述語変換子意味論と余代数

こっとん (@CottonShampoo)

2025 年 12 月 20 日 土曜日

Mathematical Logic Advent Calendar 2025 の 20 日目の投稿です。
このドキュメントは，見やすく読み間違えにくいユニバーサルデザインフォントを採用しています。

目次

0　前置き

0.1　この記事が想定する対象読者

0.2　この記事の 4 つの目標

この記事では、以下の 4 つのことを目標とします。

1.
2.
3.
4.

0.3　この記事では扱わないこと

1　ホーア論理と述語変換子

ホーア論理（Hoare logic）と**述語変換子**（predicate transformer）は、プログラムの正しさ（意図した挙動をするか、バグがないかどうか）を数学的に検証するための、異なる２つのアプローチです。プログラムの満たしていてほしい性質（仕様）を論理式で記述し、それが満たされているかを論理学的手法を用いて確かめます。

たとえば、以下の例「入力 N に対し、階乗 $N!$ を計算して出力するプログラム」を考えてみましょう。

```
n := N
k := 1
while (n>0) {
  k := k*n;
  n := n-1;
}
```

【このプログラムについて知りたいこと】

1. 事後条件「$k = N!$」（出力 k が入力 N の階乗になる）を保証するには、何が分かればよいか？
2. 事後条件「$k = N!$」を成り立たせるためには、最低限、どんな事前条件を課せばよいか？

N に正の自然数を入力してこのプログラムを実行した後に、事後条件「$k = N!$」を満たしてほしいものとします。ループが何回実行されるかわからない中で、あらゆる入力 N に対してどうやって保証すればよいでしょうか。答えは、毎回のループを通過する前後で（各変数の値が変わっても）常に保存される性質（ループ不変量）「$k \cdot n! = N!$」に注目する、というアイデアです。そうすることで「階乗を出力する」という仕様が必ず満たされることを保証できます。つまり、事後条件を保証するためには、ループ不変量を探せばよいのです。これが**ホーア論理**の考え方です。

では今度は、この事後条件「$k = N!$」を成立させるためには事前条件として最低限どんな条件を課す必要があるか、という問題を考えてみましょう。答えは、「$N \geq 0$」です。これを求めるには、事後条件から出発してプログラムを後ろから前へたどり、各計算に応じて条件を置き換えていく、という操作をする必要があります。ただし、ループが何回実行されるかわからないため、この操作について不動点をとる必要があります。これが**述語変換子**の考え方です。

1.1　ホーア論理

ホーア論理では $\{P\}\,C\,\{Q\}$ という三つ組を使います。これは「事前条件 P から始めて、プログラム C を実行した結果、事後条件 Q が成り立つ」という関係です。

定義 1　ホーア論理の導出規則は以下のものからなる。

$$\frac{}{\{P\}\ \mathsf{skip}\ \{P\}}\ (\mathrm{skip})$$

$$\frac{}{\{P\,[a/x]\}\ x := a\ \{P\}}\ (\mathrm{assignment})$$

$$\frac{\{P\}\,C_1\,\{S\} \qquad \{S\}\,C_2\,\{Q\}}{\{P\}\,C_1; C_2\,\{Q\}}\ (\mathrm{sequential\ composition})$$

$$\frac{\{P \land \varphi\}\,C_1\,\{Q\} \qquad \{P \land \neg\varphi\}\,C_2\,\{Q\}}{\{P\}\ \mathsf{if}\ \varphi\ \mathsf{then}\ C_1\ \mathsf{else}\ C_2\ \{Q\}}\ (\mathrm{if})$$

$$\frac{\{P \land \varphi\}\,C\,\{P\}}{\{P\}\ \mathsf{while}\ \varphi\ \mathsf{do}\ C\ \{P \land \neg\varphi\}}\ (\mathrm{while})$$

$$\frac{P \Rightarrow P' \qquad \{P'\}\,C\,\{Q'\} \qquad Q' \Rightarrow Q}{\{P\}\,C\,\{Q\}}\ (\mathrm{consequence})$$

例 2　たとえば以下は、ホーア論理の証明図の例である。

$$
\dfrac{
\dfrac{\begin{array}{c}x{\geq}0\wedge x{>}0\\ \Rightarrow x{-}1{\geq}0\end{array} \qquad \dfrac{}{\{x{-}1{\geq}0\}\ x{:=}x{-}1\ \{x{\geq}0\}}\,(\mathrm{a})}{\{x{\geq}0\wedge x{>}0\}\ x{:=}x{-}1\ \{x{\geq}0\}}\,(\mathrm{c})
\qquad
\dfrac{\begin{array}{c}x{\geq}0\wedge\neg(x{>}0)\\ \Rightarrow x{\geq}0\end{array} \qquad \dfrac{}{\{x{\geq}0\}\ x{:=}x\ \{x{\geq}0\}}\,(\mathrm{a})}{\{x{\geq}0\wedge\neg(x{>}0)\}\ x{:=}x\ \{x{\geq}0\}}\,(\mathrm{c})
}{\{x{\geq}0\}\ \text{if } x{>}0 \text{ then } x{:=}x{-}1 \text{ else } x{:=}x\ \{x{\geq}0\}}\,(\mathrm{i})
$$

注意　ループ不変量とは、以下のような while 文の証明図に現れる I のこと。つまり、

1. ループの入口で事前条件によって含意され（$A \Rightarrow I$）、
2. 各回のループ実行時に常に保たれ（$\{I \wedge \varphi\}\ C\ \{I\}$）、
3. ループの出口で事後条件を含意する（$I \wedge \neg\varphi \Rightarrow B$）

ような論理式 I のことである。

$$
\dfrac{A \Rightarrow I \qquad \dfrac{\{I \wedge \varphi\}\ C\ \{I\}}{\{I\}\ \text{while } \varphi \text{ do } C\ \{I \wedge \neg\varphi\}}\,(\text{while}) \qquad I \wedge \neg\varphi \Rightarrow B}{\{A\}\ \text{while } \varphi \text{ do } C\ \{B\}}\,(\text{conseq})
$$

