A Bibliography About Formalized Binding

Randy Pollack

October 19, 2011

Some questions to consider when you look at these papers.

Is the representation general? Most representations only treat unitary binding; what about binding a list at one time? What about binding a set at one time? Can simultaneous substitution be represented? Explicit substitution? Some of the representations are even more special: [BHKM] only treats simply typed langages. Some Lambda-free logical frameworks [Ada08] can only represent object systems with specific finite orders.

Every approach mentioned below has something to say about instantiation of abstractions and substitution. Most have something to say about induction over the structure of the object language. You probably also need to define relations inductively, (e.g. typing, reduction) and functions recursively (e.g. height of a term, set of free variables). You will want to do rule induction (e.g. weakening of a typing relation). Maybe you need unification respecting binding over the language.

In some cases you might choose a representation tailored to the problem at hand, but I am interested in formalizing large parts of mathematics in the long term, and any special purpose representation will turn out to be inadequate for some naturally occurring example. Generality and expressiveness are important for me.

Is the representation easy to use? Basically, you need tool support to do significant reasoning about languages with binding. For example some object languages have several syntactic classes of variable: term variables, type variables, module variables, ... Defining substitution by hand for term variables in terms, for type variables in terms, ..., may become not just annoying, but prohibitive. Tool support is needed. In [BHKM], using new tool support, some extremely smart people do some rather specialized examples that no one had been able to do before. At the moment [PSR11] is a beautiful representation, but with no tool support. The similar representation [ACP$^+$08] has a useful library of metatheory and tactics. Nominal Isabelle [UBN07, Urb08, BU08] has several person-years of development and good tool support. Similarly Twelf [HL07] is highly developed, with many users. Newer systems like Abella [GMN09, Gac08] may have excellent ideas behind them, but not be ready for large scale use. Sysems with very few users [HMS01, FM09] are likely hard to get into.

1 First-order Representations

The type of lambda terms contains no actual binding in these representations; "binding" is in how terms are used.

1.1 Local representations

These use two species of names: local bound variables vs. global or free variables (parameters), following classical practice of Frege, Gentzen and Prawitz. The original locally named development [MP99, MP93] is not a canonical representation. Locally nameless [Gor93, ACP+08, Cha] was introduced to get a canonical representation. Finally a canonical named representation [PSR11] developed.

—

[ACP+08] B. Aydemir, A. Charguéraud, B. C. Pierce, R. Pollack, and S. Weirich. Engineering Formal Metatheory. In *Principles of Programming Languages POPL 2008*. ACM Press, 2008.

[Cha] Arthur Charguéraud. The locally nameless representation. *Journal of Automated Reasoning*, pages 1–46. 10.1007/s10817-011-9225-2.

[Gor93] Andrew Gordon. A mechanism of name-carrying syntax up to alpha-conversion. In *Higher Order Logic Theorem Proving and its Applications. Proceedings, 1993*, LNCS 780, pages 414–426. Springer-Verlag, 1993.

[MP93] J. McKinna and R. Pollack. Pure Type Systems formalized. In M.Bezem and J.F.Groote, editors, *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93, Utrecht*, number 664 in LNCS, pages 289–305. Springer-Verlag, March 1993.

[MP99] J. McKinna and R. Pollack. Some lambda calculus and type theory formalized. *Journal of Automated Reasoning*, 23(3–4), November 1999.

[PSR11] Randy Pollack, Masahiko Sato, and Wilmer Ricciotti. A canonical locally named representation of binding. *J. Automated Reasoning*, 2011.

1.2 Concrete or quotiented representation

Very few suggest unquontiented concrete representation [VB03, Sto88]. With extensional logics such as HOL4 and Isabelle/HOL it is possible to work with quotients [Nor06].

—

[Nor06] Michael Norrish. Mechanising λ-calculus using a classical first order theory of terms with permutations. *Higher-Order and Symbolic Computation*, 19:169–195, 2006.

[Sto88] A. Stoughton. Substitution revisited. *Theoretical Computer Science*, 17:317–325, 1988.

[VB03] René Vestergaard and James Brotherston. A formalised first-order confluence proof for the λ-calculus using one-sorted variable names. *Information and Computation*, 183(2):212 – 244, 2003. Special edition with selected papers from RTA01.

