Exam report for COMP4181/9181 (13s2)

z3416506

Critical assessment of Paper 1.

Problem that the paper tries to address

Jan Bracker and Andy Gill's technical paper, *Sunroof: A Monadic DSL to Generate JavaScript*, attempt to generate JavaScript programs through the domain specific language, *Sunroof*, which is embedded in Haskell. They discuss the usefulness of JavaScript (e.g. graphical canvases, event handling, and first-class functions), but also note that it lacks some desirable features, such as Haskell's static typing.

Bracker and Gill propose Sunroof as an alternative to JavaScript, since Sunroof is able to introduce many of Haskell's features to programmers that JavaScript is unable to natively facilitate (e.g. a threading model, a static type checker, etc.). Since Haskell has an extremely powerful type system, JavaScript programs that are generated through Sunroof are more likely to be correct than if the JavaScript was handwritten.

Sunroof is implemented through a monad similar to the `IO` monad found in Haskell, but uses an extra argument to determine which threading model is to be used. Unlike native, handwirtten JavaScript, Sunroof is able to provide concurrent JavaScript, since it is embedded in Haskell. This is an important step up from handwritten JavaScript, since parallel computations are increasingly becoming important.

Coverage of related work

The authors claim that their work differs from previous research since the previously published papers do not attempt to directly bridge Haskell and JavaScript. Of the thirty references to other works made in this paper, seventeen of these references are explicitly considered to be related in some way to Sunroof. The authors note the similarities of related work, but do not go into great detail about any of them. This is not necessarily bad; there are too many to go into great detail of each, and their level of definition is more than enough to encourage interested readers (with sufficient time) to investigate the related works.

The remaining works do not appear to be directly related to research associated with Sunroof; they are more related to Haskell features used to implement Sunroof. Consider '*Our example type* `JSString` *has a* `Monoid` *and an* `IsString` *instance that are not provided for other wrappers, e.g.* `JSBool` *or* `JSNumber`*. This approach was first introduced by Svenningsson [29].*' Svenningson and Axelsson had done previous research regarding shallow and deep embedding in reference 29, and Bracker and Gill were able to capitalise on this. They provided reference to a highly detailed technical paper written by Svenningsson and Axelsson regarding the topic[1], which encourages further research should the reader wish to learn more about Sunroof's implementation.

Originality and technical soundness of the underlying ideas

The idea of generating JavaScript through Haskell is not original, as it is mentioned in the Related Work section that *Fay* compiles subsets of Haskell to JavaScript. Bracker and Gill are quick to point out that a direct connection between Haskell and JavaScript is most likely novel. Furthermore, it is their belief that Sunroof is the only EDSL that generates JavaScript inside Haskell that is type-safe. No evidence has been presented to suggest otherwise at the time of writing this analysis.

This doesn't make the process novel. *Accelerate* generates *CUDA* in a similar manner[1], and thus it is important for this paper to be augmented with some sort of technical background. This is somewhat lacking, since a Sunroof codebase isn't supplied; only a series of code snippets have been made available in the paper itself.

There are two completely original aspects to this

[1] The author of this critical analysis didn't have time to properly read this paper, but did read through enough of it to get the gist of what Bracker and Gill were alluding to.

paper[2]:

- Sunroof implements threading models for JavaScript, and,
- The Sunroof server '*provides the infrastructure to communicate with the currently calling website through the Kansas cornet [16] push mechanism [20].*'

Evaluation of the presented approach

References

1. AccelerateHS, https://github.com/AccelerateHS/accelerate

End of critical assessment of Paper 1.

[2] To the author of this critical analysis' knowledge.

Critical assessment of Paper 2.

Problem that the paper tries to address

Coverage of related work

Originality and technical soundness of the underlying ideas

Evaluation of the presented approach

End of critical assessment of Paper 2.

This page is intended to be blank.

The purpose of this page is to prevent accidental scrolling and revealing the identity of the student.

Full name: **Di Bella**, Christopher James
Student number: z3416506

By submitting this report for assessment as the exam component of COMP4181/9181 (13s2), I declare that this submission is my own work, and I have not received any help whatsoever.