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 because:

- Other works do not attempt to create a direct connection between Haskell and JavaScript continuations,
- The Sunroof server provides support for communication with websites requesting code snippets via the *Kansas comet push mechanism*, and,
- Sunroof is the only EDSL to support JavaScript generation inside type-safe Haskell.

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 point out that 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].

The only completely novel proposal identified is the Sunroof server previously mentioned. Unfortunately, this section is done little justice, as the content spans for only half a page, including the provided code. While the section is short, the idea presented is really innovative and lightweight in contrast to other web frameworks such as *HAppS*, *Snap*, and *Yesod*, and does not require much explanation; even Haskell novices should be able to infer the code necessary to achieve what is being communicated here.

The paper presents a technical overview of the topic from too high a level. The problem with this is that they do not provide much depth or analysis, and often do not justify why they made such decisions. It would be relatively easy to reproduce the calculator discussed in the paper, and given that the authors have specified how many lines of code are necessary to complete the task, it is possible to evaluate their accuracy, provided that the authors have a sample calculator to test against. Unfortunately, this does not seem the case, as neither mention of the calculator's specification, nor an external reference to such a specification is made in this paper. Additionally, the table provided is not informative, since it does not contrast how many lines of code would be necessary in handwritten JavaScript; although it is possible that the 'Percentage' column alludes to this, insufficient information in the column's heading is provided to prove this.

Evaluation of the presented approach

Overall, the paper is structured well. However, there are a few key things to take note of. Their analysis is not quite as deep as some, and this paper certainly does not make any attempt to motivate using Sunroof over other EDSLs until

[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.

the very end. Additionally, they do not provide any hard evidence to prove that Sunroof is indeed a superior EDSL. This is crucial for implementors as they will need to be convinced that Sunroof is worth their time (and possibly money) learning; this may not be achieved, since there are neither code snippet comparisons, nor anything to quantify that the Sunroof server model is superior to other web frameworks in practice.

Many of the ideas communicated are brilliant and a sense of closure is almost always apparent; however, without the any of the above, it is difficult to believe that this paper has a convincing tone. Perhaps if the paper (or a subsequent paper published by Bracker and Gill) was able to meet any one of the above criteria, it would be able to provide more insight into why Sunroof is a desirable EDSL.

References

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

End of critical assessment of Paper 1.

Critical assessment of Paper 2.

Problem that the paper tries to address

Mio: A High-Performance Multicore IO Manager for GHC, written by Andreas Voellmy, Junchang Wangm, Paul Hudak, and Kazuhiko Yamamoto, addresses a very real concurrency problem. They argue that Haskell threads are key to high-performance, concurrent programs. They suggest that naive implementations are likely to use native threads, which involve expensive context switching. Haskell threads are lightweight, and do not require the operating system to perform any expensive context switching. The authors state that Haskell threads are likely to promote simple threaded models, and discourage event-driven programming. They convincingly present the idea that event-driven programs are logically complex, and that network server logic is better understood as a single execution thread.

Although GHC provides support for Haskell threads, its runtime system doesn't natively scale well on multicore processors, and so network applications that try to take advantage of lightweight Haskell threads will perform poorly. The problems that this paper address are:

- Identifying why GHC's RTS performance doesn't natively scale on multicore processors.
-

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.