

1 Agda

Report by:	Andreas Abel
Participants:	Nils Anders Danielsson, Ulf Norell, Makoto Takeyama, Stevan Andjelkovic, Jean-Philippe Bernardy, James Chapman, Dominique Devriese, Peter Divianszki, Fredrik Nordvall Forsberg, Olle Fredriksson, Daniel Gustafsson, Alan Jeffrey, Fredrik Lindblad, Guilhem Moulin, Nicolas Pouillard, Andrs Sicard-Ramrez and many others
Status:	actively developed

Agda is a dependently typed functional programming language (developed using Haskell). A central feature of Agda is inductive families, i.e., GADTs which can be indexed by *values* and not just types. The language also supports coinductive types, parameterized modules, and mixfix operators, and comes with an *interactive* interface—the type checker can assist you in the development of your code.

A lot of work remains in order for Agda to become a full-fledged programming language (good libraries, mature compilers, documentation, etc.), but already in its current state it can provide lots of fun as a platform for experiments in dependently typed programming.

Since the release of Agda 2.3.2 in November 2012 the following has happened in the Agda project and community:

- Ulf Norell gave a keynote speech at ICFP 2013 on dependently typed programming in Agda.
- Agda has attracted new users, the traffic on the mailing list (and bug tracker) is increasing.
- About 100 bugs of Agda 2.3.2 have been fixed; and small enhancements improve the usability.
- Copatterns are being added to Agda as a new way to define record and coinductive values.

Release of Agda 2.3.4 is planned to happen soon after the one of GHC 7.8.

Further reading

The Agda Wiki: <http://wiki.portal.chalmers.se/agda/>