

Twenty Years of Agda Implementor's Meetings

Andreas Abel¹

¹Department of Computer Science and Engineering
Chalmers and Gothenburg University, Sweden

39th Agda Implementor's Meeting (AIM XXXIX)
OOTO Cafe, Chalmers, Gothenburg, Sweden
27 November 2024

Agda meetings (AIMs) 2004-2014

		2004	Oct		Gothenburg (I)	
Apr		Osaka (II)	2005	Sep		Gothenburg (III)
spring		Osaka (IV)	2006	fall		Gothenburg (V)
May		Gothenburg (VI)	2007	Nov		Osaka (VII)
Jun		Gothenburg (VIII)	2008	Nov		Sendai (IX)
		2009	Sep		Gothenburg (X)	
Mar		Awaji (XI)	2010	Sep		Nottingham (XII)
Apr		Gothenburg (XIII)	2011	Sep		Shonan (XIV)
Feb		Fischbachau (XV)	2012	Oct		Copenhagen (XVI)
May		Shonan (XVII)	2013	Sep		Gothenburg (XVIII)
May		Paris (XIX)	2014	Oct		Tallinn (XX)

Agda meetings (AIMs) 2015-2024

Jun		Gothenburg (XXI)	2015	Sep		Leuven (XXII)
Apr		Glasgow (XXIII)	2016	Oct		Utrecht (XXIV)
May		Gothenburg (XXV)	2017	Feb'18		Budapest (XXVI)
Jun		Gothenburg (XXVII)	2018	Oct		Nottingham (XXVIII)
Mar		Tokyo (XXIX)	2019	Sep		Munich (XXX)
May+		online (XXXII)	2020	Oct		online (XXXIII)
Jun		online (XXXIV)	2021			
May		online (XXXV)	2022	Nov		Edinburgh (XXXI)
May		Delft (XXXVI)	2023	Nov		Taipei (XXXVII)
May		Swansea (XXXVIII)	2024	Nov		Gothenburg (XXIX)

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

Agda 2.3.0 (Nov 2011, post AIM 13/14)

- 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 (Alan Jeffrey[†])

Agda 2.3.2 (Nov 2012)

- Let record patterns
- Pattern synonyms
- Qualified mixfix operators
- Anonymous parameterized modules
- Smarter coverage checker
- Interactive syntax highlighting
- $\text{\LaTeX}$ 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 2.5.1 (Apr 2016)

- User manual <https://agda.readthedocs.io/>
- Agda.Builtin modules
- Library management (.agdalib files)
- GHC backend: bind data types, foreign code, copatterns
- UHC backend
- --exact-split
- --sharing experiment begins
- DISPLAY pragma
- operator sections
- Idris-style reflection (TC monad)
- overloaded literals (FROMNAT/NEG/STRING)
- no-eta-equality, records from modules
- instance constructors and fields
- New unifier --without-K
- agda-ghc-names

Agda 2.5.2 (Dec 2016)

- Literate Agda: reStructuredText
- Support stack
- Dot patterns optional
- Module parameter refinement by pattern matching
- @-patterns
- idiom brackets
- lambda-where
- overloaded and postfix projections
- POLARITY pragma
- Improved floats
- Reflection: no longer normalize
- overlappable instance fields
- Rewritten JS backend

Agda 2.5.3 (Sep 2017)

- Literate Agda: Markdown
- INJECTIVE pragma
- COMPILE and FOREIGN pragma
- HTML backend: symbolic anchors
- QuickLaTeX backend
- `--safe` as pragma

Agda 2.5.4 (Jun 2018)

- Call-by-need reduction machine
- Automatic inlining
- do-notation
- overloaded pattern synonyms
- forced constructor patterns
- Word64
- `--caching`
- end of `--sharing`
- `pragma WARNING_ON_USAGE`
- token-based highlighting

Agda 2.6.0 (Apr 2019)

- Cubical Agda
- variable generalization
- Prop
- non-overlapping instance search

Agda 2.6.1 (Mar 2020)

- Agda logo
- Confluence checker
- fractional precedence levels
- irrefutable with
- record patterns in telescopes
- flat modality
- `--cumulativity`
- tactic arguments

