Ten Years of Agda2

Andreas Abel\textsuperscript{1}

\textsuperscript{1}Department of Computer Science and Engineering
Chalmers and Gothenburg University, Sweden

21st Agda Implementor’s Meeting (AIM XXI)
Teknikparken, Chalmers, Gothenburg, Sweden
3 June 2015
## Agda meetings (AIMs)

<table>
<thead>
<tr>
<th>Month</th>
<th>Location</th>
<th>Year</th>
<th>Season</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apr</td>
<td>Osaka (II)</td>
<td>2004</td>
<td>Oct</td>
<td>Gothenburg (I)</td>
</tr>
<tr>
<td>Spring</td>
<td>Osaka (VI)</td>
<td>2005</td>
<td>Sep</td>
<td>Gothenburg (III)</td>
</tr>
<tr>
<td>May</td>
<td>Gothenburg (VI)</td>
<td>2006</td>
<td>fall</td>
<td>Gothenburg (V)</td>
</tr>
<tr>
<td>Jun</td>
<td>Gothenburg (VIII)</td>
<td>2007</td>
<td>Nov</td>
<td>Osaka (VII)</td>
</tr>
<tr>
<td>Mar</td>
<td>Awaji (XI)</td>
<td>2008</td>
<td>Nov</td>
<td>Sendai (IX)</td>
</tr>
<tr>
<td>Apr</td>
<td>Gothenburg (XIII)</td>
<td>2009</td>
<td>Sep</td>
<td>Gothenburg (X)</td>
</tr>
<tr>
<td>Feb</td>
<td>Fischbachau (XV)</td>
<td>2010</td>
<td>Sep</td>
<td>Nottingham (XII)</td>
</tr>
<tr>
<td>May</td>
<td>Shonan (XVII)</td>
<td>2011</td>
<td>Sep</td>
<td>Shonan (XIV)</td>
</tr>
<tr>
<td>May</td>
<td>Paris (XIX)</td>
<td>2012</td>
<td>Oct</td>
<td>Copenhagen (XVI)</td>
</tr>
<tr>
<td>May</td>
<td>Gothenburg (XVIII)</td>
<td>2013</td>
<td>Sep</td>
<td>Gothenburg (XVIII)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2014</td>
<td>Oct</td>
<td>Tallinn (XX)</td>
</tr>
</tbody>
</table>
Agda 2: Beginnings

2004 Prototype *AgdaLight* started in the Cover project: Thierry, Ulf, and me.

2005 Nisse formalizes LF in AgdaLight.

2005 September: Jeff Polakow and Ulf discuss the design of Agda 2. Coding starts.

2006 Agda 2 is implemented.

2007 Release of Agda 2.0.0 after AIM6 (May 2007).

2007 Sep: Agda Language Review.
Agda 2.0.0 Feature list

- Inductive families.
- The `with`-construct (magic from 2.1.0).
- Records.
- Termination checker for structural recursion.
- The Alonzo compiler.
- Fancily coloured Emacs mode.
Agda 2.2.0 (Mar 2009) – 2.2.4

- Coverage checker and interactive case splitting (C-c C-c)
- codata
- Sized types
- MAlonzo compiler and Haskell FFI
- Unicode syntax and Agda input mode
- HTML rendering
- Standard library
- Windows installer and hackage
Agda 2.2.6 (Dec 2009)

- Universe polymorphism: AIM 8 (May 2008) – AIM 10 (Sep 2009)
- Records: constructors and hidden fields
- rewrite
- trustMe
- intro command
- Relocatable interface files containing highlighting info
- Retired: Agate and Alonzo compilers
Agda 2.2.8 (Sep 2010, post AIM 12)

- Record pattern matching
- Non-dependent irrelevance
- syntax
- Musical coinduction
- Reflection: quote, quoteGoal
- Auto (Agsy) (C-c C-a)
Agda 2.2.10 (Feb 2011)

- --without-K without foundation
- Epic compiler
Mutual definitions without *mutual* keyword
Pattern matching lambdas (extended lambdas)
Record update
Unification improvements following Abel/Pientka (TLCA 2011)
Instance arguments
Dependent irrelevance
Reflection: unquote, quoteTerm
JavaScript compiler
Agda 2.3.2 (Nov 2012)

- Let record patterns
- Pattern synonyms
- Qualified mixfix operators
- Anonymous parameterized modules
- Smarter coverage checker
- Interactive syntax highlighting
- LaTeX backend
Agda 2.4.0 (June 2014)

- Development now using git on github.com
- Levels defined in Agda.Primitive
- Copatterns
- Anonymous top level module
- Lets in telescopes
- Proper --without-K
- Generate helper function (C-c C-h)
- Improvements, improvements, ...
Agda 2.4.2 (August 2014)

- Recursive instance search
- tactic keyword
- Unquoting declarations
Agda: Outside View

- A language to formalize math and computer science
- A language to write papers (ca. 100 on the wiki)
- A language for teaching (15 courses on the wiki)
- A brand: *Coq and Agda*
- A user community
  - Martin Escardo, Wolfram Kahl, Dan Licata, Serge Mechveliani, Andy Pitts...
Agda: Inside View

- 3 main developers (part-time)
- 1 packaging, integration, and release engineer (part-time)
- A dozen contributors (sporadically active)
- A long tail of single-patch submitters
- 93,000 loc (3.4MB) [2014: 70,000 loc]
- 1528 issues on the bug tracker (249 open) [2014: 1076 (165)]

2007-2010 100 bugs/year
2010-2013 200 bugs/year
2013- 300 bugs/year ?
Agda: Where do the 93kloc come from?

<table>
<thead>
<tr>
<th>Component</th>
<th>loc ('15)</th>
<th>loc ('14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility functions</td>
<td>8000</td>
<td>4100</td>
</tr>
<tr>
<td>Syntax (parsing, printing, scope checking)</td>
<td>19000</td>
<td>16000</td>
</tr>
<tr>
<td>Type checker (+ eval, coverage)</td>
<td>39000</td>
<td>30000</td>
</tr>
<tr>
<td>Termination checker</td>
<td>5800</td>
<td>4600</td>
</tr>
<tr>
<td>Interaction (+ highlighting, imports)</td>
<td>7400</td>
<td>6600</td>
</tr>
<tr>
<td>Agsy</td>
<td>4200</td>
<td>4100</td>
</tr>
<tr>
<td>Compiler</td>
<td>8000</td>
<td>5200</td>
</tr>
</tbody>
</table>
Bug fixing is sometimes like Whack-A-Mole
Long on the Wish List

- User manual
- Packaging
- Type classes (WIP)
- Universe cumulativity
- Reflection/tactics (WIP)
- Efficient type-checking
- Usable compiler (WIP)
Core Language / Internal Syntax

- Sharing
- Independent checking
- Termination certificates
- Shared optimizations/transformations used by compiler backends
Research topics

- Equality (HoTT, OTT)
- Parametricity/colors
- Sized dependent types
- Proof-instance search and unification
- Foundation for hidden/named arguments (Msc: Marcus, Jesper)
- Telescopes/Σ-types at framework level
- Printing