1.3 Pure de Bruijn representation

Rather than binding by name equality, use natural numbers to index binding depth. Original paper [dB72]. Some applications [NV07, Alt93, Ber11, Vou11]. A priori it seems like a clumsy representation (consider object theorems about permutation of typing contexts, for example), but its proponents often manage concise and elegant examples.

—

[Alt93] Thorsten Altenkirch. A formalization of the strong normalization proof for System F in LEGO. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications, TLCA'93*, volume 664 of *LNCS*. Springer-Verlag, March 1993.

[Ber11] Stefan Berghofer. A solution to the POPLmark challenge using de Bruijn indices in Isabelle/HOL. *Journal of Automated Reasoning*, 2011.

[dB72] N.G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indag. Math.*, 34(5), 1972.

[NV07] Michael Norrish and René Vestergaard. Proof pearl: De bruijn terms really do work. In Klaus Schneider and Jens Brandt, editors, *TPHOLs*, volume 4732 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2007.

[Vou11] Jerome Vouillon. A solution to the POPLmark challenge based on de Bruijn indices. *Journal of Automated Reasoning*, 2011.

2 Nominal representation

Nominal set theory [GP02]. Nominal logic as a theory [Pit03, Pit06]. An up-to-date review of nominal unification [Urb10]. Nominal representation implemented in Isabelle/HOL [UBN07, Urb08, BU08]. Second generation nominal Isabelle [HU10, UK11] explicitly supports binding of sets and lists of names.

—

[BU08] Stefan Berghofer and Christian Urban. Nominal inversion principles. In *Theorem Proving in Higher Order Logics, TPHOLs 2008*, LNCS. Springer-Verlag, 2008.

[GP02] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.

[HU10] Brian Huffman and Christian Urban. Proof pearl: A new foundation for nominal isabelle. In *Proceecings of the 1st Conference on Interactive Theorem Proving*, volume 6172 of *Lecture Notes in Computer Science*, pages 35–50. Springer Verlag, 2010.

[Pit03] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.

[Pit06] A. M. Pitts. Alpha-structural recursion and induction. *Journal of the ACM*, 53:459–506, 2006.

[UBN07] C. Urban, S. Berghofer, and M. Norrish. Barendregt's variable convention in rule inductions. In *Automated Deduction – CADE-21*, number 4603 in LNCS. Springer-Verlag, 2007.

[UK11] C. Urban and C. Kaliszyk. General bindings and Alpha-Equivalence in nominal isabelle. In *Proceedings of the 20th European Symposium on Programming (ESOP)*, volume 6602 of *LNCS*, pages 480–500, 2011.

[Urb08] Christian Urban. Nominal techniques in Isabelle/HOL. *J. Autom. Reasoning*, 40(4):327–356, 2008.

[Urb10] Christian Urban. Nominal unification revisited. In Maribel Fernandez, editor, *UNIF*, volume 42 of *EPTCS*, pages 1–11, 2010.

3 Higher-order Representations

Higher order properties of the meta language, such as function spaces or recursive datatypes, are used in these representations.

3.1 Higher Order Abstract Syntax (HOAS)

Original LF paper [HHP93]. Read [HL07] to understand LF and TWELF, the modern implementation of LF. Formal proof of adequacy of representation [CNV]. Many other HOAS approaches [PGO10, FP10, Mil00]. Two-layer approaches are interesting [GMN09, Gac08, FM09]. Another variant: Weak HOAS [HMS01, Chl08]. Lambda-free logical frameworks [Ada08, Luo03] are designed to have good algebraic properties, and elementary metatheory so that the proof of adequacy of representation of object systems is simple.

—

[Ada08] Robin Adams. Lambda-free logical frameworks. *CoRR*, abs/0804.1879, 2008.

[Chl08] Adam Chlipala. Parametric higher-order abstract syntax for mechanized semantics. In James Hook and Peter Thiemann, editors, *ICFP*, pages 143–156. ACM, 2008.

[CNV] James Cheney, Michael Norrish, and Rene Vestergaard. Formalizing adequacy: A case study for higher-order abstract syntax. *Journal of Automated Reasoning*, pages 1–31.