先ほどの「階乗を計算するプログラム」のホーア論理の証明図を書くと（長くなるので省略するが）I のところに現れる論理式はまさに「$k \cdot n! = N!$」となる。これが、先ほどのプログラムの while 文の「ループ不変量」にあたる。

1.2　述語変換子

述語変換子は、述語 Q を別の述語 $wp[\![C]\!](Q)$ に変換します。これは「プログラム C について、事後条件 Q を成立させる事前条件のうち、論理的含意関係に関して最も弱いもの」、つまり**最弱事前条件** (weakest precondition) です。

定義 3（述語変換子意味論）　状態集合を S とする。S 上の述語の集合を $\mathrm{Pred}(S) = \{Q \mid Q : S \to \{0,1\}\}$ と定める。なお、各述語 $Q : S \to \{0,1\}$ は、$Q = \{s \in S \mid s \models Q\}$ だと考えればよい。述語どうしの順序は、含意関係によって定めるものとする。そうすると、$(\mathrm{Pred}(S), \leq)$ は完備束をなす。

$$
P \leq Q \quad \text{iff} \quad P \Rightarrow Q
$$

プログラム C の述語変換子 (predicate transformer) とは、写像

$$
wp[\![C]\!] : \mathrm{Pred}(S) \to \mathrm{Pred}(S)
$$

であって単調性 $Q_1 \leq Q_2 \ \Rightarrow\ wp[\![C]\!](Q_1) \leq wp[\![C]\!](Q_2)$ を満たすものである。具体的には以下のように帰納的に定義される。

- $wp[\![\mathrm{skip}]\!](Q) = Q$
- $wp[\![x := a]\!](Q) = Q[a/x]$
- $wp[\![C_1; C_2]\!](Q) = wp[\![C_1]\!](wp[\![C_2]\!](Q))$
- $wp[\![\text{if } \varphi \text{ then } C_1 \text{ else } C_2]\!](Q) = (\varphi \wedge wp[\![C_1]\!](Q)) \vee (\neg\varphi \wedge wp[\![C_2]\!](Q))$
- $wp[\![\text{while } \varphi \text{ do } C]\!](Q) = \mu X.\ \underbrace{((\varphi \wedge wp[\![C]\!](X)) \vee (\neg\varphi \wedge Q))}_{\text{loop characteristic function } \Phi_Q(X)}$

注意　ここで、μ は順序 $\leq$ についての最小不動点をとる操作である。$\Phi_Q : \mathrm{Pred}(S) \to \mathrm{Pred}(S)$ はスコット連続なので、クリーネの不動点定理より、$\mu X.\,\Phi_Q(X) = \bigvee_{n \in \mathbb{N}} \Phi_Q^n(\bot)$ が得られる。

上記では、「ホーア論理」と「述語変換子」という 2 つのアプローチが導入されました。ここで、両者を整理しておきましょう。

$$\begin{cases} \{P\}\, C\, \{Q\} & \text{Hoare-style} \\ P \Rightarrow wp[\![C]\!](Q) & \text{Dijkstra-style} \end{cases}$$

両者の関係として、以下が同値になることが知られています。

$$\{P\}\, C\, \{Q\} \quad \text{iff} \quad P \Rightarrow wp[\![C]\!](Q)$$

両者には、以下の違いがあります。

- ホーア三つ組は「関係的」(relational)。
  各述語 P に対し、$\{P\}\, C\, \{Q\}$ が成り立つような述語 Q は複数ありうる。
  各述語 Q に対し、$\{P\}\, C\, \{Q\}$ が成り立つような述語 P は複数ありうる。
- 最弱事前条件は「関数的」(functional)
  各述語 Q に対し、述語 $wp[\![C]\!](Q)$ は一意に定まる。

また、wp から Hoare triple を「再現」することができます。

- $\{\, wp[\![C]\!](Q)\,\}\, C\, \{Q\}$ はいつでも成り立つ。

さて、これら Hoare-style と Dijkstra-style には、それだけでなく、より本質的な違いがあります。それは、部分正当化 (partial correctness) か、全正当性 (total correctness) か、の違いです。次節ではその点について考えます。

1.3 部分正当性と全正当性

プログラムの正しさを検証するうえで、**部分正当性** (partial correctness) と**全正当性** (total correctness) という 2 種類の「正しさ」の考え方があります。ホーア三つ組 $\{P\}\, C\, \{Q\}$ は前者（部分正当性）を主張するものです。ここでさしあたり、後者（全正当性）のほうを $[P]\, C\, [Q]$ と表記して区別することにし、両者の違いを比べてみましょう。

- 部分正当性 (partial correctness) $\{P\}\, C\, \{Q\}$
  - ▷「事前条件 P から始めて、もしプログラム C が停止したら、その実行結果は事後条件 Q を満たす。」
  - ▷ プログラムが停止しないときには何も保証しない。
- 全正当性 (total correctness) $[P]\, C\, [Q]$
  - ▷「事前条件 P から始めて、プログラム C が必ず停止し、かつ、その実行結果は事後条件 Q を満たす。」
  - ▷ プログラムの停止性 (termination) も保証する。

ホーア三つ組は部分正当性 (partial correctness) なのに対し、述語変換子は全正当性 (total correctness) です。この違いは、while ループの不動点のとり方に端を発しています。

- 述語変換子における while 文の解釈は、最小不動点で与えられていたことを、思い出しましょう。

$$wp[\![\text{while } \varphi \text{ do } C]\!](Q) = \mu X.\ \underbrace{((\varphi \wedge wp[\![C]\!](X)) \vee (\neg\varphi \wedge Q))}_{\text{loop characteristic function } \Phi_Q(X)}$$

- それに対し、ホーア論理における while 文の解釈は、実は、最大不動点によって与えられていたのです。

$$\dfrac{P \Rightarrow I \qquad \dfrac{\{I \wedge \varphi\}\, C\, \{I\}}{\{I\}\, \text{while } \varphi \text{ do } C\, \{I \wedge \neg\varphi\}}\ \text{(while)} \qquad I \wedge \neg\varphi \Rightarrow Q}{\{P\}\, \text{while } \varphi \text{ do } C\, \{Q\}}\ \text{(conseq)}$$

