A Coq Proof-Mode for APRHL

Pierre-Yves Strub — LIX - École Polytechnique

2017

Differential privacy via approximate probabilistic liftings. Differential privacy is a rigorous definition of statistical privacy proposed by Dwork, McSherry, Nissim and Smith [DMNS06], and considered to be the gold standard for privacy-preserving computations. Over the last decade, differential privacy has achieved widespread adoption within the privacy community. Moreover, it has attracted significant attention from the verification community, resulting in several successful tools for formally proving differential privacy. Relational program logics [BFG+16, BGG+16, BKOB13, BO13] and relational refinement type systems [BGA+15] are currently the most flexible techniques for reasoning formally about differentially private computations. Their expressive power stems from approximate probabilistic liftings, a generalization of probabilistic liftings involving a metric on distributions. In particular, differential privacy is a consequence of a particular form of approximate lifting. These approaches have successfully verified differential privacy for many algorithms.

The Coq proof assistantand its *proof-mode* extensions. The Coq proof assistant is an environment for developing mathematical facts. This includes defining objects (integers, sets, trees, functions, programs, ...); making statements (using basic predicates and logical connectives); and finally writing proofs. The Coq compiler automatically checks the correctness of definitions (wellformed sets, terminating functions, ...) and proofs. The Coq environment helps with: advanced notations; proof search; modular developments. It also provides program extraction towards languages like Ocaml and Haskell for efficient execution of algorithms and linking with other libraries. Impressive examples have been done using Coq, including the formal verification of an optimizing C compiler [Ler16] or the formalization of the Feit-Thompson Theorem[†] [GAA+13]

As noted in [KTB17], *"when using a proof assistant to reason in an embedded logic (i.e. in a logic that has been formalized into the logic of the proof assistant), one cannot benefit from the proof contexts and basic tactics of the proof assistant. This results in proofs that are at a too low level of abstraction because they are cluttered with bookkeeping code related to manipulating the objects of the logic."* To remedy this situation, they designed [KTB17] a so-called proof mode that extends the Coq proof assistant with named contexts for managing the hypotheses of the embedded logic. Using their proof mode, it is possible to reason in the embedded logic as seamless as reasoning in the meta logic of Coq. Thy applied their solution to an impredicative higher-order separation logic for fine-grained concurrency named Iris [JSS+15], but the solution is general and could be applied to a variety of different embedded logics.

Goal of this internship. The goal of this internship is to develop a *proof-mode*, for the Coq proof assistant and following the lines of [KTB17], for the logic APRHLof [BFG+16]. To this end, the intern will formalize a proof of soundness of APRHL, relying on a new Coq library for (discrete) probabilities that is currently developed. When done, the intern will formalize, using her proof-mode, the security of mechanisms known to be differentially-private (Exponential mechanism, above threshold algorithm).

If time permits, the intern will try to extend APRHL(and its Coq proof-mode) for the verification adaptive data analysis algorithms used to prevent false discoveries, such as the one proposed by Dwork et al. [DFH+15], and for the formal verification of mechanism design [BGA+16].

[†] "Every finite group of odd order is solvable"

References

[BFG+16] Gilles Barthe, Noémie Fong, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. Advanced probabilistic couplings for differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 55–67. ACM, 2016.

[BGA+15] Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub. Higher-order approximate relational refinement types for mechanism design and differential privacy. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, pages 55–68. ACM, 2015.

[BGA+16] Gilles Barthe, Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Pierre-Yves Strub. Computer-aided verification for mechanism design. In Yang Cai and Adrian Vetta, editors, *Web and Internet Economics - 12th International Conference, WINE 2016, Montreal, Canada, December 11-14, 2016, Proceedings*, volume 10123 of *Lecture Notes in Computer Science*, pages 279–293. Springer, 2016.

[BGG+16] Gilles Barthe, Marco Gaboardi, Benjamin Grégoire, Justin Hsu, and Pierre-Yves Strub. Proving differential privacy via probabilistic couplings. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 749–758. ACM, 2016.

[BKOB13] Gilles Barthe, Boris Köpf, Federico Olmedo, and Santiago Zanella Béguelin. Probabilistic relational reasoning for differential privacy. *ACM Trans. Program. Lang. Syst.*, 35(3):9:1–9:49, 2013.

[BO13] Gilles Barthe and Federico Olmedo. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, volume 7966 of *Lecture Notes in Computer Science*, pages 49–60. Springer, 2013.

[DFH+15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 117–126. ACM, 2015.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[GAA+13] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. A machine-checked proof of the odd order theorem. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 163–179. Springer, 2013.

[JSS+15] Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, and Derek Dreyer. Iris: Monoids and invariants as an orthogonal basis for concurrent reasoning. In Sriram K. Rajamani and David Walker, editors, *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, pages 637–650. ACM, 2015.

[KTB17] Robbert Krebbers, Amin Timany, and Lars Birkedal. Interactive proofs in higher-order concurrent separation logic. In Giuseppe Castagna and Andrew D. Gordon, editors, *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, pages 205–217. ACM, 2017.

[Ler16] Xavier Leroy. The CompCert C verified compiler: Documentation and user's manual. Intern report, Inria, June 2016.