Agda 2.6.2 (Jun 2021)

- Improved erasure
- Guarded Cubical Agda
- Renaming Set to Type possible
- Native inspect idiom (`with`)
- `execTC`

Agda 2.6.3 (Jan 2023)

- Erased Cubical Agda
- `--cubical-compatible`
- `declareData...`

Agda 2.6.4 (Oct 2023)

- Cubical boundaries displayed interactively
- opaque
- Consolidation of `--erasure`
- Reflection: `pragmaCompile...`

Agda 2.7.0 (Aug 2024)

- Mimer replacing Agsy
- `using` (let binding in with)
- Indexed instance search

Agda: Outside View (2015)

- 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 (2015)

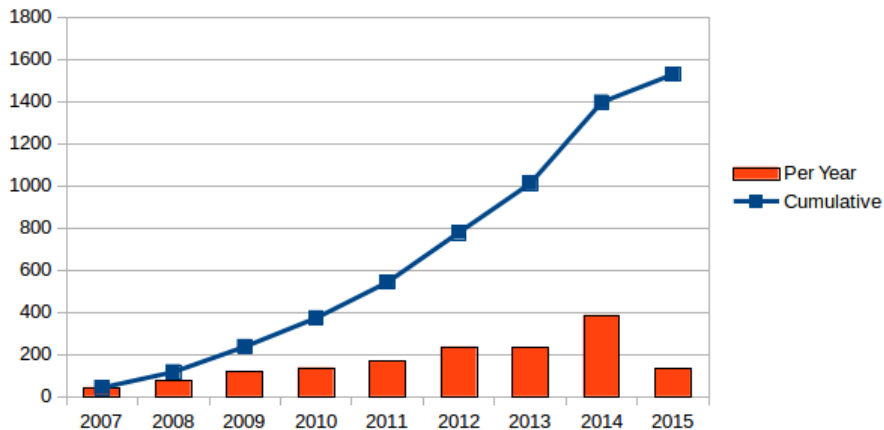
- 3 main developers (part-time)
- 1 packaging, integration, and release engineer (part-time)
- a dozen contributors (sporadically active)
- a long tail of single-patch submitters
- 93.000 loc (3.4MB) [2014: 70.000 loc]
- 1528 issues on the bug tracker (249 open) [2014: 1076 (165)]

2007-2010	100 bugs/year
2010-2013	200 bugs/year
2013-	300 bugs/year ?

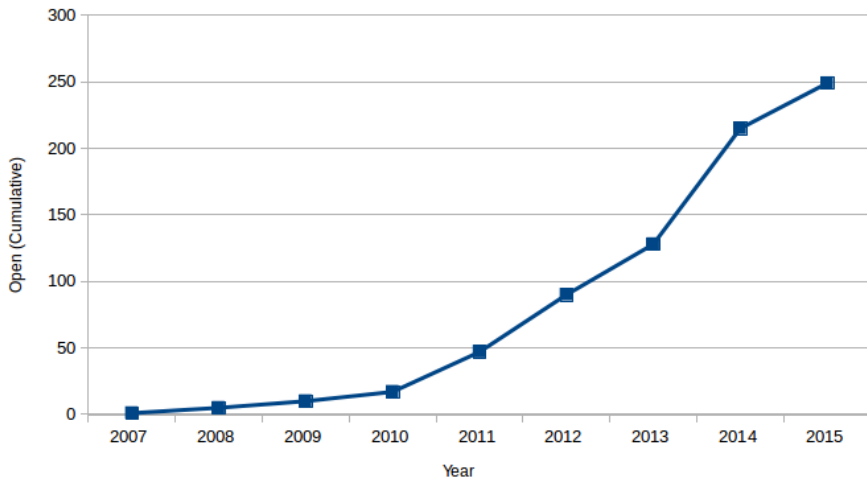
Agda: Where do the loc come from?