命題 4　L を完備束とし、$f : L \to L$ を単調写像とする。このとき、以下の帰結関係（上から下）が成り立つ。

$$\frac{p \le i \qquad i \le f(i) \qquad i \le q}{p \le \nu(f(-) \wedge q)}$$

証明　$\wedge$ の普遍性より、以下の同値関係が成り立つ。

$$\frac{i \le f(i) \qquad i \le q}{\overline{\overline{i \le f(i) \wedge q}}}$$

ナスター・タルスキの定理より、

$$\nu(f(-) \wedge q) = \bigvee \{x \in L \mid x \le f(x) \wedge q\}$$

であるから、特に任意の $i \in L$ について以下の帰結関係が成り立つ。

$$\frac{i \le f(i) \wedge q}{i \le \nu(f(-) \wedge q)}$$

ここで、条件 $p \le i$ と推移性から、最終的に

$$\frac{p \le i \qquad i \le f(i) \qquad i \le q}{p \le \nu(f(-) \wedge q)}$$

が得られた。 □

上記の命題において、p は事前条件、q は事後条件、i はループ不変量に、おおよそ相当しています。ここで、

- $\boxed{f(i)}$ を $\boxed{\varphi \Rightarrow wp[\![C]\!](I)}$ に読み替える
- $\boxed{q}$ を $\boxed{\neg\varphi \Rightarrow Q}$ に読み替える

とすれば、先ほどの while ループの規則に対応していることがわかりやすいのではないかと思います。

$$\frac{p \le i \qquad i \le f(i) \qquad i \le q}{p \le \nu(f(-) \wedge q)} \quad \rightsquigarrow \quad \frac{P \Rightarrow I \qquad \dfrac{\{I \wedge \varphi\}\, C\, \{I\}}{\{I\}\, \text{while}\ \varphi\ \text{do}\ C\, \{I \wedge \neg\varphi\}} \qquad I \wedge \neg\varphi \Rightarrow Q}{\{P\}\, \text{while}\ \varphi\ \text{do}\ C\, \{Q\}}$$

つまり、ホーア論理では最大不動点 ν (gfp) を、最弱事前条件では最小不動点 μ (lfp) をとっていたことがわかります。

なお、述語変換子の中にも、部分正当化をおこなうものもあり、それは「最弱リベラル事前条件」(weakest liberal precondition, wlp) と呼ばれます。これは、while ループの解釈に lfp ではなく gfp を用いる点だけが異なります。

- $wp[\![\text{while}\ \varphi\ \text{do}\ C]\!](Q) = \mu X.\, ((\varphi \wedge wp[\![C]\!](X)) \vee (\neg\varphi \wedge Q))$
- $wlp[\![\text{while}\ \varphi\ \text{do}\ C]\!](Q) = \nu X.\, ((\varphi \wedge wp[\![C]\!](X)) \vee (\neg\varphi \wedge Q))$

以上のことをまとめると、以下の表の通りです。

	Hoare logic	predicate transformer
gfp (最大不動点)	partial correctness $\{P\}\, C\, \{Q\}$	weakest liberal precondition $wlp[\![C]\!](Q)$
lfp (最小不動点)	total correctness $[P]\, C\, [Q]$	weakest precondition $wp[\![C]\!](Q)$

さて、この節で扱った述語変換子の考え方は、圏論の言葉を使うととても見通しがよくなります。そのための準備として、次節では、圏論の関手を使ってプログラムの振舞いを捉える方法について紹介します。

2　圏論の関手でシステムの振舞いを捉える

様々な種類の状態遷移システム（例: 二分木グラフ、ストリーム、クリプキフレーム、オートマトン、マルコフ連鎖、マルコフ決定過程など）を圏論的に扱うには、**余代数**（coalgebra）と呼ばれるものが役立ちます。

2.1　余代数の定義

定義 5（余代数）　$\mathbb{C}$ を圏とし、$F : \mathbb{C} \to \mathbb{C}$ を関手とする。F-余代数とは、$\mathbb{C}$ の対象 X と射 $c : X \to F(X)$ の組 (X, c) のことと定める。また、2 つの余代数 $(X, c : X \to F(X))$, $(Y, d : Y \to F(Y))$ の間の余代数準同型とは、射 $f : X \to Y$ であって $d \circ f = F(f) \circ c$ を満たすものとする。

$$\begin{tikzcd} F(X) \arrow[r, "F(f)"] & F(Y) \\ X \arrow[u, "c"] \arrow[r, "f"'] \arrow[ur, phantom, "="] & Y \arrow[u, "d"'] \end{tikzcd}$$

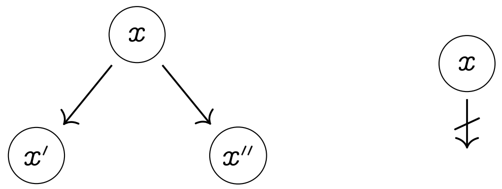
定義 6（余代数の圏）　F-余代数の圏を $\mathrm{Coalg}(F)$ と書くことにする。対象が余代数であり、射が余代数準同型であるような圏である。

$$\begin{pmatrix} F(X) \\ \uparrow c \\ X \end{pmatrix} \xrightarrow{f} \begin{pmatrix} F(Y) \\ \uparrow d \\ Y \end{pmatrix}$$

2.2　余代数の様々な例

例 7　二分木グラフは、**Set** 上の関手 $F(X) = 1 + (X \times X)$ に対する余代数 $c : X \to F(X)$ として与えられる。

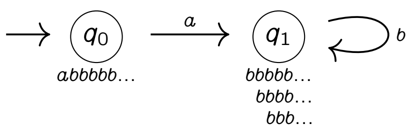
$$\text{分岐} \quad c(x) = (x', x'') \qquad \text{or} \qquad \text{停止} \quad c(x) = *$$



