MCTS-GEB: MCTS is a Good E-graph Builder

Guoliang He Zak Singh Eiko Yoneki

University of Cambridge

Background and motivation

Rewrite

Equality Saturation (EqsSat) uses an e-graph data structure to simultaneously represent many rewrite sequences, so it does not result in *destructive rewrite*.

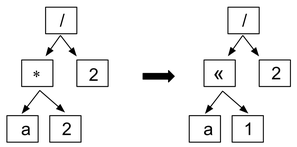


Figure 1. A term rewriting example.

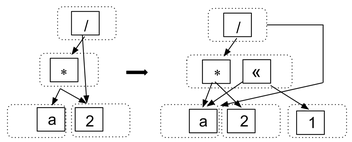


Figure 2. An e-graph rewrite example.

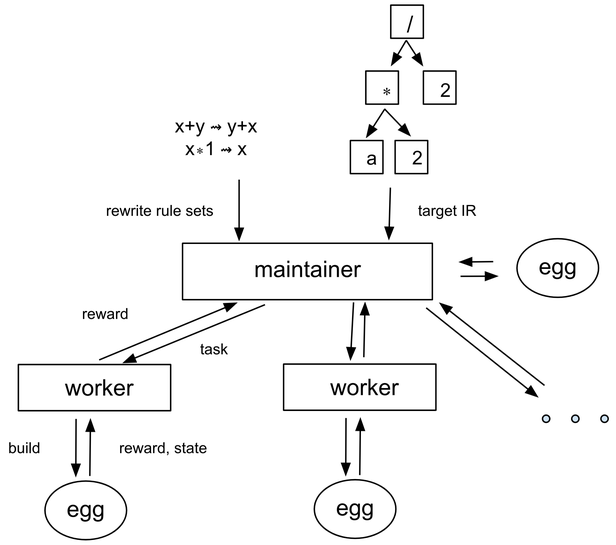
Limitations: phase-ordering during e-graph construction

- E-graphs can easily grow to very large graph data structures and continue indefinitely.
- The e-graph construction phase is upper-bounded by a pre-defined node limit.
- Some rewrite rules may result in faster hitting the node limit, so we need to reason about which rewrite rule to apply during the construction phase.

MCTS for e-graph construction

- Effectiveness: MCTS is proven to be effective for a wide range of tasks, including AlphaGo.
- Efficiency: MCTS can be easily parallelised.
- Scalability: MCTS may scale with large e-graphs.

MCTS-GEB



- MCTS-GEB takes as input an initial IR and a rewrite rule set, then outputs the optimised IR.
- MCTS-GEB uses multi-core to parallelise its planning phase.

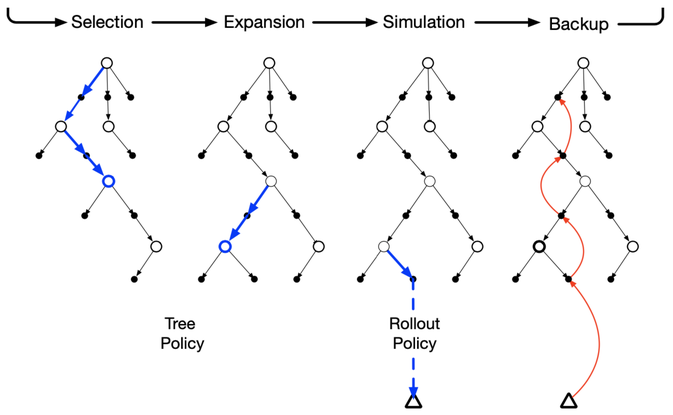


Figure 3. The MCTS algorithm consists of 4 steps: selection, expansion, simulation, and backpropagation.

Experiment results

We evaluate in the standard benchmark suites from EGG [a].

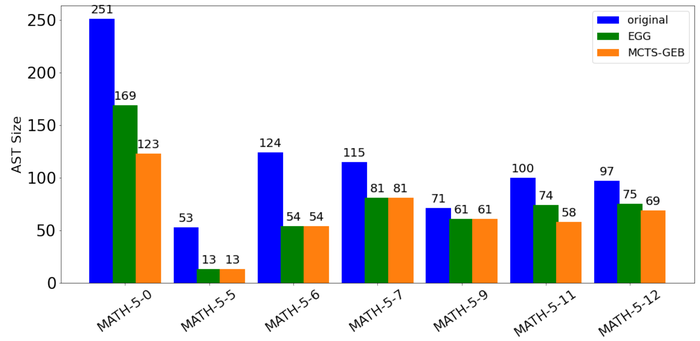


Figure 4. End-to-end expression simplification in the *Math* domain.

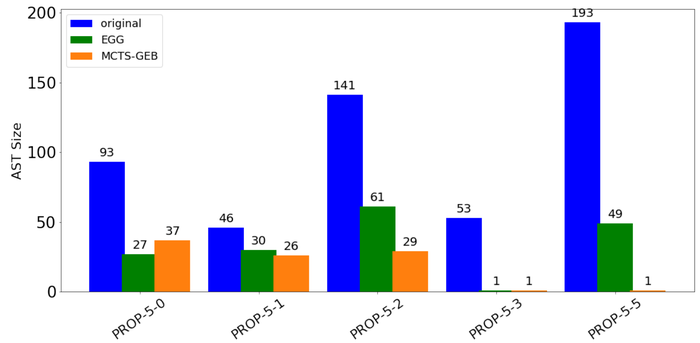


Figure 5. End-to-end expression simplification in the *Prop* domain.

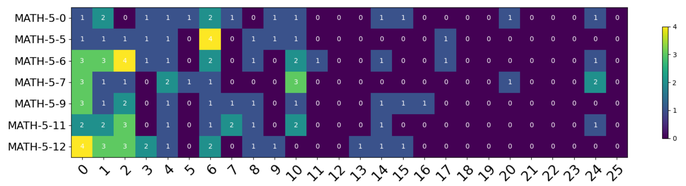


Figure 6. The heatmap shows MCTS-GEB can plan to apply rewrite rules.

[a] https://github.com/egraphs-good/egg