Efficient coinductives through state-machine corecursors

Supervisor: Alex Keizer

Marking supervisor: Jamie Vicary

Director of studies: Russell Moore and Paula Buttery

Word count: 358

```
1         42 text files.
2         42 unique files.
3          0 files ignored.
4
5   github.com/AlDanial/cloc v 2.06  T=0.07 s (638.9 files/s, 94671.7 lines/s)
6   -------------------------------------------------------------------------------
7   Language                     files          blank        comment           code
8   -------------------------------------------------------------------------------
9   Lean                            42            954            148           5122
10  -------------------------------------------------------------------------------
11  SUM:                            42            954            148           5122
12  -------------------------------------------------------------------------------
```

1 What has been done

I have implemented a generic implementation of the `M`-type, which is asymptotically optimal. I proved the equivlence between this and the source implementation, and used this to instantiate the ABI type (a more fleshed out version of shrink). This is as powerful as the generalization of the `M`-type I partially upstreamed.

Using this I implemented the non-termination monad, and made tools to make this usable. I also subsequently implemented major parts of (Xia et al., 2019).

I have also implemented something along the line of a futumorphism, which I refer to as a `DeepThunk`. This transform is what I am currently working on providing a way to transform any terminating function into a productive one. Formalization of a step lemma of futumorphisms is highly technical and I am currently marking it as a goal.

I am also taking algebraic methods which will be a great test of my dissertation in practice, for this I will at least implement a hylomorphism and probably other combinators as seen in (Yang & Wu, 2022).

2 Schedule

From my original proposal I have completed:

1. Variable universe `Ms`
2. `Stream` special example

3. Multivariate `M`
4. `Cofix`
5. NTMonad

Completing my core. I decided the implementing the equivlence for Univariate `M` was superfluous. The reason for this is I decided to tackle the more general case first, rather than trying this easier case I had in case for derisking.

I also completed the extention in my original proposal (shrink), along with one more.

1. Shrinking the representations
2. Implementing interaction trees.

This puts me around 8-weeks ahead of schedule.

3 Difficulties

My original supervisor took a term out, this was expected and handled by the precautions established at the start of the project.

I also lost my grandfather and aunt, this delayed my work as I had to go home and grieve.

Bibliography

Xia, L.-y., Zakowski, Y., He, P., Hur, C.-K., Malecha, G., Pierce, B. C., & Zdancewic, S. (2019). Interaction trees: representing recursive and impure programs in Coq. *Proc. ACM Program. Lang.*, *4*(POPL). https://doi.org/10.1145/3371119

Yang, Z., & Wu, N. (2022). Fantastic Morphisms and Where to Find Them. In E. Komendantskaya (Ed.), *Mathematics of Program Construction: Mathematics of Program Construction*.