例 8　ストリームは、**Set** 上の関手 $F(X) = A \times X$ に対する余代数 $c : X \to F(X)$ として与えられる。

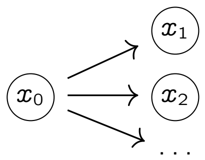
$$c(x) = (\mathrm{head}(x), \mathrm{tail}(x))$$

たとえば、ストリーム $x = (x_1, x_2, x_3, \ldots)$ について、$\mathrm{head}(x) = x_1$, $\mathrm{tail}(x) = (x_2, x_3, \ldots)$ である。



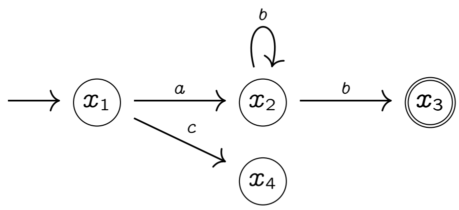
例 9　様相論理におけるクリプキフレーム (Kripke frame) は、余代数 $\delta : X \to \mathcal{P}(X)$ で与えられる。

$$\begin{array}{cccc} \delta: & X & \longrightarrow & \mathcal{P}(X) \\ & \Psi & & \Psi \\ & x & \longmapsto & \delta(x) = \{x \mid x \to x'\} \end{array}$$



例 10　非決定的オートマトンは、余代数 $\gamma : X \to 2 \times \mathcal{P}(A \times X)$ で与えられる。なお $2 = \{\bigcirc, \circledcirc\}$ とする。

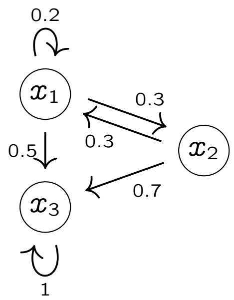
$$\gamma(x) = \big(\circledcirc, \{(a, x'), \ldots\}\big) \quad \text{or} \quad \gamma(x) = \big(\bigcirc, \{(a, x'), \ldots\}\big)$$



例 11　マルコフ連鎖は、余代数 $\gamma : X \to \mathcal{D}(X)$ で与えられる。なお、関手 $\mathcal{D} : \mathbf{Set} \to \mathbf{Set}$ は

$$\begin{aligned} \mathcal{D}(X) &= \{d : X \to [0,1] \mid \text{finite support}, \ \textstyle\sum d(x) = 1\} \\ &= \{\textstyle\sum p_i |x_i\rangle \text{, where each } p_i \text{ is a probability } d(x_i) \in [0,1].\} \\ \mathcal{D}(f) &: \mathcal{D}(X) \to \mathcal{D}(Y) \\ & \mathcal{D}(f)(\textstyle\sum p_i |x_i\rangle) = \textstyle\sum p_i |f(x_i)\rangle \end{aligned}$$

と定められているものとする。



$$\begin{aligned} \gamma(x_1) &= 0.2|x_1\rangle + 0.3|x_2\rangle + 0.5|x_3\rangle, \\ \gamma(x_2) &= 0.3|x_1\rangle + 0.7|x_3\rangle, \\ \gamma(x_3) &= |x_3\rangle. \end{aligned}$$

注意（分布関手のためのケット記法）　$\mathcal{D}(X) = \{\, d : X \to [0,1] \mid \sum d(x) = 1 \,\}$ の各要素を $\sum p_i |x_i\rangle$ と書く。ここで p_i は確率 $d(x_i) \in [0,1]$ である。例えば、コイントス $X = \{H, T\}$ を考えるとき、「公平なコイン」の分布は $d = \frac{1}{2}|H\rangle + \frac{1}{2}|T\rangle$ と書ける。ここで $|H\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$、$|T\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ である。

2.3　終余代数とその構成

ここでは、終余代数の定義と、その構成のしかたを紹介します。まずは定義です。

定義 12（終余代数）

命題 13（ランベックの補題）

系 14　べき集合の関手 $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$ には、終余代数は存在しない。

証明 $\square$

ここからは、どうやって終余代数を構成するかについて見ていきます。終余代数の構成は、順序集合についての「クーゾー・クーゾーの定理」(the Cousot-Cousot theorem) を一般化したものなので、まずはその復習から始めましょう。

命題 15（クーゾー・クーゾーの定理）

2.4　計算的作用 (computational effect) について

2.2 節でいくつか例を挙げながら紹介したように、余代数 (coalgebra) は「1 ステップの状態遷移」を表します。

$$c : X \to F(X)$$

より一般に、計算 (computation) は、以下で表されます。($X = Y$ とすれば余代数です。)

$$f : X \to F(Y)$$

ここで「ただの関数」(pure function) と「作用つき計算」(effectful computation) を区別することは重要です。

pure function	vs	effectful computation
$f : X \to Y$		$f : X \to F(Y)$

典型例としては、例外処理 (exception handling) と呼ばれる計算的作用を表す関手 $F(Y) = Y + E$ が挙げられます。ここで $+$ は余積を表し、E は "エラーの集合" と考えます。このとき、計算 $f : X \to Y + E$ は、単に入力値 $x \in X$ から出力値 $f(y) \in Y$ を生成するだけでなく、「もし例外が生じたらエラーを返す」という計算的作用を持ちます。この例のような、計算的作用を持つ計算（作用つき計算）を詳しく扱うためには、次節で導入する「モナド」と「クライスリ圏」という概念が役立ちます。(次節でもあらためてこの例外処理の例について取り上げます。)

なお、作用つき計算 $f : X \to F(Y)$ を、便宜的に $f : X \nrightarrow Y$ という特殊な矢印で表記することがよくあります。(厳密に言えば、これはモナド F のクライスリ圏における射を表す記法ですが、ここでは詳細の説明は省略します。この点については、次節を読んで「モナド」や「クライスリ圏」について学んだ後にはわかるはずです。)

