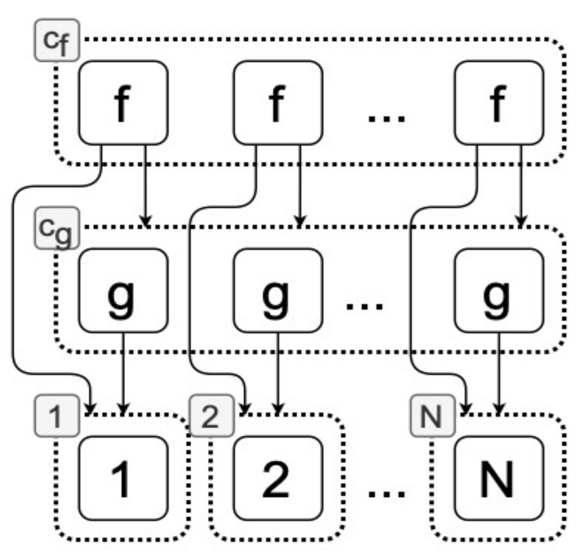
E-matching ⊆ Relational Join

Simpler, faster, and optimal

Yihong Zhang, yz489@cs.washington.edu
Ongoing collaboration with Remy Wang and Max Willsey. Work presented is done by the author.

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 denotes the e-class id. Represented terms include $f(1, g(1))$, $f(1, g(2))$, $f(2, g(1))$, etc. ($O(N^2)$ in total).

eclass-id	child₁	child₂
c_f	1	c_g
c_f	2	c_g
...	...	...
c_f	N	c_g

eclass-id	child₁
c_g	1
c_g	2
...	...
c_g	N

R_f : relation representing f (left). R_g : relation representing g (right).

REDUCING E-MATCHING TO CONJUNCTIVE QUERIES

$$f(\alpha, g(\alpha))$$

An e-matching pattern that matches all expressions where
1. the 1st argument to f is g and
2. the 2nd argument of f and the 1st argument of g refer to the same e-class.

$$Q(root, \alpha) :- R_f(root, \alpha, x), R_g(x, \alpha)$$

The conjunctive query derived from the pattern. Nested functions are flattened by introducing auxiliary variables (x).

Terms enumerated by backtracking-based e-matching ($O(N^2)$ many)

$f(1, g(1))$ ✓
$f(2, g(1))$
$f(2, g(2))$ ✓
$f(3, g(1))$
$f(3, g(2))$
$f(3, g(3))$ ✓
...

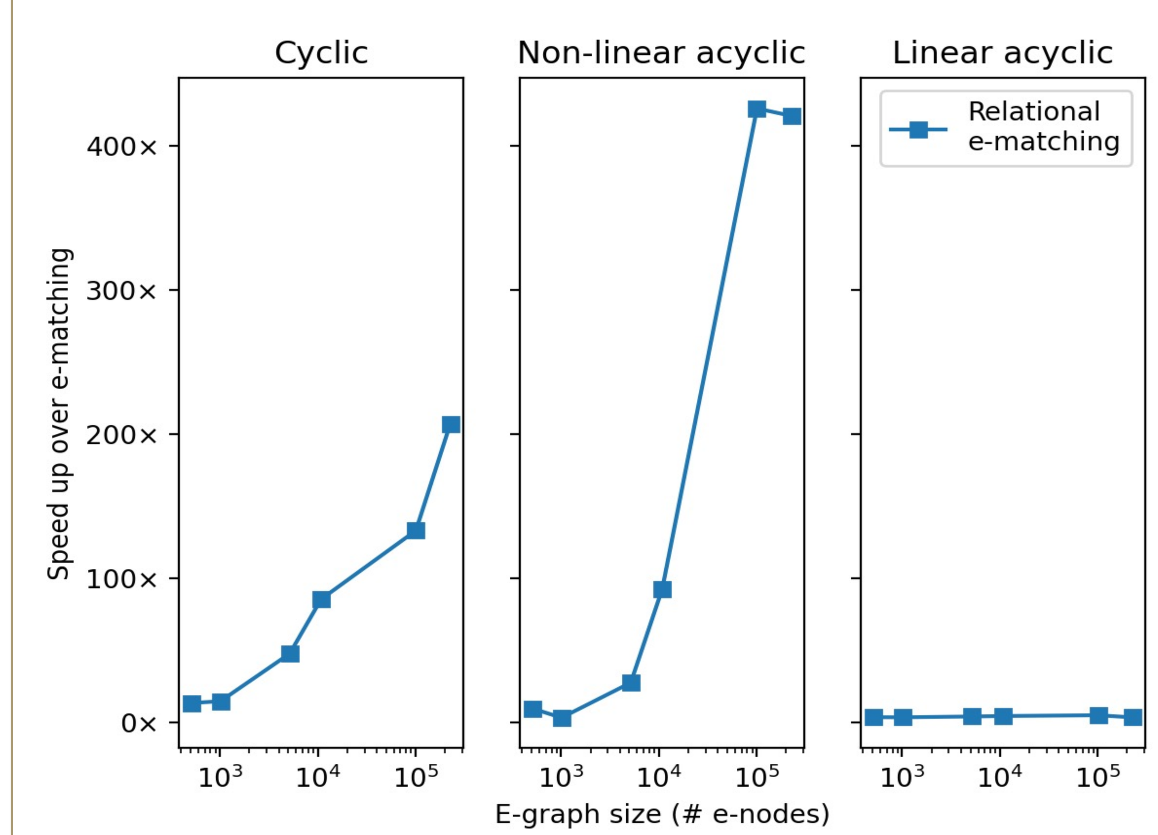
Tuples visited by relational e-matching ($O(N)$ many)

R_f	R_g
$(c_f, 1, c_g)$	$(c_g, 1)$ ✓
$(c_f, 2, c_g)$	$(c_g, 2)$ ✓
$(c_f, 3, c_g)$	$(c_g, 3)$ ✓
...	...

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 achieve **asymptotic speed-ups up to 426**× 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 (de Moura and Bjørner)



More details at

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 operation 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

$$\tilde{O}(\max_E(|M(p, E)|)).$$

PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING