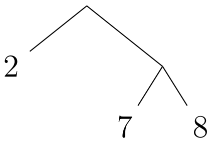
Forky definition:

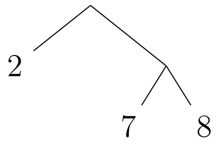
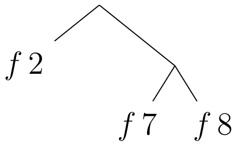
```
data Forky a = Tip a | Branch (Forky a) (Forky a)
```

In the pictures, `Branch` is shown as a branching point, and `Tip` is shown as a leaf with an element. For example



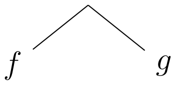
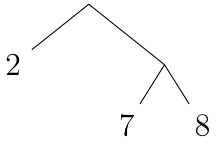
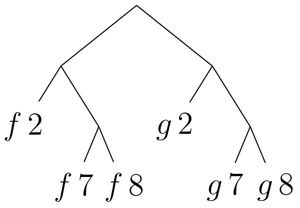
stands for `Branch (Tip 2) (Branch (Tip 7) (Tip 8))`.

fmap f t applies f to the elements in the tips. Example:

t	
fmap f t	

pure makes a singleton tree.

$tf \langle * \rangle t$ begins with the tree shape of tf, but then each of its *Tip f* is replaced by the result of *fmap f t*. Example:

tf	
t	
$tf \langle * \rangle t$	

You can think of *Forky* as a tree data structure, and $\langle * \rangle$ as extending one tree such that the leaves grow into clones of the other tree, and the new leaves have the respective functions applied to the respective arguments.

You can also think of *Forky* as modeling non-deterministic programs that, at each moment, can either split into two universes or finish with an answer; moreover, not only the multiple answers are recorded, but also the splitting histories of getting those answers are recorded as a tree. Then $\langle * \rangle$ runs two programs sequentially and records all ways of appending their histories.