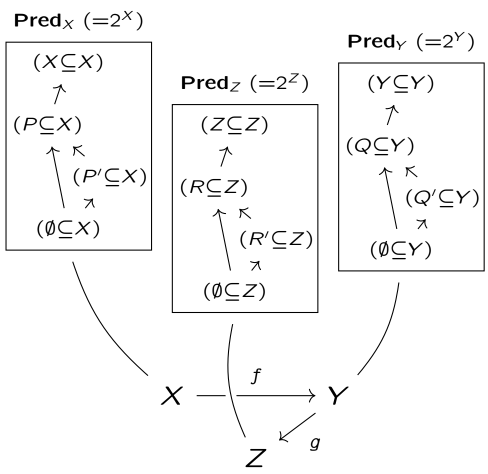
2.5　モナドとクライスリ圏

2.6 グロタンディーク・ファイブレーション入門

さて本節では、次節以降で使う「グロタンディーク・ファイブレーション」をごく簡単に導入します。複雑に見えるかもしれませんが、これを使うと、述語変換子やループ不変量を「引き戻し」と「持ち上げ」によって統一的に書けます。

ここでは、まず直観を説明してから、後ほど正確な定義を述べます。例として $p : \mathbf{Pred} \to \mathbf{Set}$ というファイブレーションを考えてみましょう。ひとつずつステップを踏んで理解していきます。

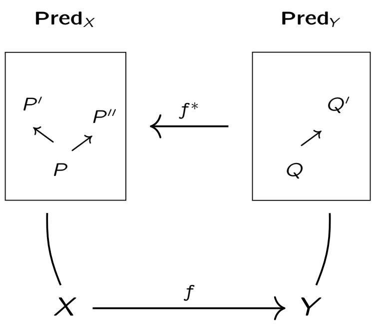
1. それぞれの対象 $X \in \mathbf{Set}$ に対して、「ファイバー」(fiber) と呼ばれる完備束 $\mathbf{Pred}_X$ $(:= 2^X)$ を割り当てます。



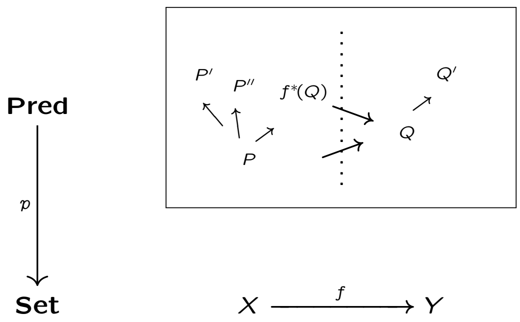
2. それぞれの射 $f : X \to Y$ に対して、$f^* : \mathbf{Pred}_Y \to \mathbf{Pred}_X$ を定めます。

$$
\begin{array}{cccc}
f^* : & 2^Y & \longrightarrow & 2^X \\
& \Cup & & \Cup \\
& (Y \xrightarrow{g} 2) & \longmapsto & (X \xrightarrow{f} Y \xrightarrow{g} 2)
\end{array}
$$

直観的には、以下の図のようなものだと理解するのがよいでしょう。



3. 以上で与えられたデータ $\Big((\mathbf{Pred}_X)_{X \in \mathbf{Set}} \, , \, (f^* : \mathbf{Pred}_Y \to \mathbf{Pred}_X)_{f : X \to Y \text{ in } \mathbf{Set}} \Big)$ をもとにして、各ファイバーを貼り合わせて、「全体圏」(total category) と呼ばれる圏 $\mathbf{Pred}$ をつくります。



4. この **Pred** は、対象が組 $(X,\ P \subseteq X)$ であり、射が写像 $(X,\ P \subseteq X) \to (Y,\ Q \subseteq Y)$ であるような圏です。
   - 対象:

$$|\mathbf{Pred}| = \bigsqcup_{X \in \mathbf{Set}} |\mathbf{Pred}_X|$$

   - 射:

$$\frac{(X,\ P \subseteq X) \longrightarrow (Y,\ Q \subseteq Y) \text{ in } \mathbf{Pred}}{X \xrightarrow{f} Y \text{ in } \mathbf{Set} \text{ s.t. } P \sqsubseteq f^*(Q) \text{ in } \mathbf{Pred}_X}$$

5. 以下の条件を満たす関手 $p : \mathbf{Pred} \to \mathbf{Set}$ は「ファイブレーション」(fibration) と呼ばれます。(詳細は後述)

$$\begin{tikzcd} \mathbf{Pred} \arrow[d, "p"'] \\ \mathbf{Set} \end{tikzcd} \qquad \begin{tikzcd} & Q \arrow[d, mapsto] \\ X \arrow[r, "f"'] & Y \end{tikzcd} \quad \Longrightarrow \quad \begin{tikzcd} {}^{\exists} f^*(Q) \arrow[r, "{}^{\exists}\overline{f}(Q)"'] \arrow[d, mapsto] & Q \arrow[d, mapsto] \\ X \arrow[r, "f"'] & Y \end{tikzcd} \quad \text{s.t.} \quad \begin{tikzcd} R \arrow[d, mapsto] \arrow[r, dashed, "\exists! s"] \arrow[rr, bend left, "\forall r"] & f^*(Q) \arrow[r, "\overline{f}(Q)"'] \arrow[d, mapsto] & Q \arrow[d, mapsto] \\ p(R) \arrow[r, "\forall g"'] \arrow[rr, bend left, "g \circ f"] & X \arrow[r, "f"'] & Y \end{tikzcd}$$

6. ここで、**Pred** を「全体圏」といい、**Set** を「基底圏」といいます。
   また、$\overline{f}(Q)$ を、f の Q による「カルテシアン持ち上げ」(cartesian lifting) といいます。
   そして、f^* を、f の「置換」(substitution) といい、対象 $P, Q \in \mathbf{Pred}$ で $pP = pQ = Y$ なるものと、**Pred** の射 $(P \xrightarrow{u} Q)$ で $pu = \mathrm{id}_Y$ なるものについて、$\overline{f}_Q \circ f^* u = u \circ \overline{f}_P$ を満たすものと定められています。

$$\begin{tikzcd} & f^*P \arrow[r, "\overline{f}_P"] \arrow[d, "f^*u"'] & P \arrow[d, "u"] \\ \mathbf{Pred} \arrow[d, "p"'] & f^*Q \arrow[r, "\overline{f}_Q"'] & Q \\ \mathbf{Set} & X \arrow[r, "f"'] & Y \end{tikzcd}$$

