中の一つ一つのグラフは $M(2)$ 。
 s は始点、 t は終点に対応。

論理との関係



矢印の集合上の同値関係 $R_i: fR_i g \Leftrightarrow \partial_i f = \partial_i g$
 $R_{0*}R_1^*(X) = X$ を満たす $X \subseteq A$ の全体は束をなす。



○ $R_{0*}R_1^*(X) = X$

○ $R_{0*}R_1^*(X) \neq X$

非ブール束(分配束)

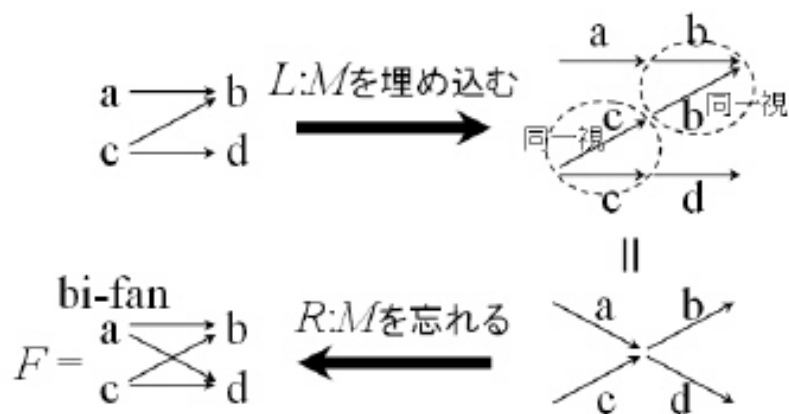
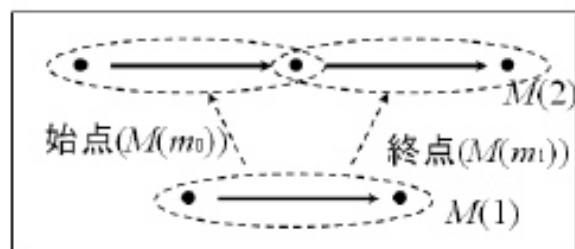
ブール束⇒モジュールの組合せ

実は、 $RL(F)=F$ を満たす有向グラフはみなブール束を誘導する。

まとめ

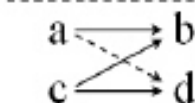
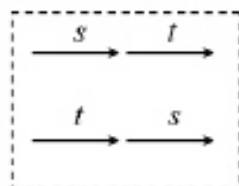
- ネットワークの局所構造の代数的見方
 - 機能と構造の双対性の下での安定化
- ここで行ったのは可能な構成の一つ
 - 有限圏上の組み合わせ構造には豊かな内容があるはず・・・「パターンの代数」

例.

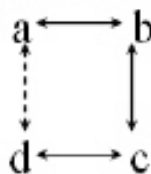
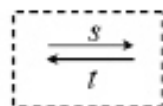


$F = RL(F)$ iff $a \rightarrow b \leftarrow c \rightarrow d$ in F then $a \rightarrow d$ in F

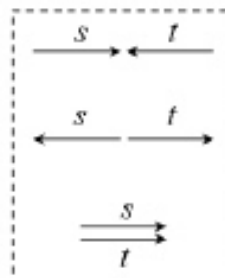
簡単な場合の計算例.



bi-fan



タンパク質間相互作用ネットワーク
のモチーフの一つ



同値関係のグラフ

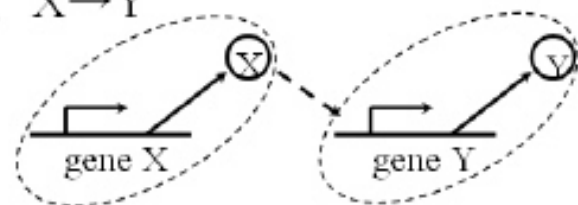
$$M(1) = \bullet \rightarrow \bullet$$

点線の中の一つ一つのグラフは $M(2)$ 。
 s は始点、 t は終点に対応。

- 先行研究
 - ネットワーク＝情報が流れる経路
 - 機能＝経路上のダイナミクス
 - 機能に対する自然選択
- 我々
 - (広い意味での)機能⇒局所構造
 - 自然選択ではない見方
 - 「普遍性の痕跡」(by I. Tsuda)を見つけ出したい
 - Bi-fanの普遍性

Step 2. 機能の表現: 情報処理パターン

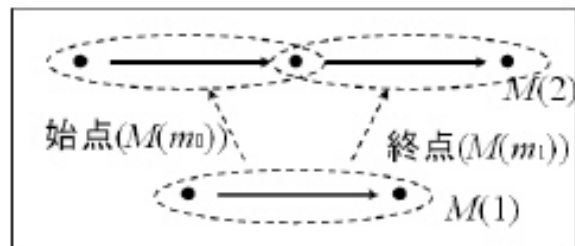
例. $X \rightarrow Y$



情報処理パターン

$$M : C \rightarrow Grph = Sets^{C^{op}}$$

C から有向グラフの圏 $Grph$ への関手



$M(1)$: 頂点の機能を有向グラフで表す

$M(2)$: 頂点の機能間の関わり方を有向グラフで表す