Improving Term Extraction with Acyclic Constraints

Mike He, Haichen Dong, Sharad Malik and Aarti Gupta

EGRAPHS 2023
Overview

+ Term Extraction
+ Extraction with ILP and its challenge
+ Our contribution: Acyclic constraints
Term Extraction

**Inputs:** e-graph; root e-class(es); cost model

**Output:** term(s) w/ the minimum cost
Term Extraction

Greedy Extraction

P1: Build costs from bottom-up

Cost model: AST Depth
Term Extraction

Greedy Extraction

P1: Build costs from bottom-up

Cost model: AST Depth

e-nodes
e-classes
Term Extraction

Greedy Extraction

P1: Build costs from bottom-up

Cost model: AST Depth
Term Extraction

Greedy Extraction

P2: choose the term with the least cost

Cost model: AST Depth
Greedy works in most cases, but may fail by yielding sub-optimal solutions
Term Extraction

Extracted by Greedy (29)

Optimal (25)
Term Extraction

Alternative: Integer linear programming (ILP)

For each e-node $n$, create a binary (0/1) variable $w_n$ (call them node variables)

Root Constraint: $\sum_{n} w_n \geq 1$

for all $n$ in the root e-class

Children Constraint: $\sum_{n' \in C_i} w_{n'} \geq w_n$

for each child $C_i$ of $n$

Objective: Minimize $\sum_{n} w_n \text{cost}(n)$
Avoiding Cycles: Topological sorting (ILP-Topo)

For each e-class $C$, create an integral value $t_C$ bounded by a sufficiently large value $\sigma$ (e.g. 2x number of the e-classes)

“If we pick an e-node $n$, then the topological order of its e-class must be greater than those of all its children.”

To expensive!
Acyclic Constraints

\[ X + Y + Z + W \leq 3 \]

\[ \neg X \lor \neg Y \lor \neg Z \lor \neg W \]
Weighted: clauses carry a positive weight.

**Background: Weighted (Partial) MaxSAT**

**Hard Clauses**

100% SAT

$H_1, H_2, \ldots, H_n$

**Soft Clauses**

SAT/ UNSAT

$S_1, S_2, \ldots, S_m$

**maximize** the sum of weights of SAT soft clauses.
Weighted MaxSAT Encoding

For each e-node $n$, create a boolean variable $w_n$ (call them node variables)

**Hard Clauses**
- Root Constraints
- Children Constraints
- Acyclic Constraints

**Soft Clauses**

$\neg w_n$
Weighted MaxSAT Encoding

Soft constraints are simply $\neg w_n$ for each e-node $n$ with weights $\text{cost}(n)$. 

Soft Clauses

$\neg W_n$
Weighted MaxSAT Encoding

Hard Clauses

Root Constraint: $\bigvee w_{n_i}$

Children Constraints: $w_n \rightarrow \bigvee \limits_{n' \in C_i} w_{n'}$

Acyclic constraints (naive): $\bigvee \neg w_n$ (for all cycles $\phi$)
Weighted MaxSAT Encoding

Reducing # of Acyclic Constraints

Acyclic constraints (naïve): $\bigvee_{n \in \phi} \neg w_n$ (for all cycles $\phi$)

The naïve encoding could yield exponentially many constraints for a single cycle of e-classes
Weighted MaxSAT Encoding

Reducing # of Acyclic Constraints

Instead of encoding on cycles of e-nodes, we could instead work with cycles of e-classes.

\[ \bigvee_{C_i} \bigwedge_{n \in C_i \cap \text{in_cycle}(n)} \neg w_n \]

Then, perform Tseitin Transformation to get CNF:

\[ \text{Tseitin} \left( (\neg w_A \land \neg w_B) \lor (\neg w_C \land \neg w_D) \lor (\neg w_E \land \neg w_F) \right) \leftrightarrow \]

\[ x_{AB} \leftrightarrow (\neg w_A \land \neg w_B) \]
\[ x_{CD} \leftrightarrow (\neg w_C \land \neg w_D) \]
\[ x_{EF} \leftrightarrow (\neg w_E \land \neg w_F) \]
\[ x_{AB} \lor x_{CD} \lor x_{EF} \]
ILP Encoding

Replacing the topological ordering constraints with acyclic constraints.

\[ x_{C_i} \leftrightarrow \bigwedge \neg w_j \]

If direction: \( (1 - x_{C_i}) + (1 - w_j) \geq 1 \)

Only-if direction: \( x_{C_i} + \sum w_j \geq 1 \)

Following Tseitin Transformation: \( \sum x_{C_i} \geq 1 \)

Call this encoding as **ILP-ACyc**
WPMA克斯AT and ILP-ACyc

Remaining Issue: # of cycles of e-classes?

$O(n)$ constraints per e-class cycle

$O(?) \times$
WPMAXSAT and ILP-ACyc

Remaining Issue: # of cycles of e-classes?

Our assumption: worst case is unlikely

\( O(n) \) constraints per e-class cycle

Worst case... \( O(2^{|C|}) \)
Experiments

Evaluation setup

Workload:
Extracting optimal terms from saturated e-graphs from Glenside
Tensor programs are obtained from ResNet-18/50, MobileNet, ResMLP and EfficientNet.

Rewrite Rules:
Im2Col: image-to-column transformations
Im2Col + SIMPL: Im2Col plus a set of simplification rewrites, including
• Operator collapsing (transpose, reshape, access, etc.)
• Operator reordering

Configurations:
5-second timeout for equality saturation
5-minute timeout for term extraction (including time of constructing constraints)

Results

E-graph statistics

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IM2COL</td>
<td>17266</td>
<td>15819</td>
<td>14754</td>
<td>21016</td>
<td>16305</td>
</tr>
<tr>
<td>IM2COL + SIMPL</td>
<td>17320</td>
<td>4247</td>
<td>4466</td>
<td>10978</td>
<td>20294</td>
</tr>
</tbody>
</table>

Table 1. Class cycle count after equality saturation

Good News!
Results

Constraint building + term extraction time

- Topological Sorting
- Acyclic Constraints
Improving Term Extraction with Acyclic Constraints

Mike He, Haichen Dong, Sharad Malik and Aarti Gupta
**Discussion**

**LP Relaxation of ILP-Topo**

Relaxation is not trivial:

“Weight vanishing”

Recall Children constraints:

\[-w_n + \sum_{n' \in C_i} w_{n'} \geq 0\]

\(w_n\) could be distributed over e-nodes in its children e-classes.

This is **bad**