7. ここで、$(-)^*$ や $\overline{(-)}$ は "関手的" (functorial)、つまり恒等射と合成に整合します。

$$\begin{aligned} \mathrm{id}_Y^* Q &= Q, & (g \circ f)^*(Q) &= f^*(g^* Q), \\ \overline{\mathrm{id}_Y}(Q) &= \mathrm{id}_Q, & \overline{g \circ f}(Q) &= \overline{g} Q \circ \overline{f}(g^* Q). \end{aligned}$$

8. 自己関手 $F : \mathbf{Set} \to \mathbf{Set}$ と、ファイブレーション $p : \mathbf{Pred} \to \mathbf{Set}$ について、
   関手 $\dot{F} : \mathbf{Pred} \to \mathbf{Pred}$ が、p に沿った F の「持ち上げ」(fibred lifting) であるとは、

$$\begin{tikzcd} \mathbf{Pred} \arrow[r, "\dot{F}"] \arrow[d, "p"'] \arrow[dr, phantom, "/\!/"] & \mathbf{Pred} \arrow[d, "p"] \\ \mathbf{Set} \arrow[r, "F"'] & \mathbf{Set} \end{tikzcd} \quad \text{かつ} \quad \begin{tikzcd} \mathbf{Pred}_Y \arrow[r, "f^*"] \arrow[d, "\dot{F}"'] \arrow[dr, phantom, "\cong"] & \mathbf{Pred}_X \arrow[d, "\dot{F}"] \\ \mathbf{Set}_{FY} \arrow[r, "(Ff)^*"'] & \mathbf{Set}_{FX} \end{tikzcd}$$

   を満たすことです。特に余代数との関わりの中では、以下の図のような状況を考えることが多いです。基底の関手 F をどうやって $\dot{F}$ に持ち上げるかが重要になります。(後述)

$$\begin{tikzcd} \mathbf{Pred} \arrow[loop left, "\dot{F}"] \arrow[d, "p"] \\ \mathbf{Set} \arrow[loop left, "F"] \end{tikzcd} \qquad \begin{tikzcd} \mathbf{Pred}_X \arrow[r, bend left, "\dot{F}_X"] & \mathbf{Pred}_{FX} \arrow[l, bend left, "c^*"] \\ X \arrow[r, "c"'] & FX \end{tikzcd}$$

以下、ファイブレーションの正確な定義を念のため書いておきます。（ですが、上記の $p\colon \mathbf{Pred} \to \mathbf{Set}$ の考え方だけをきちんと押さえておけば、次節以降の内容を理解するうえでまったく問題はありません。）

定義 16（カルテシアン）　$p : \mathbb{E} \to \mathbb{B}$ を関手とします。$\mathbb{E}$ の射 $(X \xrightarrow{f} Y)$ が p-カルテシアン (p-cartesian) であるとは、すべての $Z \in \mathbb{E}$ について、以下のプルバック図式が可換になることである。

$$
\begin{tikzcd}
\mathbb{E}(Z,X) \arrow[r, "f\circ(-)"] \arrow[d, "p_{ZX}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E}(Z,Y) \arrow[d, "p_{ZY}"] \\
\mathbb{B}(pZ,pX) \arrow[r, "pf\circ(-)"'] & \mathbb{B}(pZ,pY)
\end{tikzcd}
\qquad \text{i.e.} \qquad
\begin{tikzcd}
\exists!\, h \arrow[r, mapsto] \arrow[d, mapsto] & \forall\, g \arrow[d, mapsto] \\
\forall\, u \arrow[r, mapsto] & pf \circ u = pg
\end{tikzcd}
$$

注意　要するに以下と同じ。

$$
\begin{tikzcd}
\mathbb{E} \arrow[d, "p"'] & Z \arrow[dr, dashed, "\exists!\, h"'] \arrow[drr, bend left, "\forall\, g"] & & \\
& & X \arrow[r, "f"'] & Y \\
\mathbb{B} & pZ \arrow[dr, "\forall\, u"'] \arrow[drr, bend left, "pg"] & & \\
& & pX \arrow[r, "pf"'] & pY
\end{tikzcd}
$$

定義 17（ファイブレーション）　関手 $p : \mathbb{E} \to \mathbb{B}$ がファイブレーション (fibration) であるとは、$\mathbb{B}$ の任意の射 $u : I \to J$ と、J の上にある（つまり $pY = J$ であるような）任意の対象 $Y \in \mathbb{E}$ に対し、$\mathbb{E}$ の p-カルテシアン射 $f : X \to Y$ が u の上に存在する（つまり $pf = u$ である）ことである。

注意　要するに以下と同じ。

$$
\begin{tikzcd}
\mathbb{E} \arrow[d, "p"'] \\
\mathbb{B}
\end{tikzcd}
\qquad
\begin{tikzcd}
& Y \\
I \arrow[r, "u"'] & J
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
\exists\, X \arrow[r, "\text{cartesian}", "\exists\, f"'] & Y \\
I \arrow[r, "u"'] & J
\end{tikzcd}
$$

定義 18（ファイバー）　$p : \mathbb{E} \to \mathbb{B}$ を関手とする。各対象 $I \in \mathbb{B}$ について、圏 $\mathbb{E}_I$ をファイバー (fiber) と呼ぶ。対象は、I の上にある（つまり $pX = I$ を満たすような）$X \in \mathbb{E}$ であり、射は id_I の上にある（つまり $pf = \mathrm{id}_I$ を満たすような）$\mathbb{E}$ の射 $f : X \to Y$ である。

