

## C Publication lists

The publications most relevant for this project are marked with an arrow ( $\Rightarrow$ ).

### Selected Publications: Patrik Jansson

**Note to non computer scientists** Conference articles in computer science are peer reviewed full articles — not 1–2 page abstracts, and are the normal form of refereed publication. The top conferences in each subfield (like *POPL* and *ICFP* below) typically have the highest impact factor within that field, higher even than any journal.

#### Most cited publications (Google Scholar, 2013-03-25)

Jansson's Hirsch-index is **16**, his total citation count is over **1200** and the following papers are the five most cited (not including the papers in the last 8 years).

- P. Jansson and J. Jeuring. PolyP — a polytypic programming language extension. In *Proc. POPL'97: Principles of Programming Languages*, pages 470–482. ACM Press, 1997.  
Number of citations: **307**.
  - R. Backhouse, P. Jansson, J. Jeuring, and L. Meertens. Generic programming: An introduction. In *Advanced Functional Programming*, volume 1608 of *LNCS*, pages 28–115. Springer, 1999.  
Number of citations: **182**.
  - J. Jeuring and P. Jansson. Polytypic programming. In J. Launchbury et al., editors, *Advanced Functional Programming '96*, volume 1129 of *LNCS*, pages 68–114. Springer-Verlag, 1996.  
Number of citations: **158**.
  - M. Benke, P. Dybjer, and P. Jansson. Universes for generic programs and proofs in dependent type theory. *Nordic Journal of Computing*, 10(4):265–289, 2003. ISSN 1236-6064.  
Number of citations: **56**.
- $\Rightarrow$  P. Jansson and J. Jeuring. Polytypic data conversion programs. *Science of Computer Programming*, 43(1):35–75, 2002.  
Number of citations: **50**.

## Journal articles (last 8 years, excluding the above)

- J.-P. Bernardy, P. Jansson, and R. Paterson. Proofs for free — parametricity for dependent types. *J. Funct. Program.*, 22(02):107–152, 2012.  
Number of citations: **4**.
- J.-P. Bernardy, P. Jansson, M. Zalewski, and S. Schupp. Generic programming with C++ concepts and Haskell type classes — a comparison. *J. Funct. Program.*, 20(3–4):271–302, 2010c. URL <http://dx.doi.org/10.1017/S095679681000016X>.  
Number of citations: **10**.
- ⇒ S.-C. Mu, H.-S. Ko, and P. Jansson. Algebra of programming in Agda: dependent types for relational program derivation. *J. Funct. Program.*, 19:545–579, 2009. doi: 10.1017/S0956796809007345.  
Number of citations: **7**.

## Articles in refereed collections and conf. proceedings (last 8 years)

- ⇒ C. Ionescu and P. Jansson. Dependently-typed programming in scientific computing: Examples from economic modelling. In R. Hinze, editor, *24th Symposium on Implementation and Application of Functional Languages (IFL 2012)*, LNCS. Springer-Verlag, 2013a.  
Number of citations: **0**.
- C. Ionescu and P. Jansson. Testing versus proving in climate impact research. In *Proc. TYPES 2011*, volume 19 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41–54, Dagstuhl, Germany, 2013b. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi: 10.4230/LIPIcs.TYPES.2011.41.  
Number of citations: **0**.
- J. Duregård, P. Jansson, and M. Wang. Feat: Functional enumeration of algebraic types. In *Haskell’12*, pages 61–72. ACM, 2012. doi: 10.1145/2364506.2364515.  
Number of citations: **4**.
- J. Jeuring, P. Jansson, and C. Amaral. Testing type class laws. In *Haskell’12*, pages 49–60. ACM, 2012. doi: 10.1145/2364506.2364514.  
Number of citations: **0**.
- J. Duregård and P. Jansson. Embedded parser generators. In *Proceedings of the 4th ACM Symposium on Haskell*, Haskell ’11, pages 107–117, New York, NY, USA, 2011. ACM. doi: 10.1145/2034675.2034689.  
Number of citations: **3**.
- ⇒ J.-P. Bernardy, P. Jansson, and R. Paterson. Parametricity and dependent types. In *Proc. of ICFP 2010*, pages 345–356. ACM, 2010b.  
Number of citations: **25**.

