

『離散構造』2-3 章

Short Quiz (亀山)

Short Quiz. プログラム言語において、「既に定義された型を使って、新しい型を作る」ものを型構成子 (type constructor) という。型構成子は、集合の演算で説明できるものが多い。

C 言語に備わっている型構成子のうち、構造体 (struct) は集合の言葉で言うと、どのような演算を表していると言えるだろうか？

Answer. まず、C 言語の構造体の例をあげる。

```
struct faculty { /* 教員を表すデータ構造 */
    int id; /* 職員番号 */
    int age; /* 年齢 */
    string name; /* 氏名, string は別途定義されているとする */
    float weight; /* 体重, こんなデータを大学が管理するとは思えないが */
};
```

```
struct faculty x; /* x の型が faculty 構造体型 */
x.id = 1234567;
x.age = 30; /* 嘘です */
x.name = "亀山 幸義";
x.weight = 50; /* これも嘘です */
```

この例では、「int 型のデータ 2 つと、string 型のデータ 1 つと、float 型のデータ 1 つ」をまとめて、1 つの faculty 型のデータとして扱っている。つまり、「型」を「集合」とおもえば、以下の等式が成立する。

$$\text{faculty} = \text{int} \times \text{int} \times \text{string} \times \text{float}$$

結局、「構造体を作る」というのは、「直積集合を作る」という操作に非常に類似していることがわかる。

補足. 「構造体」がぴったり「直積集合」というわけではない。上記の例でも、C 言語の faculty 型の定義と、対応する直積集合の定義とでは、情報量が違っている。すなわち、前者では、構造体の各要素には、id や age などの名前がついていて、その名前でアクセスできるのに対して、後者では、名前はついておらず、「組の 1 番目の要素」とか「組の 2 番目の要素」として指定するしかないのである。

プログラム言語では、各フィールドに名前がついている直積のことを「レコード」と呼ぶ。C 言語の構造体が大変便利であるのと同様、いくつかのプログラム言語では、レコード型が大変便利でよく使われる。(たとえば、ML 言語)

Short Quiz. もし知っている人がいれば、共用体 (union) についても考えよ。

Answer. C 言語の共用体は、メモリを節約したいシステムプログラミングなどでは使われるが、通常のプログラミングではあまり使われない。従って、当面は覚えなくても良いのだが、そういうものが C 言語にあるということぐらいは知っておくと良いだろう。

共用体は、union という名前があらわしているように、和集合に非常に類似しているものである。

共用体の例として、カーニハン/リッチーの C 言語の本では以下のものがあがっている。

```
union u_tag {  
    int ival;  
    float fval;  
    char *sval;  
};
```

これは、int の値か、float の値か、char へのポインタかどれか 1 つを記憶できるデータ型である。この種の「異なる型のデータを 1 箇所に格納したい」というのは、普通のプログラミングではあまり存在しないが、言語処理系を作っているときなどは、「読みこんだ単語が、整数か浮動小数点数か識別子か」によらずに、1 つのデータとして返したいことがある。ただし、共用体は使いにくく、上記の例では u_tag にはいつているものが ival だったのか fval だったのか sval だったのか、というのはデータだけを見てもわからないので、プログラマが自分でわかっていないといけない。

まともなプログラム言語なら、「和集合」のデータ構造なんかは用意せず、「直和集合」のデータ構造を用意するはずである。つまり、データそのもののの中に、ival であるか fval であるか sval であるかが書いてあるデータ構造の方がずっと使いやすい。OCaml などの関数型言語では、そのようなデータ構造を使って、わかりやすいプログラムが書ける。

なお、授業で「直和」を説明した後に、共用体の話をしたから、「共用体 = 直和」と思った人もいたようで、これは授業でちょっと説明を早くやりすぎた私の責任である。