定義 19（置換）　$p : \mathbb{E} \to \mathbb{B}$ をファイブレーションとし、$u : I \to J$ を $\mathbb{B}$ の射とします。u の置換 (substitution) とは、関手 $u^*\colon \mathbb{E}_J \to \mathbb{E}_I$ であって以下を満たすものである。（ここで $\overline{u}_X\colon u^*X \to X$ および $\overline{u}_Y\colon u^*Y \to Y$ は p-カルテシアン射でかつ $p(\overline{u}_X) = u$ と $p(\overline{u}_Y) = u$ を満たすものとする。）

$$
\begin{tikzcd}
u^*X \arrow[r, "\overline{u}_X"] \arrow[d, dashed, "u^*f"'] & X \arrow[d, "f"] \\
u^*Y \arrow[r, "\overline{u}_Y"] & Y
\end{tikzcd}
$$

$$
I \xrightarrow{u} J
$$

- 対象 $X \in \mathbb{E}_J$ に対し、対象 $u^*X \in \mathbb{E}_I$ を返し、

- $\mathbb{E}_J$ の射 $f : X \to Y$ に対し、射 $u^*f : u^*X \to u^*Y$ が一意に定まり、以下のプルバック図式を満たす：

$$\begin{tikzcd} \mathbb{E}(u^*X, u^*Y) \arrow[r, "\overline{u}_Y\circ(-)"] \arrow[d, "p"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbb{E}(u^*X, Y) \arrow[d, "p"] \\ \mathbb{B}(I, I) \arrow[r, "u\circ(-)"'] & \mathbb{B}(I, J) \end{tikzcd} \quad \text{i.e.} \quad \begin{tikzcd} \exists!\, u^*f \arrow[r, mapsto, "②"] \arrow[d, mapsto, "①"'] & f\circ\overline{u}_X \arrow[d, mapsto] \\ \mathrm{id}_I \arrow[r, mapsto] & u \end{tikzcd}$$

すなわち、① $p(u^*f) = \mathrm{id}_I$ と ② $f \circ \overline{u}_X = \overline{u}_Y \circ u^*f$ を満たす。

注意　同一の射の上にあるカルテシアン射は "一意な同型を除いて一意" (unique up to unique iso) である。より具体的に言うと、ファイブレーション $p : \mathbb{E} \to \mathbb{B}$ と、$\mathbb{B}$ の射 $u : I \to J$ が与えられ、$Y \in \mathbb{E}_J$ があり、2つの p-カルテシアン射 $\overline{u}_Y : u^*Y \to Y$ と $f : X \to Y$ がいずれも u の上にあるとき、一意な同型射 $\alpha : X \xrightarrow{\cong} u^*Y$ がファイバー $\mathbb{E}_I$ に存在し、$f = \overline{u}_Y \circ \alpha$ が成り立つ。(証明略)

$$\begin{tikzcd} & X \arrow[r, "f"] \arrow[d, phantom, "\cong" rotate=90] & Y \arrow[d, equal] \\ \mathbb{E} \arrow[dd, "p"'] & u^*Y \arrow[r, "\overline{u}_Y"'] & Y \\ & & \\ \mathbb{B} & I \arrow[r, "u"] & J \end{tikzcd}$$

命題 20　$p : \mathbb{E} \to \mathbb{B}$ をファイブレーションとする。$F : \mathbb{B} \to \mathbb{B}$, $\dot{F} : \mathbb{E} \to \mathbb{E}$ をそれぞれ $\mathbb{B}, \mathbb{E}$ の自己関手とし、$p \circ \dot{F} = F \circ p$ を満たすものとする。

$$\begin{tikzcd} \mathbb{E} \arrow[r, "\dot{F}"] \arrow[d, "p"'] & \mathbb{E} \arrow[d, "p"] \\ \mathbb{B} \arrow[r, "F"'] & \mathbb{B} \end{tikzcd}$$

このとき、$\mathbb{B}$ の任意の射 $(I \xrightarrow{u} J)$ に対し、以下の自然変換 $\gamma : \dot{F} \circ u^* \Rightarrow (Fu)^* \circ \dot{F}$ が存在する。

$$\begin{tikzcd} \mathbb{E}_J \arrow[r, "u^*"] \arrow[d, "\dot{F}"'] \arrow[dr, phantom, "\swarrow\gamma"] & \mathbb{E}_I \arrow[d, "\dot{F}"] \\ \mathbb{E}_{FJ} \arrow[r, "(Fu)^*"'] & \mathbb{E}_{FI} \end{tikzcd} \quad \text{i.e.} \quad \begin{tikzcd} X \arrow[r, mapsto, "u^*"] \arrow[dd, mapsto] & u^*X \arrow[d, mapsto] \\ & \dot{F}(u^*X) \arrow[d, "\gamma_X"] \\ \dot{F}X \arrow[r, mapsto] & (Fu)^*\dot{F}X \end{tikzcd}$$

証明　$X \xmapsto{(pX=J)} J$ および $\dot{F}X \xmapsto{(p\dot{F}X=FpX=FJ)} FJ$ と考えると

$$\begin{tikzcd} \dot{F} \circlearrowright \mathbb{E} \arrow[d, "p"] & u^*X \arrow[r, "\overline{u}"] & X & & \dot{F}(u^*X) \arrow[r, "\dot{F}\overline{u}"] & \dot{F}X \\ F \circlearrowright \mathbb{B} & I \arrow[r, "u"'] & J & \longmapsto & FI \arrow[r, "u"'] & FJ \end{tikzcd}$$

が得られる。それゆえ、

$$\begin{tikzcd} & \dot{F}(u^*X) \arrow[d, dashed, "\gamma_X"'] \arrow[dr, bend left, "\dot{F}\overline{u}"] & \\ \mathbb{E} \arrow[d, "p"'] & (Fu)^*\dot{F}X \arrow[r, "\overline{Fu}"'] & \dot{F}X \\ \mathbb{B} & FI \arrow[r, "Fu"'] & FJ \end{tikzcd}$$

となる。(γ の自然性の証明はここでは省略する。)　□

定義 21（カルテシアンを保つ） $p : \mathbb{E} \to \mathbb{B}$ をファイブレーションとする。関手 $\dot{F} : \mathbb{E} \to \mathbb{E}$ が p-カルテシアンを保つ (preserves p-cartesian) とは、$\dot{F}$ が各 p-カルテシアン射 f を別の p-カルテシアン射 $\dot{F}f$ に送ること。

$$(X \xrightarrow{f} Y) \text{ in } \mathbb{E} \text{ is } p\text{-cartesian} \implies (\dot{F}X \xrightarrow{\dot{F}f} \dot{F}Y) \text{ in } \mathbb{E} \text{ is } p\text{-cartesian}.$$

