Recap computational structure of possible

```
NonDetSvs : Set \rightarrow Set
DetSvs : Set \rightarrow Set
DetSvs X = X \rightarrow X
                                           NonDetSvs X = X \rightarrow List X
detFlow f zero = id
                                           nonDetFlow f zero = n_{List}
detFlow f (suc n) = detFlow f n \circ f
                                          nonDetFlow f (suc n) = f > list nonDetFlow f n
detTrif zero x =
                                           nonDetTri\ f\ zero \ \ x =
                                             fmap_{list}(x::)(n_{list}[])
  x :: []
detTrif(suc n) x =
                                           nonDetTrj\ f\ (suc\ n)\ x\ =
                                             fmap_{list}(x :: \_) ((f >>_{List} (nonDetTrj f n)) x)
  x :: detTrif n (f x)
```

Botta FPClimate 26 / 72

Recap computational structure of possible

```
\eta_{List} :
>=>1 ist
fmap_{List}: (A \rightarrow B) \rightarrow List A \rightarrow List B
\mu_{List}: List (List A) \rightarrow List A
\gg =_{List}:
\eta_{List} x = [x]
fmap_{List} = map
nonDetToDet f xs = xs \gg ist f
\forall (f : A \rightarrow B) (a : A) \rightarrow fmap_{list} f (\eta_{list} a) \equiv \eta_{list} (f a)
\forall (as : List A) \rightarrow \mu_{list} (\eta_{list} as) \equiv as
```

Botta FPClimate 28 / 72

Deterministic, non-deterministic, stochastic, etc. systems are instances of monadic systems

```
M : Set \rightarrow Set

fmap_{M} : \{A \ B : Set\} \rightarrow (A \rightarrow B) \rightarrow M \ A \rightarrow M \ B

\eta_{M} : \{A : Set\} \rightarrow A \rightarrow M \ A

\mu_{M} : \{A : Set\} \rightarrow M \ (M \ A) \rightarrow M \ A

\longrightarrow =_{M_{-}} : \{B \ C : Set\} \rightarrow M \ B \rightarrow (B \rightarrow M \ C) \rightarrow M \ C

\longrightarrow M_{-} : \{A \ B \ C : Set\} \rightarrow \{A \ A \ B \ C : Set\} \rightarrow \{A \ B \ C : Set\}
```

Botta FPClimate 29 / 72

All results extend to monadic systems systems

```
monFlow: \{X : Set\} \rightarrow MonSys X \rightarrow \mathbb{N} \rightarrow MonSys X
monFlow f zero = \eta_M
monFlow f (suc n) = f >=>_M monFlow f n
monTrj: \{X : Set\} \rightarrow MonSys X \rightarrow (n : \mathbb{N}) \rightarrow X \rightarrow M (Vec X (suc n))
monTri\ f\ zero \quad x = fmap_M(x ::_)(\eta_M[])
monTri\ f\ (suc\ n)\ x\ =\ fmap_M\ (x::_)\ (f\ x\ \gg=_M\ (monTrj\ f\ n))
detToMon: \{X : Set\} \rightarrow DetSys X \rightarrow MonSys X
detToMon f = \eta_M \circ f
```

Botta FPClimate 30 / 72

```
monToDet : \{X : Set\} \rightarrow MonSys X \rightarrow DetSys (M X)
monToDet f mx = mx \gg =_M f
Det \equiv Mon : \{X : Set\} \rightarrow (f : DetSys X) \rightarrow (n : \mathbb{N}) \rightarrow (x : X) \rightarrow (f : DetSys X) \rightarrow (f : Det
                                                                                                                                                                                                                                         n_M (detFlow f n x) \equiv monFlow (detToMon f) n x
Mon \equiv Det : \{X : Set\} \rightarrow (f : MonSys X) \rightarrow (n : \mathbb{N}) \rightarrow (mx : M X) \rightarrow (f : MonSys X) \rightarrow (f : 
                                                                                                                                                                                                                                             monToDet (monFlow f n) mx \equiv detFlow (monToDet f) n mx
```

And more . . .

Botta FPClimate 31/72

The computational structure of *possible*: Monadic dynamical systems

Botta FPClimate 32 / 72

The computational structure of possible: Monadic dynamical systems

- The bottom line is that, when the functor F is also a monad, possible s: F Evolution can be defined in terms of computations like monFlow, monTri and their combinations
- Example 1: Evolution = State, possible = monFlow next 5
- Example 2: Evolution = Vec State 5, possible = monTrj next 4
- Example 3: Evolution = State², possible = fmap_M (λ s' \rightarrow (s , s')) (monFlow next 5 s)

FPClimate 33 / 72 Botta

Botta FPClimate 34 / 72

• We have seen that the monadic operations fulfil certain equations, for example

$$\forall (f:A \rightarrow B)(a:A) \rightarrow fmap_{List} f(\eta_{List} a) \equiv \eta_{List} (f a)$$

• For arbitrary A, B, C : Set one has

$$\forall \ (f:A\rightarrow FB)\rightarrow \qquad \qquad (_\gg=f) \stackrel{.}{=} \mu \circ \mathit{fmap} \ f$$

$$\forall \ (f:A\rightarrow FB)\rightarrow (g:B\rightarrow FC)\rightarrow f \gg g \stackrel{.}{=} \mu \circ \mathit{fmap} \ g \circ f$$

$$\mu \circ \eta \stackrel{.}{=} \mathit{id}$$

$$\mu \circ \mathit{fmap} \ \eta \stackrel{.}{=} \mathit{id}$$

$$\mu \circ \mu \stackrel{.}{=} \mu \circ \mathit{fmap} \ \mu$$

$$\forall \ (f:A\rightarrow B)\rightarrow \mathit{fmap} \ f \circ \eta \stackrel{.}{=} \eta \circ f$$

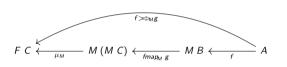
$$\forall \ (f:A\rightarrow B)\rightarrow \mathit{fmap} \ f \circ \mu \stackrel{.}{=} \mu \circ \mathit{fmap} \ (\mathit{fmap} \ f)$$

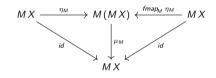
• In this specification, $f \doteq g$ means that f is extensionally equal to g:

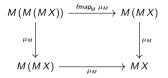
$$f \doteq g = (x : dom f) \rightarrow f x \equiv g x$$

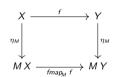
Botta FPClimate 35 / 72

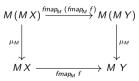
• Monadic laws are best understood diagrammatically











Botta FPClimate 36 / 72

Exercise 4.14

Postulate the monadic laws in Agda.

Exercise 4.15

Using the postulated monadic laws, prove $Det \equiv Mon$. (It should be very similar to the earlier proof of $Det \equiv NonDet$, but now for an arbitrary monad.)

Botta FPClimate 37 / 72