Project Proposal: First-Order Equational Reasoning via E-graphs and λ-terms

Michael Rawson

University of Southampton, UK
`michael@rawsons.uk`

Abstract

E-graphs are a popular data structure for reasoning about equality between variable-free terms, but it's not clear how to use them for universally-quantified equations, or even whether this is a good idea. I investigate one possibility by representing such equations as an equation between two λ-terms.

E-graphs and congruence closure algorithms are very effective for various tasks, and have been known for a long time [8]. In particular they are of great importance for satisfiability modulo theories [6]. Recently they have found a new lease of life in *equality saturation* [9] and `egg` [11] in particular.

The unit equational fragment of first-order logic considers a set of universally-quantified equation axioms of the form $\forall \bar{x}.\ l = r$ and a goal $s = t$ consisting of two ground terms s and t that should be shown to be equal to each other under the axioms. Congruence closure is not directly applicable here because of the quantifier, and so the unit equational fragment is generally tackled with *rewriting* techniques descended from Knuth-Bendix completion [2]. Various approaches have been developed to work around this, including SMT quantifier-instantiation routines, congruence closure with free variables [3], algorithms up to bounded term size [4], and using SOUR graphs [5] to combine congruence closure with completion [7].

I propose something a little different, which I believe to be novel. If one squints at a unit equation, say an equation declaring f to be commutative:

$$\forall xy.\ f(x, y) = f(y, x)$$

late enough in the day, it starts to look a little like an equation between two λ-terms:

$$(\lambda xy.\ f\ x\ y) = (\lambda xy.\ f\ y\ x)$$

which is probably most intuitive if thought of as *extensional* equality.

By closing the expressions, we can put them into an e-graph more easily. Concepts and processes like β-reduction from the λ-calculus can be implemented as local operations [10, 1] on the e-graph using, for example, string diagrams or a calculus of explicit substitutions. By applying these λ-terms to ground terms and β-reducing we can simulate instance generation, and with a more involved construction equations can be overlapped, but there may be an approach to unit equational reasoning which is more sympathetic to the e-graph approach.

Apart from the usual e-graph benefit of considering "all possible rewrites at once", there may be other advantages to consider equations as λ-terms. Taking the above example and η-reducing, we obtain

$$f = \lambda xy.\ f\ y\ x$$

If we suppose the goal is $f(a, b) = f(b, a)$, we obtain $f\ a\ b = (\lambda xy.\ f\ y\ x)\ a\ b$ immediately by transitive closure, and β-reduce to $f\ b\ a$, showing the goal with very little if any *search*.

Between now and AITP I plan to investigate this idea further and produce a prototype implementation, as I am curious about the possibilities here. I would greatly appreciate any feedback or recommended reading from reviewers, *especially* if this has been done before.

References

[1] Binding in e-graphs. https://pavpanchekha.com/blog/egg-bindings.html. Accessed: 2025-05-18.

[2] Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.

[3] Haniel Barbosa, Pascal Fontaine, and Andrew Reynolds. Congruence closure with free variables. In Axel Legay and Tiziana Margaria, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings, Part II*, volume 10206 of *Lecture Notes in Computer Science*, pages 214–230, 2017.

[4] Hendrik Leidinger and Christoph Weidenbach. Non-ground congruence closure. *CoRR*, abs/2412.10066, 2024.

[5] Christopher Lynch and Polina Strogova. SOUR graphs for efficient completion. *Discret. Math. Theor. Comput. Sci.*, 2(1):1–25, 1998.

[6] Greg Nelson and Derek C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Syst.*, 1(2):245–257, 1979.

[7] Christelle Scharff and Leo Bachmair. On the combination of congruence closure and completion. In Bruno Buchberger and John A. Campbell, editors, *Artificial Intelligence and Symbolic Computation, 7th International Conference, AISC 2004, Linz, Austria, September 22-24, 2004, Proceedings*, volume 3249 of *Lecture Notes in Computer Science*, pages 103–117. Springer, 2004.

[8] Robert E. Shostak. An algorithm for reasoning about equality. In Raj Reddy, editor, *Proceedings of the 5th International Joint Conference on Artificial Intelligence. Cambridge, MA, USA, August 22-25, 1977*, pages 526–527. William Kaufmann, 1977.

[9] Ross Tate, Michael Stepp, Zachary Tatlock, and Sorin Lerner. Equality saturation: a new approach to optimization. In Zhong Shao and Benjamin C. Pierce, editors, *Proceedings of the 36th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2009, Savannah, GA, USA, January 21-23, 2009*, pages 264–276. ACM, 2009.

[10] Aleksei Tiurin, Dan R Ghica, and Nick Hu. E-graphs with bindings. *arXiv preprint arXiv:2505.00807*, 2025.

[11] Max Willsey, Chandrakana Nandi, Yisu Remy Wang, Oliver Flatt, Zachary Tatlock, and Pavel Panchekha. egg: Fast and extensible equality saturation. *Proc. ACM Program. Lang.*, 5(POPL), January 2021.