Free Types

Exercise 10.1 (Frame constructors) In a token ring, a frame may be either a data frame or a token frame.

- If it is a token frame, then it has an origin, a priority value and a reservation value.
- If it is a data frame, then it has an origin, a destination, a priority value, a reservation value, and a collection of data values.

There are no other kinds of frame, and no frame can be both a token frame and a data frame. Defining any basic types that you need, and using any set or type constructors that seem appropriate, show how the type of all frames could be modelled in our mathematical language. □

Exercise 10.2 (Counting and flattening) The set of all binary trees with natural number leaves is defined by

$$Tree ::= \mathsf{stalk} \mid \mathsf{leaf}\langle\langle \mathbb{N} \rangle\rangle \mid \mathsf{branch}\langle\langle Tree \times Tree \rangle\rangle$$

(a) Define a function *count* that returns the total number of leaves in an element of *Tree*.

(b) Define a function *flatten* that maps a tree to a sequence of natural numbers while preserving the multiplicity of each number that appears.

(c) Show that, for any tree t:

$$\#(flatten\, t) = count\, t$$

□