『圏論入門』第 1 版第 1 刷及び第 2 刷正誤表

p.4 下から 4–5 行目

誤：関係 $F \subset A \times B$，$G \subset B \times C$ があったとき，関係 $G \circ F \in A \times C$ を

正：関係 $F \subset A \times B$，$G \subset B \times C$ があったとき，関係 $G \circ F \subset A \times C$ を

p.17 図 1.5

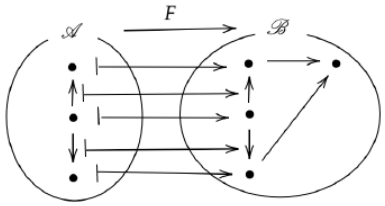


図 1.5: 順序集合間の関手

p.23 図 1.10

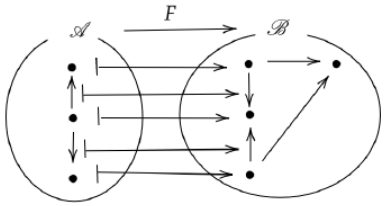


図 1.10: 逆向きの矢印が対応する

p.24 プログラム 1.2 の 3 行目

誤：　`fmap :: (a -> b) -> f a - f b`

正：　`fmap :: (a -> b) -> f a -> f b`

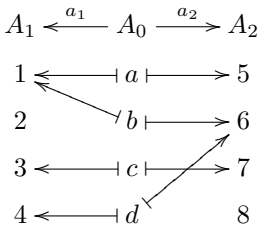
p.39 図 1.22 の flat にされたリストの数字の並び

誤：1 2 3 4 5 6 7

正：3 2 4 1 6 5 7

p.88 下から 1 行目〜p.91 7 行目までを次に差し替え

[**例 3.15**] $A_0 = \{a, b, c, d\}$, $A_1 = \{1, 2, 3, 4\}$, $A_2 = \{5, 6, 7, 8\}$ とする.



のような関数 $a_1 : A_0 \to A_1$ と $a_2 : A_0 \to A_2$ を持ってくると，押し出し，あるいはファイバー余積は,

$$A_1 \underset{A_0}{\cup} A_2 = \{L2,\ G[1,4][5,6],\ G[3][7],\ R8\},$$

になっている．記号について説明しておく．$L2$ と $R8$ は貼り合わせの操作の対象外となった点で，それぞれ A_1 の要素 2，A_2 の 8 と同一視してよい．$G[1,4][5,6]$ は A_1 の点 1，4，と A_2 の点 5，6 を貼り合わせたものを表す．これら 4 点は 6 本の $\mapsto$ で結ばれていることを確認してほしい．$G[3][7]$ は A_1 の点 3，と A_2 の点 7 を貼り合わせたものを表す.

このファイバー余積は以下の手順で構成される．まず，単純に A_1 と A_2 の排他的合併をとる．次に，$x_1 \in A_1$ と $x_2 \in A_2$ がある $x_0 \in A_0$ によって $x_1 = a_1(x_0)$ かつ $x_2 = a_2(x_0)$ であったとき，x_1 と x_2 を貼り合わせる．同一視すると言ってもよい．この同一視を推移的にやり切った結果が押し出しである．**貼り合わせ付き排他的合併** (disjoint union with glueing) と呼ばれることもある.

この例をプログラムしてみよう.

