

『離散構造』 3 章 (関数) の演習問題 の解答

2010.01.08、 亀山

問題 1 以下に示す対応付けは、 $\mathcal{R}$ から $\mathcal{R}$ への関数となるか、また、関数である場合、単射であるか、さらに、全射であるか答えなさい。

同様に、以下の対応付けが、 $\mathcal{N}$ から $\mathcal{N}$ への関数となるか、また、関数である場合、単射であるか、さらに、全射であるか答えなさい。

(a) x に対して $x^3 - x$ を対応付ける。

答: $\mathcal{R}$ 上の関数 ($\mathcal{R}$ から $\mathcal{R}$ への関数) になる。理由: $x \in \mathcal{R}$ に対して $x^3 - x \in \mathcal{R}$ はただひとつ存在するから。

答: $\mathcal{R}$ から $\mathcal{R}$ への単射にならない。理由: $x = 0$ と $x = 1$ の両方とも、0 に対応するから。

答: $\mathcal{R}$ から $\mathcal{R}$ への全射になる。理由: x が負の無限大に近付くとき $x^3 - x$ も負の無限大に近付く、 x が正の無限大に近付くとき $x^3 - x$ も正の無限大に近付く、さらに、 $f(x) = x^3 - x$ という関数は連続関数なので、結局、負の無限大から正の無限大まで全ての实数値へ対応付く。

答: $\mathcal{N}$ 上の関数になる。理由: $x \in \mathcal{N}$ に対して $x^3 - x \in \mathcal{N}$ はただひとつ存在するから。

答: $\mathcal{N}$ から $\mathcal{N}$ への単射にならない。理由: $x = 0$ と $x = 1$ の両方とも、0 に対応するから。

答: $\mathcal{N}$ から $\mathcal{N}$ への全射にならない。理由: $x^3 - x = 2$ となる自然数 x は存在しないから。

(b) y に対して $x^3 - x = y$ となる数 x を対応付ける。(問題を 2010/01/05 に修正)

答: $\mathcal{R}$ 上の関数にならない。理由: $y = 0$ に対して $x = 0$ と $x = 1$ (と $x = -1$) が対応付けられるから、関数にならない。

答: $\mathcal{N}$ 上の関数にならない。理由: 上と同じ。

(c) x に対して x より大きくない (x と等しいか小さい) 最大の自然数を対応付ける。

答: $\mathcal{R}$ 上の関数にならない。理由: $x = -1$ に対して、それより大きくない自然数は存在しないから。(補足: この対応付けは、 $\mathcal{R}$ 上の部分関数にはなる。)

答: $\mathcal{N}$ 上の関数になる。理由: この対応付けは、恒等関数である。

答: $\mathcal{N}$ 上の単射かつ全射になる。理由: 恒等関数は単射かつ全射である。

問題 2 (像、合成、全射、単射)

$f: \mathcal{R} \rightarrow \mathcal{R}, g: \mathcal{R} \rightarrow \mathcal{R}$ となる関数 f, g を次のように定義する。

$$f(x) = x^2 - 4x + 5$$

$$g(x) = x^3 - 6x^2 + 12x - 7$$

また、 $A = \{x \in \mathcal{R} \mid 0 \leq x \leq 3\}$ とする。

(a) f による A の像 $f(A)$: 関数 f は $x = 2$ のとき極小値 1 を取る 2 次関数である。よって $0 \leq x \leq 3$ の範囲で、 $f(2) \leq f(x) \leq \max(f(0), f(3))$ である。結局、 $f(A) = \{x \in \mathcal{R} \mid 1 \leq x \leq 5\}$ となる。

g による A の像 $g(A)$: まず、関数 g の事を調べる。 $g'(x) = (x-2)^2$ であるので、 $x \in \mathcal{R}$ に対して $g'(x)$ は非負であり、 g は $\mathcal{R}$ 全域にわたって単調増加である。従って、 $0 \leq x \leq 3$ のとき $g(0) \leq g(x) \leq g(3)$ となり、結局、 $g(A) = \{x \in \mathcal{R} \mid -7 \leq x \leq 2\}$ となる。