[FM09] Amy P. Felty and Alberto Momigliano. Reasoning with hypothetical judgments and open terms in hybrid. In António Porto and Francisco Javier López-Fraguas, editors, *PPDP*, pages 83–92. ACM, 2009.

[FP10] Amy P. Felty and Brigitte Pientka. Reasoning with higher-order abstract syntax and contexts: A comparison. In Matt Kaufmann and Lawrence C. Paulson, editors, *ITP*, volume 6172 of *Lecture Notes in Computer Science*, pages 227–242. Springer, 2010.

[Gac08] Andrew Gacek. The Abella interactive theorem prover (system description). In A. Armando, P. Baumgartner, and G. Dowek, editors, *Proceedings of IJCAR 2008*, volume 5195 of *Lecture Notes in Artificial Intelligence*, pages 154–161. Springer, August 2008.

[GMN09] Andrew Gacek, Dale Miller, and Gopalan Nadathur. Nominal abstraction. Submitted, 2009.

[HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the ACM*, 40(1):143–184, 1993. Preliminary version in LICS'87.

[HL07] Robert Harper and Daniel R. Licata. Mechanizing metatheory in a logical framework. *Journal of Functional Programming*, 17(4–5):613–673, July 2007.

[HMS01] Furio Honsell, Marino Miculan, and Ivan Scagnetto. An axiomatic approach to metareasoning on nominal algebras in HOAS. In *ICALP*, pages 963–978, 2001.

[Luo03] Zhaohui Luo. PAL$^+$: a lambda-free logical framework. *Journal of Functional Programming*, 13(2):317–338, 2003.

[Mil00] Dale Miller. Abstract syntax for variable binders: An overview. In John W. Lloyd, Vernica Dahl, Ulrich Furbach, Manfred Kerber, Kung-Kiu Lau, Catuscia Palamidessi, Lus Moniz Pereira, Yehoshua Sagiv, and Peter J. Stuckey, editors, *Computational Logic - CL 2000, First International Conference, London, UK, 24-28 July, 2000, Proceedings*, volume 1861 of *Lecture Notes in Computer Science*, pages 239–253. Springer, 2000.

[PGO10] Andrei Popescu, Elsa L. Gunter, and Christopher J. Osborn. Strong normalization for System F by HOAS on top of FOAS. *Logic in Computer Science*, pages 31–40, 2010.

3.2 Nested datatypes

[BP99, Ada06, BHKM]. [AR99] is for category theorists. [Chl08], categorized as Weak HOAS above, also uses nested datatypes.

—

[Ada06] Robin Adams. Formalized metatheory with terms represented by an indexed family of types. In Jean-Christophe Fillitre, Christine Paulin-Mohring, and Benjamin Werner, editors, *Types for Proofs and Programs*, volume 3839 of *LNCS*. Springer Berlin / Heidelberg, 2006.

[AR99] Thorsten Altenkirch and Bernhard Reus. Monadic presentations of lambda terms using generalized inductive types. In *Computer Science Logic, 13th International Workshop, CSL '99*, pages 453–468, 1999.

[BHKM] Nick Benton, Chung-Kil Hur, Andrew Kennedy, and Conor McBride. Strongly typed term representations in Coq. *Journal of Automated Reasoning*.

[BP99] Richard S. Bird and Ross Paterson. de bruijn notation as a nested datatype. *J. Funct. Program.*, 9:77–91, January 1999.

[Chl08] Adam Chlipala. Parametric higher-order abstract syntax for mechanized semantics. In James Hook and Peter Thiemann, editors, *ICFP*, pages 143–156. ACM, 2008.

4 Theories of binding

We already saw nominal logic as a first order theory [Pit03]. Here are more first order or axiomatic approaches [PR09, GM96, DHK02, PG11].

—

[DHK02] Gilles Dowek, Thrse Hardin, and Claude Kirchner. Binding logic: Proofs and models. In *In: LPAR 02: Proceedings of the 9th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning*, page 130. Springer-Verlag, 2002.

[GM96] Andrew Gordon and Tom Melham. Five axioms of alpha conversion. In Von Wright, Grundy, and Harrison, editors, *Ninth Conference on Theorem Proving in Higher Order Logics TPHOL'96, Turku*, volume 1125 of *LNCS*, pages 173–190. Springer-Verlag, August 1996.

