

Kahn

Christophe Saad

April 10, 2020

1 Sequential

The implementation of the sequential part is mainly taken from the paper *A poor man's concurrency monad*. We consider the process as a monad transformer which takes an action (which contains the actual computation) and links it to a continuation. Its type is

```
type 'a process = ('a -> action) -> action
```

The type action is

```
type action = Atom of (unit -> action) | Fork of action * action | Stop
```

We use **Fork** to instantiate new processes and **Atom** to represent a computation. In the **run** function, we recreate a pipeline containing the first **Fork** action to execute. Each time a **Fork** is executed, we push the two actions in the pipeline. When an **Atom** is read, we execute its computation and store its continuation back in the pipeline. This procedure ends when all the continuations are **Stop**.

2 Mandelbrot

- $P_c(z) = z^2 + c$
- $c \in \mathcal{M} \iff |\mathcal{P}_c^n(0)| \leq 2 \quad \forall n \geq 0$
- Goes to infinity if ever crosses 2
- $[-2, \frac{1}{4}]$ intersection of \mathcal{M} with the real axis