Component	loc ('15)	loc ('24)
Utility functions	8000	11000
Syntax (parsing, printing, scope checking)	19000	36000
Type checker (+ eval, coverage)	39000	79000
Termination checker	5800	3600
Interaction (+ highlight (LaTeX...), imports)	7400	16000
Agsy / Mimer	4200	1800
Compiler	8000	7900
Tests		136000

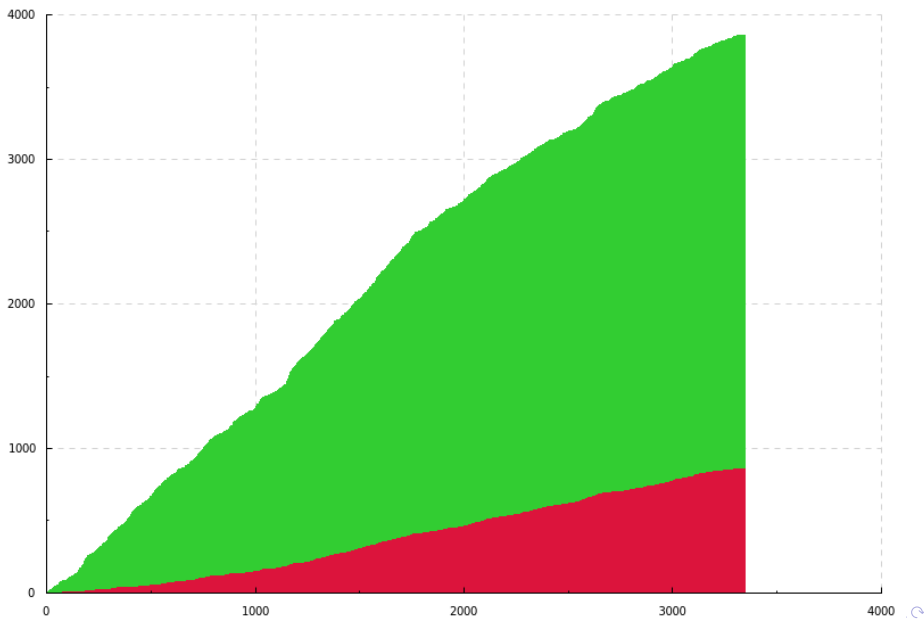
All Agda Issues



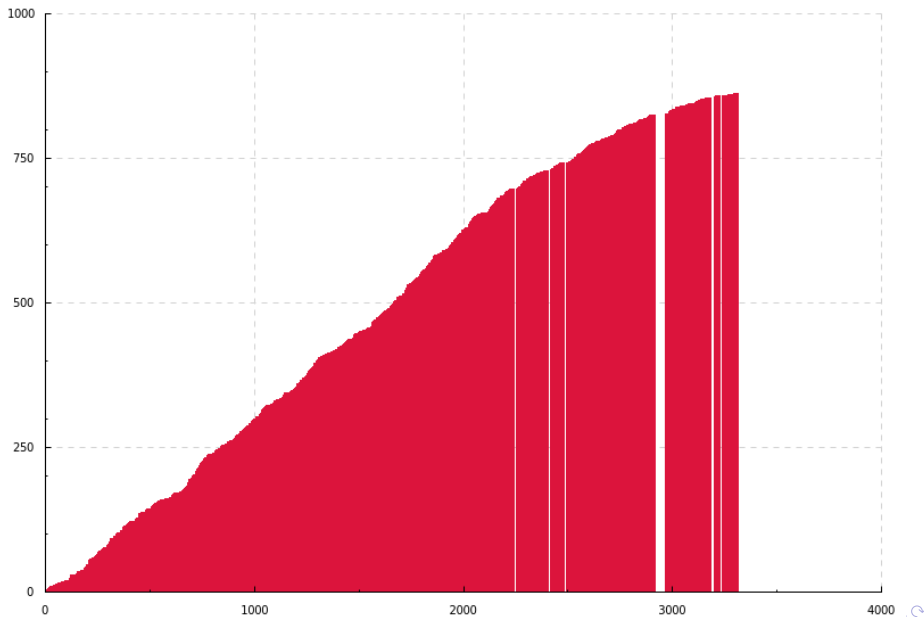
Open Issues (May 2015)



