Rapport d'echange : Ecole Normale Superieure

Christophe Saad

April 9, 2020

The implentation of the sequencial part is manly taken from the paper A poor mans concurrency monad. We consider the process as a monad transformer which takes an action (which contains the actual computation) and link it 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 instanciate 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 his continuation back in the pipeline. This procedures ends when all the continuations are `Stop`.