[PG11] Andrei Popescu and Elsa L. Gunter. Recursion principles for syntax with bindings and substitution. In Manuel M. T. Chakravarty, Zhenjiang Hu, and Olivier Danvy, editors, *ICFP*, pages 346–358. ACM, 2011.

[Pit03] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.

[PR09] Andrei Popescu and Grigore Roşu. *Term-Generic Logic*, pages 290–307. Springer-Verlag, Berlin, Heidelberg, 2009.

5 The POPLmark challenge

A set of challenge problems involving reasoning about $F_{<:}$. See

https://alliance.seas.upenn.edu/ plclub/cgi-bin/poplmark

for solutions showing formal reasoning about binding in several proof tools (Twelf, Coq, Isabelle/HOL, Matita, Abella) using several representations (pure de Bruijn, HOAS, locally nameless, concrete first order, nominal, two level HOAS, nested datatypes).

6 Programmng with binders

Logic programming [CU08, NM88]. Functional programming in simply-typed languages [MM04, WYS11, Pot06, BH94, Che05]. Dependently typed languages [Pou11, PS09, Pie10, PD10, PC12, LZH08, LH09].

—

[BH94] Françoise Bellegarde and James Hook. Subsitution: A formal methods case study using monads and transformations. *Sci. Comput. Program.*, 23(2-3):287–311, 1994.

[CHD11] Manuel M. T. Chakravarty, Zhenjiang Hu, and Olivier Danvy, editors. *Proceeding of the 16th ACM SIGPLAN international conference on Functional Programming, ICFP 2011, Tokyo, Japan, September 19-21, 2011*. ACM, 2011.

[Che05] James Cheney. Scrap your nameplate: (functional pearl). In Olivier Danvy and Benjamin C. Pierce, editors, *ICFP*, pages 180–191. ACM, 2005.

[CU08] James Cheney and Christian Urban. Nominal logic programming. *ACM Trans. Program. Lang. Syst.*, 30:26:1–26:47, September 2008.

[LH09] Daniel R. Licata and Robert Harper. A universe of binding and computation. (Submitted for publication.), March 2009.

[LZH08] Daniel R. Licata, Noam Zeilberger, and Robert Harper. Focusing on binding and computation. In *LICS*, pages 241–252. IEEE Computer Society, 2008.

[MM04] Conor McBride and James McKinna. Functional pearl: i am not a number–i am a free variable. In *Proceedings of the 2004 ACM SIGPLAN workshop on Haskell*, Haskell '04, pages 1–9, New York, NY, USA, 2004. ACM.

[NM88] Gopalan Nadathur and Dale Miller. An overview of lambda prolog. In Kenneth A. Bowen and Robert A. Kowalski, editors, *Fifth International Logic Programming Conference, Seattle, Washington, August 1988*, pages 810–827. MIT Press, 1988.

[PC12] Brigitte Pientka and Andrew Cave. Programming with binders and indexed data-types. In *POPL*, 2012.

[PD10] Brigitte Pientka and Joshua Dunfield. Beluga: A framework for programming and reasoning with deductive systems (system description). In Jürgen Giesl and Reiner Hähnle, editors, *IJCAR*, volume 6173 of *Lecture Notes in Computer Science*, pages 15–21. Springer, 2010.

[Pie10] Brigitte Pientka. Beluga: Programming with dependent types, contextual data, and contexts. In Matthias Blume, Naoki Kobayashi, and Germán Vidal, editors, *FLOPS*, volume 6009 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2010.

[Pot06] Francois Pottier. An overview of Cαml. In *ACM Workshop on ML*, volume 148 of *Electronic Notes in Theoretical Computer Science*, pages 27–52, March 2006.

[Pou11] Nicolas Pouillard. Nameless, painless. In Chakravarty et al. [CHD11], pages 320–332.

[PS09] Adam Poswolsky and Carsten Schürmann. System description: Delphin - a functional programming language for deductive systems. *Electr. Notes Theor. Comput. Sci.*, 228:113–120, 2009.

[WYS11] Stephanie Weirich, Brent A. Yorgey, and Tim Sheard. Binders unbound. In Chakravarty et al. [CHD11], pages 333–345.