Sets, continued

1 Repetitions

If A is a finite set with n elements, then any list of its elements that has length $n+1$ must have a repetition. This is called the *pigeonhole principle*: if there are $n+1$ pigeons and only n holes, then some hole must contain at least two pigeons.

To put this differently, any set A is either infinite or finite with size at least m iff there is a list of m elements with no repetition.

In the same way, a set A is infinite iff there is an infinite sequence of elements with no repetition.

2 Tagged elements

A special use of ordered pairs is to place a tag onto an element. Given two sets A and B, their *sum* (or tagged union) is given as follows:

$$A + B \stackrel{\text{def}}{=} \{(0,x) \mid x \in A\} \cup \{(1,y) \mid y \in B\}$$

For example, if A is the set of maths lecturers $\{\text{Paul}, \text{Rajesh}, \text{Usman}\}$ and B the set of logic lecturers $\{\text{Usman}, \text{Vincent}\}$, then we have

$$\begin{aligned} A \cup B &= \{\text{Paul}, \text{Rajesh}, \text{Usman}.\text{Vincent}\} \\ A + B &= \{(0, \text{Paul}), (0, \text{Rajesh}), (0, \text{Usman}) \\ &\quad (1, \text{Usman}), (1, \text{Vincent})\} \end{aligned}$$

Notice that, in $A + B$, each maths lecturer is tagged with 0 and each logic lecturer with 1, so there are two copies of Usman.

What about adding three sets A, B, C? We define

$$A + B + C \stackrel{\text{def}}{=} \{(0,x) \mid x \in A\} \cup \{(1,y) \mid y \in B\} \cup \{(2,z) \mid z \in C\}$$

In the same way, any list of sets has a sum. The empty set is the sum of the empty list.

Here's an application of set addition. Suppose my program either returns a natural number or crashes. We can represent my program's behaviour by an element of $\mathbb{N} + 1$. Returning 17 is represented by $(0, 17)$, and crashing by $(1, ())$.

Another application. Suppose that I have a file of customer records. Each entry represents either an adult with a name, address, and membership number, or a child with a name and an age. Let's say that String is the set of strings. The set of possible entries is

$$\text{String} \times \text{String} \times \mathbb{N} + \text{String} \times \mathbb{N}$$

In programming, a type that corresponds to a sum of sets is often called a "variant record type".