- (b) 合成関数 $f \circ g$ と $g \circ f$ は定義されるか調べなさい。また、定義される場合、それらを具体的に求めよ。
 答. 合成関数 $f \circ g$ が定義されるのは、 g のコドメイン (行き先の集合) と、 f の定義域が一致するときである。この場合、両方とも $\mathcal{R}$ なので一致し、 $f \circ g$ は定義される。 $x \in \mathcal{R}$ に対して、

$$(f \circ g)(x) = f(g(x)) = x^6 - 12x^5 + 60x^4 - 162x^3 + 252x^2 - 216x + 82$$

$g \circ f$ も上と同様の理由で定義される。 $x \in \mathcal{R}$ に対して、

$$(g \circ f)(x) = g(f(x)) = x^6 - 12x^5 + 57x^4 - 136x^3 + 171x^2 - 108x + 28$$

[補足] こんな面倒な数式の計算はどうやってやるかといえば、「数式処理システム」と呼ばれるソフトウェアがある。これは、(近似計算を行う) 数値計算とは本質的に異なり、「数式 (や分数) をそのまま」処理してくれるものであ、現在広く使われている数式処理システムの 1 つは、Mathematica (マセマティカ) というシステムであり、情報科学類のコンピュータで使えるはずである。これはかなり便利で、ちょっとした数式処理は (上記のような合成関数や数式同士のかけ算などはもちろんのこと、微分や不定積分なども) あっという間に解いてくれる。亀山の隣の研究室の、井田先生と Marin 先生の研究グループは、これを使って、コンピュータ折り紙の計算をやっているそうである。

- (c) 集合 B を $\mathcal{R}$ の部分集合とする。 $f: \mathcal{R} \rightarrow B$ が全射となる集合 B を 1 つ求めよ。

答. $B = f(\mathcal{R})$ (f による $\mathcal{R}$ の像) である。

理由. f が $\mathcal{R}$ から B への関数となるためには、 $B \supset f(\mathcal{R})$ でなければいけない。(集合 B は f による行き先を全て含まないといけない。) また、 $B - f(\mathcal{R})$ が空集合でなければ、 f は $\mathcal{R}$ から B への全射とならない。よって、 $B = f(\mathcal{R})$ である。

なお、 $f(\mathcal{R})$ を具体的に計算すると、 $f(\mathcal{R}) = \{y \in \mathcal{R} \mid 1 \leq y\}$ となる。

- (d) 集合 C を $\mathcal{R}$ の部分集合とする。 $f: C \rightarrow \mathcal{R}$ が単射となる C を 1 つ求めよ。

答. このような C はたくさんあるが、たとえば、 $C = \{1\}$ や $C = \{x \in \mathcal{R} \mid x \geq 2\}$ とすればよい。

なお、極端な例として、 $C = \{\}$ (定義域が空集合である関数) も単射である。(定義域が空集合である関数というのは、かなり奇異な感じであるし、あまり使い道もないのであるが、一応関数の定義に合致しているので、関数である。また、単射の定義にも合致する。)

- (e) $f \circ h = g \circ h$ となる関数 $h: \mathcal{R} \rightarrow \mathcal{R}$ を 1 つ求めよ。

答. これはややこしい問題であり、魔法の公式はないが、地道に 1 つずつ考えていけば解ける。

まず、 $f \circ h = g \circ h$ なので、任意の実数 x に対して $f(h(x)) = g(h(x))$ である。 $y = h(x)$ と置くと、 y は実数であり、 $f(y) = g(y)$ となっている。そこで $g(y) - f(y) = 0$ という方程式を立てれば、その解が y (つまり $h(x)$) の候補となる。

$g(y) - f(y) = y^3 - 7y^2 + 16y - 12 = (y - 2)^2(y - 3)$ となり、 $y = 2, 3$ と求まる。よって、 h は、任意の実数を 2 か 3 に写せばよい。

逆に、任意の実数 x に対して、 $h(x) = 2$ または $h(x) = 3$ であれば、 $f \circ h = g \circ h$ となることが容易にわかる。

答. $h(x) = 2$ となる関数 h . (あるいは、 $h(x) = 3$ となる関数 h , あるいは、 $x \geq 0$ ならば $h(x) = 2$ で、 $x < 0$ ならば $h(x) = 3$ となる関数 h など)