- ⇒ J.-P. Bernardy, P. Jansson, and K. Claessen. Testing polymorphic properties. In A. Gordon, editor, *European Symposium on Programming*, volume 6012 of *LNCS*, pages 125–144. Springer, 2010a.  
Number of citations: **16**.
- A. Rodriguez, J. Jeuring, P. Jansson, A. Gerdes, O. Kiselyov, and B. C. d. S. Oliveira. Comparing libraries for generic programming in Haskell. In *Haskell'08*, pages 111–122. ACM, 2008.  
Number of citations: **56**.
  - D. Lincke, P. Jansson, M. Zalewski, and C. Ionescu. Generic libraries in C++ with concepts from high-level domain descriptions in Haskell: A DSL for computational vulnerability assessment. In *IFIP Working Conf. on Domain Specific Languages*, volume 5658/2009 of *LNCS*, pages 236–261, 2009.  
Number of citations: **5**.
  - J.-P. Bernardy, P. Jansson, M. Zalewski, S. Schupp, and A. Priesnitz. A comparison of C++ concepts and Haskell type classes. In *Proc. ACM SIGPLAN Workshop on Generic Programming (WGP)*, pages 37–48. ACM, 2008a.  
Number of citations: **20**.
  - S.-C. Mu, H.-S. Ko, and P. Jansson. Algebra of programming using dependent types. In *Mathematics of Program Construction*, volume 5133/2008 of *LNCS*, pages 268–283. Springer, 2008.  
Number of citations: **12**.
  - P. Jansson, J. Jeuring, and students of the Utrecht University Generic Programming class. Testing properties of generic functions. In Z. Horvath, editor, *Proceedings of IFL 2006*, volume 4449 of *LNCS*, pages 217–234. Springer-Verlag, 2007.  
Number of citations: **4**.
  - N. A. Danielsson, J. Hughes, P. Jansson, and J. Gibbons. Fast and loose reasoning is morally correct. In *POPL '06*, pages 206–217. ACM Press, 2006.  
Number of citations: **51**.

## Publicly available implementations (last 8 years)

I have participated in the development of the Agda proof engine (mainly through my PhD students Ulf Norell, Nils Anders Danielsson and Jean-Philippe Bernardy),

- U. Norell et al. Agda — a dependently typed programming language. Implementation available from Google Code: <http://code.google.com/p/agda/>, 2008.

The first description of Agda was in the PhD thesis of Ulf Norell (2007) and it has been cited  $\simeq 50$  times / year since then, indicating a quick spread in academia.

## Publication list for Jean-Philippe Bernardy

Database used for citation data: Google scholar.

### Peer-reviewed publications in journal

- ⇒ J.-P. Bernardy, P. Jansson, and R. Paterson. Proofs for free — parametricity for dependent types. *J. Funct. Program.*, 22(02):107–152, 2012.  
Number of citations: **4**.
- J.-P. Bernardy, P. Jansson, M. Zalewski, and S. Schupp. Generic programming with C++ concepts and Haskell type classes — a comparison. *J. Funct. Program.*, 20(3–4):271–302, 2010c. URL <http://dx.doi.org/10.1017/S095679681000016X>.  
Number of citations: **10**.

### Peer-reviewed publications in conferences and workshops

- ⇒ J.-P. Bernardy and G. Moulin. A computational interpretation of parametricity. In *Proc. of the Symposium on Logic in Comp. Sci.* IEEE, 2012.  
Number of citations: **4**.
- J.-P. Bernardy and M. Lasson. Realizability and parametricity in pure type systems. In M. Hofmann, editor, *FoSSaCS*, volume 6604 of *LNCS*, pages 108–122. Springer, 2011.  
Number of citations: **11**.
- ⇒ J.-P. Bernardy, P. Jansson, and R. Paterson. Parametricity and dependent types. In *Proc. of ICFP 2010*, pages 345–356. ACM, 2010b.  
Number of citations: **25**.
- ⇒ J.-P. Bernardy, P. Jansson, and K. Claessen. Testing polymorphic properties. In A. Gordon, editor, *European Symposium on Programming*, volume 6012 of *LNCS*, pages 125–144. Springer, 2010a.  
Number of citations: **16**.
- ⇒ J.-P. Bernardy. Lazy functional incremental parsing. In *Proc. of the 2nd ACM SIGPLAN symposium on Haskell*, pages 49–60. ACM, 2009.  
Number of citations: **8**.
- J.-P. Bernardy, P. Jansson, M. Zalewski, S. Schupp, and A. Priesnitz. A comparison of C++ concepts and Haskell type classes. In *WGP '08: Proc. of the ACM SIGPLAN workshop on Generic programming*, pages 37–48. ACM, 2008b.  
Number of citations: **20**.

## Non peer-reviewed publications

- J.-P. Bernardy. Yi: an editor in Haskell for Haskell. In *Proc. of the first ACM SIGPLAN symposium on Haskell*, pages 61–62. ACM, 2008.  
Number of citations: **6**.

## Publicly available implementations

I am the main contributor to the Yi project

- J.-P. Bernardy. Yi: an editor in Haskell for Haskell. In *Proc. of the first ACM SIGPLAN symposium on Haskell*, pages 61–62. ACM, 2008.

I have contributed the Agda proof engine. The first description of Agda was in the PhD thesis of Ulf Norell and it has been cited  $\simeq 50$  times / year since then, indicating a quick spread in academia.