

# The Homotopy Type of E-Graph Explanations: Measuring Proof Multiplicity in Equality Saturation

Third Projection Collective

Pseudonymous

---

## Abstract

Equality saturation engines such as egg record why two terms are equal. A note on higher e-graphs argues that the equivalence classes an e-graph computes do not determine its *space* of proofs; its example is a four-vertex diamond. We make the measure concrete on real proof records. egg’s explanation data form a multigraph, the *proof graph*, whose union-tree edges give a spanning forest. We prove that its geometric realization is, per component, a wedge of  $\beta_1$  circles. Explanations between two equal terms, taken up to homotopy, therefore form a torsor for a free group of rank  $\beta_1$ , and every recorded redundant connection is exactly one fundamental cycle.  $\beta_1$  can be maintained online with union-find. We implement the measure by exposing egg 0.9.5’s recorded connections, and run 600 equality-saturation instances over arithmetic and propositional rule sets. In both suites the proof space is far from contractible: the median  $\beta_1$  is 526.5 (arithmetic) and 233.5 (propositional). Almost all of these cycles pass through congruence edges; cycles made of rule edges alone are rare (at most 6). We also count redundant rule applications that egg drops by memoization; these make the recorded  $\beta_1$  a lower bound. The higher refinement, which attaches 2-cells that make congruence edges homotopic to the child paths inducing them, is stated but left *open*.

**2012 ACM Subject Classification** Theory of computation → Equational logic and rewriting

**Keywords and phrases** e-graphs, equality saturation, explanations, proof spaces, fundamental group

**Digital Object Identifier** 10.4230/LIPIcs...

## 1 Introduction

An e-graph [?] represents a congruence closure. With explanations enabled it also records a proof forest from which proofs of equalities are extracted [?, ?]. The note [?] (Proposition 5.1) observes that the connected components of a term-and-rewrite graph, i.e. the congruence classes, do not determine its homotopy type, and recommends that implementations “retain rewrite witnesses and coherence boundaries and declare [their] observational level”. That note gives a four-vertex example and no implementation. Here we compute the first non-trivial homotopy invariant of egg’s actual proof records.

### Contributions.

- (1) A definition of the proof graph of an explanation-enabled e-graph and a structure theorem for its realization (Theorem 2).
- (2) A torsor description of explanations up to homotopy and an explicit basis from egg’s union forest (Corollary 3).
- (3) An online algorithm (Proposition 4).
- (4) An implementation on egg 0.9.5 and an evaluation on 600 instances (Section 3).
- (5) A precise statement of the open 2-dimensional refinement.

## 2 The proof graph

Fix an explanation-enabled e-graph. egg stores one *explanation node* per e-node ever added. Each node has a list of *connections*: one for each union it took part in (justified by a rule or by congruence), and one for each *alternate rewrite*. An alternate rewrite is a rule application



© Third Projection Collective;  
licensed under Creative Commons License CC-BY 4.0  
Leibniz International Proceedings in Informatics  
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

whose two sides were already equal, recorded unless a one-step explanation between the same two nodes is already memoized.

► **Definition 1.** *The proof graph  $G$  has the explanation nodes as vertices and one edge per forward connection. Let  $V$ ,  $E$  and  $c$  be its vertex, edge and component counts, and put  $\beta_1(G) = E - V + c$ .*

Every union is performed between nodes in different classes and adds one edge, so the union edges form a spanning forest  $F$  of  $G$  with  $V - c$  edges whose components are the e-classes. Every other edge is an alternate rewrite.

► **Theorem 2.** *Each component  $G_j$  of  $G$  has realization  $|G_j| \simeq \bigvee_{\beta_1(G_j)} S^1$ , so  $\pi_1(|G_j|)$  is free of rank  $\beta_1(G_j)$  and  $H_1(|G_j|) \cong \mathbb{Z}^{\beta_1(G_j)}$ . Moreover  $\beta_1(G)$  equals the number of alternate-rewrite edges.*

**Proof.** A connected graph is homotopy equivalent to the quotient collapsing a spanning tree, which is a wedge of one circle per non-tree edge [?, Prop. 1A.2]. The tree  $F_j$  has  $V_j - 1$  edges, so there are  $E_j - V_j + 1$  non-tree edges. Summing over components gives  $E - V + c$ . The non-tree edges are exactly the alternate rewrites, by the remark above. ◀

► **Corollary 3.** *For vertices  $u, v$  in one component, the set of explanations from  $u$  to  $v$  (edge paths up to homotopy rel endpoints) is a torsor for the free group  $\pi_1(|G_j|, u)$ . A free basis consists of the fundamental cycles  $\gamma_e = F[u, s(e)] \cdot e \cdot F[t(e), u]$ , one per alternate rewrite  $e$ , where  $F[x, y]$  is the unique forest path.*

**Proof.** Pre-composing with loops at  $u$  acts freely and transitively on homotopy classes of paths  $u \rightarrow v$  (path lifting in the universal cover). The basis claim is the standard description of  $\pi_1$  of a graph relative to a spanning tree [?, Prop. 1A.2]. ◀

So  $\beta_1 = 0$  means the recorded explanation between any two equal terms is unique up to homotopy.  $\beta_1 > 0$  means egg holds non-homotopic proofs; its proof minimizer [?] chooses among exactly these.