プログラム 3.5: fibercoproduct.hs

```
1  data A0 = Va | Vb | Vc | Vd deriving (Ord, Eq, Enum, Show)
2  data A1 = V1 | V2 | V3 | V4 deriving (Ord, Eq, Enum, Show)
3  data A2 = V5 | V6 | V7 | V8 deriving (Ord, Eq, Enum, Show)
4
5  -- Disjoint Union with Gluing
6  data DUwG a b c =
7    L b | R c | G [b] [c]  deriving (Ord, Eq, Show)
8
9  fibercoproduct :: (Ord a, Ord b, Ord c) =>
10   (a -> b) -> (a -> c) -> [a] -> [b] -> [c] -> [DUwG a b c]
11 fibercoproduct p q as xs ys =
12   [L x | x <- xs, not (x `elem` (fmap p as)) ] ++
13   glue p q as xs ys ++
14   [R y | y <- ys, not (y `elem` (fmap q as)) ]
15
16 glue :: (Ord a, Ord b, Ord c) =>
```

```
17   (a -> b) -> (a -> c) -> [a] -> [b] -> [c] -> [DUwG a b c]
18 glue p q as xs ys =
19   let
20       oneStepGlues = [([p a],[q a]) | a<-as]
21       classes = collect oneStepGlues
22   in
23       map (\xys -> G (fst xys) (snd xys)) classes
24
25 collect :: (Ord b, Ord c) => [([b],[c])] -> [([b],[c])]
26 collect []     = []
27 collect [x]    = [x]
28 collect (x:xs) =
29     fst folded : snd folded
30   where
31     folded = foldl f (x,[]) (collect xs)
32     f (z,zs) y
33       | equivQ z y  = (merge z y, zs)
34       | otherwise   = (z, y:zs)
35
36 equivQ :: (Ord b, Ord c) => ([b],[c]) -> ([b],[c]) -> Bool
37 equivQ (bs1,cs1) (bs2,cs2) =
38     [ x | x <- bs1, y <- bs2, x == y] /= [] ||
39     [u | u <- cs1, v <- cs2, u == v] /= []
40
41 merge :: (Ord b, Ord c) => ([b],[c]) -> ([b],[c]) -> ([b],[c])
42 merge (bs1,cs1) (bs2,cs2) =
43     (uniq (bs1++bs2), uniq (cs1++cs2))
44
45 -- unique sort
46 uniq :: Ord a => [a] -> [a]
47 uniq xs =
48   foldr ins [] xs
49     where
50       ins y [] = [y]
51       ins y (z:zs)
52         | y<z   = y:z:zs
53         | y==z  = z:zs
54         | otherwise = z:ins y zs
55
56 --------------------------------
57 -- testdata
58 -- Both a1 and a2 are injective.
```

```
59
60 a0s = [Va .. Vd]
61 a1s = [V1 .. V4]
62 a2s = [V5 .. V8]
63
64 a1 :: A0 -> A1
65 a1 Va = V1
66 a1 Vb = V2
67 a1 Vc = V3
68 a1 Vd = V4
69
70 a2 :: A0 -> A2
71 a2 Va = V5
72 a2 Vb = V6
73 a2 Vc = V7
74 a2 Vd = V8
75
76 {-- suggested tests
77 fibercoproduct a1 a2 a0s a1s a2s
78 ==> [G [V1] [V5],G [V3] [V7],G [V4] [V8],G [V2] [V6]]
79 -}
80
81 --------------------------------
82 -- b2 is injective while b1 is not.
83
84 b1 :: A0 -> A1
85 b1 Va = V1
86 b1 Vb = V1
87 b1 Vc = V3
88 b1 Vd = V3
89
90 b2 :: A0 -> A2
91 b2 Va = V5
92 b2 Vb = V6
93 b2 Vc = V7
94 b2 Vd = V8
95
96 {-- suggested test
97 fibercoproduct b1 b2 a0s a1s a2s
98 ==> [L V2,L V4,G [V1] [V5,V6],G [V3] [V7,V8]]
99 -}
100
```

```
101 ----------------------------------------------
102 -- c1 and c2 are non-injective.
103
104 c1 :: A0 -> A1
105 c1 Va = V1
106 c1 Vb = V1
107 c1 Vc = V3
108 c1 Vd = V3
109
110 c2 :: A0 -> A2
111 c2 Va = V6
112 c2 Vb = V6
113 c2 Vc = V8
114 c2 Vd = V8
115
116 {-- suggested test
117 fibercoproduct c1 c2 a0s a1s a2s
118 ==> [L V2,L V4,G [V1] [V6],G [V3] [V8],R V5,R V7]
119 -}
120
121 ----------------------------------------------
122 -- non trivial transition
123 d1 :: A0 -> A1
124 d1 Va = V1
125 d1 Vb = V1
126 d1 Vc = V3
127 d1 Vd = V4
128
129 d2 :: A0 -> A2
130 d2 Va = V5
131 d2 Vb = V6
132 d2 Vc = V7
133 d2 Vd = V6
134
135 {- suggested test
136 fibercoproduct d1 d2 a0s a1s a2s
137 ==> [L V2,G [V1,V4] [V5,V6],G [V3] [V7],R V8]
138 -}
```

1-3 行目は例に登場する対象にあたるデータ型の定義だ.

6-7 行目は貼り合わせ付き排他的合併のためのデータ型を定義している．値コンストラクターの L と R は貼り合わせが行われない点のコンストラクターで排他性を保証するためのラベル付けを

行っている．G は貼り合わせによって出来上がった点のコンストラクターだ．

9-14 行目はファイバー余積を構成する関数である．引数 p，q は余射影，as は余射影たちの共通の始域の要素のリスト，xs，ys は 2 つの余射影のそれぞれの終域の要素のリストになっている．この関数は 3 つのリストを作って，それを連結して戻り値としている．詳しく見てみよう．まず，12 行目のリストの意図は，余射影 p の終域の要素を x <- xs で取り出して，個別に，余射影の像になっていないかどうかを

```
not (x `elem` (fmap p xs))
```

