CMSC-16100

Honors Introduction to Programming, I
Autumn Quarter, 2020

Lecture 11: Monads

Exercise 11.1 Show that `(>>= id) :: Monad m => m (m a) -> m a`, and therefore that `join = (>>= id)` is a plausible definition for `join` in a `Monad`. It is, in fact, the definition.

***Exercise 11.2** Consider the trivial two-element list `[(),()]`. Because this is an element of the list monad, we can include it on any line of list-defining `do` expression. Consider the two statements:

```
do
    [(),()]
    x <- [1,2,3]
    pure x
```

vs

```
do
    x <- [1,2,3]
    [(),()]
    pure x
```

These produce very different values. Explain the difference. The first expression can be re-written, using the techniques described above, into a particularly simple form that does not involve `do`. Do so.

Optionally: if you're feeling especially brave, the second form can be re-written in the same way, albeit not quite so simply. Do so.