► **Proposition 4.** *Maintaining a union-find over explanation nodes and incrementing a counter whenever a connection joins two already-connected nodes computes  $\beta_1$  in  $O(E \alpha(V))$  total time.*

**Proof.** Adding an edge within a component increases  $E - V + c$  by 1; adding an edge between components decreases  $c$  by 1 and increases  $E$  by 1; adding a vertex adds 1 to both  $V$  and  $c$ . ◀

### Memoized redundancy.

egg skips an alternate rewrite between  $x$  and  $y$  when a one-step explanation  $x \leftrightarrow y$  is already memoized. Let  $s$  be the number of skipped applications. Adding them as edges gives a graph with  $\beta_1 + s$ : each skipped edge joins nodes in one component. Many skipped applications are the same rule instance firing again in later iterations, so  $\beta_1 + s$  over-counts distinct proofs. The recorded  $\beta_1$  is the measure we report, and  $s$  is reported separately.

### 3 Experiments

#### Implementation.

We vendor egg 0.9.5 and add two public read-only accessors (`EGraph::proof_graph`, which returns the recorded connections with a congruence flag) and a counter of skipped alternates. No behaviour of egg changes. The driver (`src/main.rs`) generates random terms with a seeded xorshift generator, runs equality saturation (iteration limit 6, node limit 3000, time limit 5 s), and computes  $\beta_1$  with union-find. A sanity check, the unfilled diamond  $s \rightarrow u \rightarrow t$ ,  $s \rightarrow v \rightarrow t$ , yields 4 nodes, 4 edges and  $\beta_1 = 1$ , matching [?, Prop. 5.1].

#### Rule sets.

The arithmetic set has 9 rules (commutativity and associativity of  $+$  and  $\times$ , distributivity, factoring, units, zero). The propositional set has 12 rules (implication elimination, double negation, associativity, distributivity, commutativity, excluded middle, units, contrapositive). For each suite we use seeds 1 to 100 and depths 3, 4 and 5, giving 300 runs per suite.

#### Results

(`summary.json`, from `results.csv`; “non-trivial” means more than 4 explanation nodes):

|                                          | arithmetic     | propositional  |
|------------------------------------------|----------------|----------------|
| runs / non-trivial                       | 300 / 222      | 300 / 208      |
| non-trivial runs with $\beta_1 > 0$      | 222            | 204            |
| median / max $\beta_1$                   | 526.5 / 1561   | 233.5 / 1383   |
| median $\beta_1/V$                       | 0.209          | 0.107          |
| runs with rule-only $\beta_1 > 0$        | 126            | 35             |
| max rule-only $\beta_1$                  | 6              | 1              |
| median / max skipped alternates $s$      | 1437.5 / 4151  | 583.5 / 2713   |
| stop: saturated / iteration / node limit | 100 / 190 / 10 | 122 / 128 / 50 |

Rule-only  $\beta_1$  is computed on the subgraph of rule-justified edges. Almost all independent cycles use congruence edges. This is why the 2-cell refinement below matters: many of these cycles are probably filled once congruence edges are related to the child explanations that induce them.

### 4 The open refinement

For each congruence edge  $f(\bar{x}) \rightarrow f(\bar{y})$  there is a family of child explanations  $x_k \rightsquigarrow y_k$ , and through congruence these induce paths  $f(\bar{x}) \rightsquigarrow f(\bar{y})$ . Attaching a 2-cell between the edge and each induced path gives a 2-complex  $X(G)$  with  $\pi_1(X(G))$  a quotient of  $\pi_1(|G|)$ . **[OPEN: compute  $\pi_1(X(G))$  or  $H_1(X(G); \mathbb{Q})$  effectively. The induced paths live in different components, and congruence edges whose child explanations themselves use congruence make the attaching maps recursive. We expect a Squier-style completion in the sense of Kraus and von Raumer [?] to apply only to terminating fragments.]** We report  $\beta_1(G)$  as an upper bound on the rank of  $H_1(X(G))$ ; this follows because attaching 2-cells cannot increase  $H_1$ .

## **5    Related work**

Nieuwenhuis and Oliveras [?] introduced proof-producing congruence closure. Flatt et al. [?] minimize proofs over the same recorded connections we count. Patterson [?] gives an initial-functor semantics of e-graphs at the level of congruence classes. Kraus and von Raumer [?] build homotopy bases from confluence and termination. Our contribution is to measure, on real e-graph data, the first homotopy invariant that these works either use implicitly or leave aside.

## **6    Conclusion**

egg’s proof records carry a large free fundamental group. It is cheap to compute and has an explicit basis. Open: the 2-complex with congruence cells; benchmarks from published egg applications instead of random terms; a stability study across iteration limits. Agent assistance: drafted with an AI agent under human direction; the authors (Third Projection Collective) are responsible for the content.