でチェックして，像になっていないものだけ取り出し，L を被せる，というものだ．13 行目は貼り合わせの結果生まれた点のリストである．14 行目は 12 行目と同様に貼り合わせに関与しない A_2 の点にラベル R をつけたもののリストになっている．これらを合併すればファイバー余積になる．

16-23 行目の glue 関数の貼り合わせの戦略は以下の通りである．

- リスト oneStepGlues はペアからなる．まず，各点 $a \in A_0$ に対して ([p a], [q a]) というペアを作る．そのようなペアからこれらは次のステップの推移的拡張の種になる．
- 25-34 行目の collect 関数により同値類のリストを作成する．実質的に推移包の計算をしている．

テストは

```
*Main> fibercoproduct d1 d2 a0s a1s a2s
[L V2,G [V1,V4] [V5,V6],G [V3] [V7],R V8]
```

のように実行すればよい．

p.110 7 行目

誤：
$$
\begin{aligned}
F(\mathrm{cod}(1_1)) &= F(1) = \{3,4,5\} \\
F(\mathrm{cod}(1_2)) &= F(2) = \{6,7\} \\
F(\mathrm{cod}(f)) &= F(2) = \{6,7\} \\
\prod_{f \in \mathrm{Mor}(J)} F(\mathrm{cod}(f)) &= \{3,4,5\} \times \{6,7\} \times \{6,7\}
\end{aligned}
$$

正：
$$
\begin{aligned}
F(\mathrm{cod}(1_1)) &= F(1) = \{3,4,5\} \\
F(\mathrm{cod}(1_2)) &= F(2) = \{6,7\} \\
F(\mathrm{cod}(f)) &= F(2) = \{6,7\} \\
\prod_{h \in \mathrm{Mor}(J)} F(\mathrm{cod}(h)) &= \{3,4,5\} \times \{6,7\} \times \{6,7\}
\end{aligned}
$$

p.116 下から 3 行目

誤：$\left(F \circ \mathrm{dom}, \quad \{p_{\mathrm{dom}(f)}\}_{f \in \mathrm{Mor}(J)}, \coprod_{j \in \mathrm{obj}(J)} \right)$

正：$\left(F \circ \mathrm{dom}, \quad \{p_{\mathrm{dom}(f)}\}_{f \in \mathrm{Mor}(J)}, \coprod_{j \in \mathrm{obj}(J)} F(j) \right)$

p.220 2 行目

誤：
```
class  Functor f  where
fmap        :: (a -> b) -> f a -> f b
```
正：
```
class Monad f where
(>>=) :: m a -> (a -> m b) -> m b
return :: a -> m a
```

p.235 下から 5 行目

誤：$H^A = \mathscr{A}(A, -) : \mathscr{A} \to \mathbf{Set}$

正：$H^A = \mathscr{A}(A, -) : \mathrm{obj}(\mathscr{A}) \to \mathbf{Set}$

p.240 下から 13 行目

誤：$\mathscr{A}^{\mathrm{op}}$

正：$\mathrm{obj}(\mathscr{A}^{\mathrm{op}})$

p.241 下から 6 行目

誤：$\{0, 1\} \to X$

正：$X \to \{0, 1\}$

p.265 本文 5 行目

誤：15-16 行目では

正：19-21 行目では

p.271 下から 7 行目 8 行目の間に以下のパラグラフを挿入する．

内包原理は以下のような形に制限する．

$$\{x \mid x \in u, \varphi(x)\},$$

ここで，u はすでに分かっている集合である．すでに分かっている集合は以下の構成原理から生まれるユニバースの要素に限定する．このようにすると厳密性を犠牲にせずに実用的な理論が展開できることが分かっている．

p.271 下から 7 行目

誤：以下の性質を満たす集合

正：以下の性質を満たす集合 U

p.271 ユニバースの性質（2）

誤：$x \in U$ かつ $y \in U$ ならば $\{x, y\} \in U$ である．

正：$x \in U$ かつ $y \in U$ ならば $\{x, y\} \in U$，$\langle x, y \rangle \in U$，$u \times v \in U$ である．

p.272 8 行目

誤：$\{x \in u \mid \varphi(x)\}$ のように使うことができる．

正：$\{x \mid x \in u,\, \varphi(x)\}$ のように使うことができる．

p.297 2 行目

誤： `\f - f y`
正： `\f -> f y`

p.246 下から 3 行目の式
誤： $y : [\mathscr{A}, \mathbf{Set}](H_{-}, X) \to X(-)$
正： $y : [\mathscr{A}, \mathbf{Set}](H^{-}, X) \to X(-)$

以上