

『離散構造』(亀山)
Short Quiz とその解説

学籍番号: _____

氏 名: _____

Quiz. プログラム言語における型 (データ型, type) は、集合に対応する。

たとえば、C 言語の `int` と `float` という型は、それぞれ、整数の集合 $\mathbb{Z}$ 、実数の集合 $\mathbb{R}$ に対応しそうである。(この対応は近似に過ぎず、現実のプログラム言語では、32bit や 64bit といった有限長のデータで表現できるものだけが `int` や `float` の要素であるが、ここでは無視する。)

問題 1. C 言語の配列 (array) や構造体 (struct) というデータ型は、どのような集合に対応するだろうか？

```
int a[10];
struct student {
    int    student_id;
    char   name[100];
    float  height;
};
```

この問題では、型を集合と見なそう、という趣旨なので (その型の要素をどう操作するかは考えないので)、構造体のメンバの名前 (`height` など) は無視してよい。

解答. C 言語の `int a[10];` という宣言によって確保される配列 `a` は、整数型 (`int` 型) の要素を 10 個セットにしたデータ型である。INT を、「整数型の値をすべて集めた集合」とするとき、配列 `a` の型に対応する集合は、以下の直積集合である。

$$\text{INT} \times \text{INT} \times \text{INT} \times \text{INT} \times \text{INT} \times \text{INT} \times \text{INT} \times \text{INT} \times \text{INT} \times \text{INT}$$

これを、 INT^{10} と書くこともある。

上記の構造体 (struct 型) についても、`int` と `char` と `float` のそれぞれの型のデータを 1 セットにした、という点では配列と同じである。異なる点は、個々のデータが属する型が異なってもよい (配列の場合は、要素がすべて同じ型である) という点と、個々のデータを指すのに、配列では `a[3]` のようにインデックス (数字) を使ったが、構造体では、`student_id` のような名前を使うことである。ここでは、名前を使って個々のデータを取り出すという側面は無視するので、対応する集合は、以下の直積集合である。

$$\text{INT} \times \text{CHAR} \times \text{FLOAT}$$

なお、C 言語の構造体については、まだ知らない受講生が多いということで、この問題が不適切だった。将来 C 言語を学んだときに、このことを思いだしてほしい。

問題 2. 集合 P と Q の直和は、以下のように定義される。

$$P + Q = \{\langle 0, x \rangle \mid x \in P\} \cup \{\langle 1, y \rangle \mid y \in Q\}$$

ただし、 $\langle 0, x \rangle$ は 0 と x の対 (ペア) をあらわす。

さて、現実のプログラミング言語で、直和に相当するものはあるだろうか？皆さんの知っている言語で、そういう型を持つものがあれば答えなさい。

あるいは、現実のプログラミングで、「直和型」があると、どんな風に使えるだろうか？たとえば、1つの変数に、異なる型 P や Q の要素を代入したくなること等はあるだろうか？(型が違えば代入できないので、このようなことを C や Java でやるためには、 $P + Q$ 型が欲しくなるはずである。) (答えは、ウェブに掲載します。)

解答. C 言語の共用体 (union) は、宣言自体は、構造体 (struct) に似ているが、決定的な違いがあり、「int 型の要素か、または、float 型の要素のどちらか一方だけ」を格納できるデータ型である。(詳細は C 言語のマニュアルを参照してほしい。) しかし、よく考えると、「int 型と int 型の共用体」は、1つ目の int 型の要素だったのか、2つ目の共用体の要素だったのかの区別がつかなくなるので、直和集合というより単なる集合の和集合と見なせる。(C 言語で直和を定義したい場合は、struct と union を組みあわせる方法がある。)

OCaml などの関数型言語では variant 型とよばれるものがあり、これは直和にぴったり対応する。たとえば、筑波大学の構成員 (学生と教職員) データを 1 つにまとめたデータ型は、

```
type member =  
  | Student of int * string * string  
  | Staff    of int * string * string
```

のようになると、Student(201412345, "A 山 B 朗", "情報学群情報科学類") や、Staff(000012345, "C 谷 D 子", "システム情報系") といったデータを 1 つの型 (集合) のもとに格納することができる。この場合、仮に、学生と同姓同名で、「情報学群情報科学類」に所属する教職員がいたとしても、Staff(000012345, "A 山 B 朗", "情報学群情報科学類") となるだけであり、Staff であることがわかるので、「和集合」ではなく「直和集合」になっている。