AFP Project: Implementing an efficient version of Data.Map in Agda

Wessel Custers
Utrecht University, Netherlands

Sebastiaan Koppen
Utrecht University, Netherlands

Daan van Westerlaak
Utrecht University, Netherlands

1 Context

Agda is a dependently typed programming language and proof assistant, which is mostly used for research purposes. It has relatively few libraries, making it a less attractive language for development. We believe that an implementation of the `Data.Map` module from Haskell's `containers` package would be a good addition for the language.

`Data.Map` is a finite map of key-value pairs. As can be read in the documentation, the implementation of `Data.Map` is based on size balanced trees [1, 2].

2 Motivation

Neither of the members of our team has experience with Agda. Based on what we do know about the language, we think this project should be a good opportunity to learn about Agda and what distinguishes it from Haskell. We purposefully chose a topic that has clear bounds over a more creative one. This way, we hope that the result of our project will be a library that is actually useful to others.

3 Deliverables

- A pure-Agda implementation of `Data.Map`.
- Benchmarks for the implementation.
- *(If possible):* Extensions on `Data.Map` (e.g. optimizations relating to underlying types or extra functions).

4 Planning

The total time for the project is approximately 6 - 7 weeks.

Week	Content
1	Reading, understanding `Data.Map`
2	
3	
4	
5	
6	

Table 1 Schedule

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

References

1 Adams, S. Functional pearls efficient sets—a balancing act. *Journal of functional programming 3*, 4 (1993), 553–561.

2 Nievergelt, J., and Reingold, E. M. Binary search trees of bounded balance. In *Proceedings of the fourth annual ACM symposium on Theory of computing* (1972), pp. 137–142.