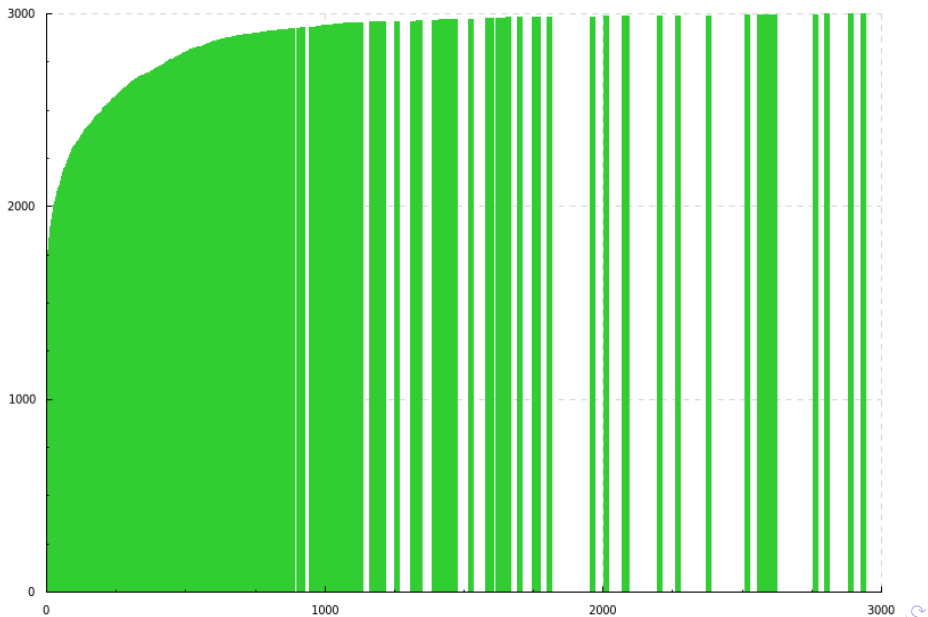
Open vs closed issues, 2015-08-08 to 2024-09-26



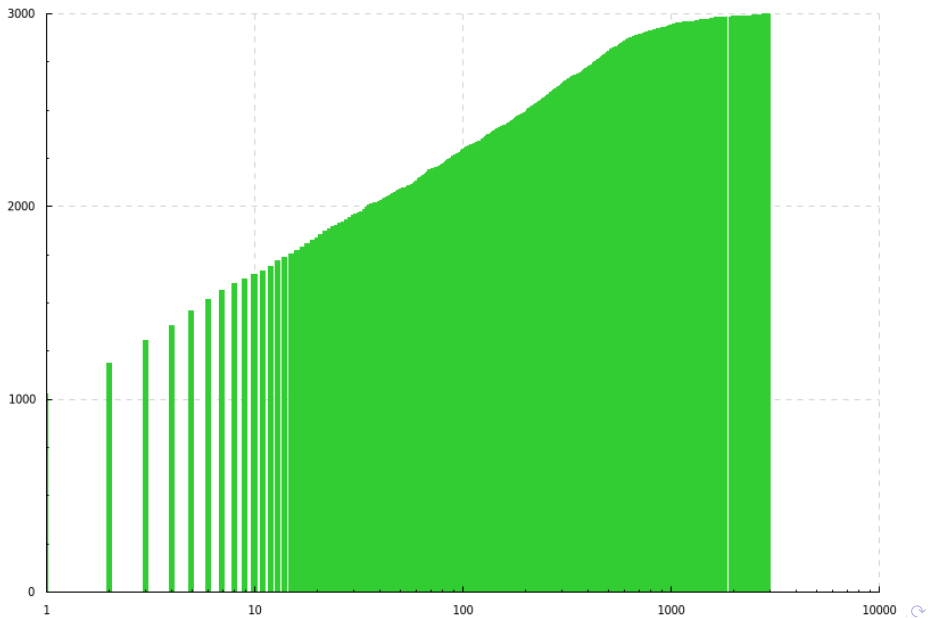
Open issue age (cumulative), 2015-08-08 to 2024-09-26



Closed issue age (cumulative), 2015-08-08 to 2024-09-26



Closed issue age logarithmically (cumulative), 2015-08-08 to 2024-09-26



Bug fixing is sometimes like Whack-A-Mole



Long on the Wish List (2015)

- 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 (2015)

- 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