E-matching ⊆ Relational Join
Simpler, faster, and optimal

Yihong Zhang, Remy Wang, Oliver Flatt, David Cao, Philip Zucker, Eli Rosenthal, Zachary Tatlock, Max Willsey

MAPPING E-GRAPHS TO RELATIONS

An example e-graph. Each solid box denotes an e-node and each dashed box denotes an e-class, which represents a set of equivalent terms. Labels at top-left corner denote the e-class id. Represented terms include \( f(1,g(1)), f(1,g(2)), f(2,g(1)), \) etc. (\( O(N^2) \) total).

REDUCING E-MATCHING TO CONJUNCTIVE QUERIES

\[ f(\alpha, g(\alpha)) \]
An e-matching pattern that matches all expressions where
1. the 1\textsuperscript{st} argument to \( f \) is \( g \) and
2. the 2\textsuperscript{nd} argument of \( f \) and the 1\textsuperscript{st} argument of \( g \) refer to the same e-class.

\[ Q(\text{root}, \alpha) \rightarrow R_f(\text{root}, \alpha, x), R_g(x, \alpha) \]
The conjunctive query derived from the pattern. Nested functions are flattened by introducing auxiliary variables \( (x) \).

Enumerated terms by backtracking-based e-matching \( (O(N^2)) \) many

\[
\begin{align*}
&f(1,g(1)) \\
&f(2,g(1)) \\
&f(2,g(2)) \\
&f(3,g(1)) \\
&f(3,g(2)) \\
&f(3,g(3))
\end{align*}
\]

Tuples visited by relational e-matching \( (O(N)) \) many

\[
\begin{align*}
&R_f \\
&\quad (c_1,1,g_1) \quad (c_2,1) \quad (c_3,1) \\
&\quad (c_2,2,g_2) \quad (c_2,2) \quad (c_3,2)
\end{align*}
\]

E-GRAPH & E-MATCHING

- E-graph is a data structure that efficiently represents sets of congruent terms.
- E-graph has wide applications in automated-theorem proving and program optimization.
- E-matching is a fundamental query of e-graphs that searches for a pattern modulo congruence.
- Existing backtracking-based e-matching algorithms rely on depth-first search over the e-graph and fail to take equality constraints over the pattern into consideration during query planning.

CQS & GENERIC JOIN

- Conjunctive query (CQ) is a restricted class of relational queries that only involve joins of relations.
- Generic join is an algorithm proposed by Ngo et al. that computes CQs in worst-case optimal time with respect to the output size.
- Has great performance especially when the CQ is complex (e.g., cyclic).

RELATIONAL E-MATCHING

- We propose relational e-matching, which reduces e-matching to CQs over a relational representation of e-graphs.
- The CQ form of e-matching fully exploits the equality constraints over the pattern, compared to existing backtracking-based algorithms where only the structural constraints are considered during query planning.
- To solve the complex CQs generated by relational e-matching, we use generic joins as our solver subroutine.
- Relational e-matching preserves the worst-case optimality of generic joins: Fix a pattern \( p \), let \( M(p,E) \) be the set of substitutions yielded by e-matching on e-graph \( E \) with size \( n \), relational e-matching runs in time \( O(\max_E(|M(p,E)|)) \).

BENCHMARK & RESULTS

We benchmarked with three representative e-matching queries, collected from the test suite for mathematical expressions of egg, a state-of-the-art e-graph framework.

- On e-matching queries with equality constraints (the cyclic and the non-linear acyclic cases), relational e-matching achieves asymptotic speed-ups up to \( 426 \times \) over the baseline e-matching algorithm by exploiting the equality constraints during query planning.
- On e-matching queries without equality constraints (the linear case), relational e-matching achieves similar performance as the baseline e-matching.

Speed-ups over backtracking-based e-matching algorithm

We compared our results to existing backtracking-based e-matching algorithms.

More details at