命題 22 $p : \mathbb{E} \to \mathbb{B}$ をファイブレーションとする。$F : \mathbb{B} \to \mathbb{B}$, $\dot{F} : \mathbb{E} \to \mathbb{E}$ をそれぞれ $\mathbb{B}, \mathbb{E}$ の自己関手とし、$p \circ \dot{F} = F \circ p$ を満たすものとする。（ここまでの状況は、先ほどの**命題 20** のときと同じ。）

$$\begin{tikzcd} \mathbb{E} \arrow[r, "\dot{F}"] \arrow[d, "p"'] & \mathbb{E} \arrow[d, "p"] \\ \mathbb{B} \arrow[r, "F"'] & \mathbb{B} \end{tikzcd}$$

このとき、以下の同値関係が成り立つ。

$$\dot{F} \text{ preserves } p\text{-cartesian} \iff \gamma : \dot{F} \circ u^* \Rightarrow (Fu)^* \circ \dot{F} \text{ is isomorphic.}$$

証明 任意の p-カルテシアン射 $f : X \to Y$ をとり、以下の状況を考える。

$$\begin{tikzcd} \mathbb{E}_{pY} \arrow[r, "(pf)^*"] \arrow[d, "\dot{F}"'] & \mathbb{E}_{pX} \arrow[d, "\dot{F}"] \arrow[dl, phantom, "\swarrow\gamma"] \\ \mathbb{E}_{FpY} \arrow[r, "(Fpf)^*"'] & \mathbb{E}_{FpX} \end{tikzcd} \quad \text{i.e.} \quad \begin{tikzcd} Y \arrow[r, mapsto] \arrow[dd, mapsto] & X \arrow[d, mapsto] \\ & \dot{F}X \arrow[d, "\gamma_Y"] \\ \dot{F}Y \arrow[r, mapsto] & (Fpf)^*\dot{F}Y \end{tikzcd}$$

言い換えると、状況は以下のとおり。

$$\begin{tikzcd} \mathbb{E} \arrow[d, "p"'] & X \arrow[r, "\overline{pf} = f"] & Y & & \dot{F}X \arrow[d, dashed, "\gamma_Y"'] \arrow[dr, bend left, "\dot{F}f"] & \\ \mathbb{B} & & & \longmapsto & (Fpf)^*FY \arrow[r, "\overline{Fpf}"'] & \dot{F}Y \\ & pX \arrow[r, "pf"'] & pY & & FpX \arrow[r, "Fpf"'] & FpY \end{tikzcd}$$

ここで、$\dot{F}f$ が p-カルテシアンである $\iff$ γ_Y が同型射である。（なぜなら、$\dot{F}f$ と $\overline{Fpf}$ はいずれも Fpf の上にあり、しかも $\overline{Fpf}$ は定義からして p-カルテシアンであるので、($\Rightarrow$): もし $\dot{F}f$ も p-カルテシアンなら、一意な同型射 γ_Y が手に入るし、($\Leftarrow$): もし γ_Y が同型射なら、$\dot{F}f$ は p-カルテシアンである。） □

定義 23（関手の持ち上げ） $p : \mathbb{E} \to \mathbb{B}$ をファイブレーションとする。$F : \mathbb{B} \to \mathbb{B}$ を $\mathbb{B}$ における自己関手とする。このとき、$\dot{F} : \mathbb{E} \to \mathbb{E}$ が p に沿った F のファイバー持ち上げ (fibered lifting) であるとは、$p \circ \dot{F} = F \circ p$ を満たし、かつ $\dot{F}$ がカルテシアンを保つ（cf. **定義 21**）ことである。

注意 すなわち、$\dot{F}$ は $\begin{tikzcd} \mathbb{E} \arrow[r, "\dot{F}"] \arrow[d, "p"'] \arrow[dr, phantom, "/\!/"] & \mathbb{E} \arrow[d, "p"] \\ \mathbb{B} \arrow[r, "F"'] & \mathbb{B} \end{tikzcd}$ と $\begin{tikzcd} \mathbb{E}_J \arrow[r, "u^*"] \arrow[d, "\dot{F}"'] \arrow[dr, phantom, "\cong"] & \mathbb{E}_I \arrow[d, "\dot{F}"] \\ \mathbb{E}_{FJ} \arrow[r, "(Fu)^*"'] & \mathbb{E}_{FI} \end{tikzcd}$ を満たす。

- 前者より、各 $X \in \mathbb{B}$ について、$\dot{F}_X : \mathbb{E}_X \to \mathbb{E}_{FX}$ を定義できることが従う。
- 後者は、「関手の持ち上げが置換と整合的である」ということを意味する。

3　最弱事前条件をどのように手に入れるか

さて、本節では、非決定的プログラムや確率的プログラムなどといった様々な種類のプログラムに対する述語変換子意味論を得る方法について、圏論的な視座から考えます。一般に、状態の集合 X, Y と真理値の集合 Ω が与えられたときに、効果付き計算 $c : X \to FY$ に基づいて、事後条件 $q : Y \to \Omega$ を最弱事前条件 $wp[\![c]\!](q) : X \to \Omega$ に変換するための関数である「述語変換子」 $wp[\![c]\!] : \Omega^Y \to \Omega^X$ を手に入れるために、どんな手続きを経たらよいでしょうか。

いくつか具体例を見てから、その後に一般論を考えましょう。

3.1　具体例

以下では 3 つのケースについて具体例を考えます。

具体例 (1) 決定的な 2 値的モデル検査の場合

……

具体例 (2) 非決定的な 2 値的モデル検査の場合

……

具体例 (3) 確率的モデル検査の場合

……

3.2　一般論

これまで見た具体例をふまえて、一般論を展開していきましょう。

参考文献

[1]