問題 3 関数 f と g に対して、合成関数 $g \circ f$ が定義されたとする。このとき、「 f と g がともに全射であるならば $g \circ f$ が全射である」が成立するかどうか調べ、証明または反証しなさい。(成立するなら証明し、そうでなければ、反例を 1 つ挙げなさい。)

答. 正しいので証明する。 $f: A \rightarrow B, g: B \rightarrow C$ とする。このとき $g \circ f: A \rightarrow C$ である。 $g \circ f$ が全射であることを示すためには、任意の $z \in C$ に対して、ある $x \in A$ があって、 $(g \circ f)(x) = z$ となることを言えばよい。そこで、任意の $z \in C$ を取る。

g は全射なので、ある $y \in B$ があって、 $g(y) = z$ となる。さらに、 f は全射なので、ある $x \in A$ があって、 $f(x) = y$ となる。この x に対して、 $(g \circ f)(x) = g(f(x)) = g(y) = z$ となる。

よって、 $g \circ f$ は全射である。

(発展課題) 「単射」についても、上記と同じこと (f, g が単射ならば、 $g \circ f$ は単射、など) が成立する。証明は省略する。(昨年度の3章の演習問題で出題したいので、興味がある人は、その解答を見てほしい。)

「全単射」についても、上記と同じことが成立する。これは、「単射」と「全射」それぞれについて言えたので、合わせれば、すぐに言える。

問題4 (関数プログラミング)

関数 $f: (\mathcal{N} \times \mathcal{N}) \rightarrow \mathcal{N}$ は、任意の自然数 x, y に対して次の性質を満たしているとする。

$$\begin{aligned} f(\langle 0, y \rangle) &= y \\ f(\langle x+1, y \rangle) &= f(\langle x, x * y + y \rangle) \end{aligned}$$

問. $g(x) = f(\langle x, 1 \rangle)$ と定義するとき、 $g: \mathcal{N} \rightarrow \mathcal{N}$ となる関数であることを示しなさい。

答. 任意の自然数 x に対して、(f は関数であるので) $f(\langle x, 1 \rangle)$ の値は一意的に決まり、それは自然数である。従って、 $g(x) = f(\langle x, 1 \rangle)$ と定めれば、任意の自然数 x に対して $g(x)$ の値となる自然数が一意的に定まる。従って、 g は $\mathcal{N}$ から $\mathcal{N}$ への関数である。

問. また、任意の $x \in \mathcal{N}$ に対して $g(x+1) = g(x) * (x+1)$ が成立することを示しなさい。

答. この問題にきちんと答える (厳密な証明を与える) ためには、数学的帰納法が必要である。それは6章で学ぶ予定なので、ここでは深入りせず、「…」をつかった (厳密でない) 証明でよいことにする。

$$g(x+1) = f(x, x+1) = f(x-1, (x+1)x) = f(x-2, (x+1)x(x-1)) = \cdots = f(0, (x+1)x(x-1) \cdots 2 \cdot 1) = (x+1)x(x-1) \cdots 2 \cdot 1$$

となり、

$$g(x) = f(x-1, x) = f(x-2, x(x-1)) = f(x-3, x(x-1)(x-2)) = \cdots = f(0, x(x-1)(x-2) \cdots 2 \cdot 1) = x(x-1)(x-2) \cdots 2 \cdot 1$$

となる。よって、 $g(x+1) = (x+1) \cdot g(x)$ である。

問. さらに、 g はどのような意味を持つ関数が答えなさい。

答. 前問からわかるように、 $g(x) = x(x-1)(x-2) \cdots 2 \cdot 1$ となる。これは、いわゆる factorial (階乗) の関数である。

[補足] 本問では、 f という補助的な関数を定義したあとに、それを利用して (本当に作りたかった) 階乗の関数 g を作った。関数型プログラム言語では、このように、比較的小さな「関数」を定義し、それらを「合成」などで組み合わせて大きな関数を作っていく、というスタイルでプログラムを作成する。この点については、6章の「帰納」の章で、再び触れる予定である。「理屈はいいから、とにかく関数型プログラム言語を使ってみよう」という人は、OCaml 言語や、Haskell 言語などについてインターネットで調べるとよい (情報科学類の計算機でも使えるはずである。)