Dynamic Lazy Grounding in Answer Set Programming

-Brian Hodges
Answer Set Programming (ASP)

- Same language as Nick Hippen’s Projector
- Uses rules to build “atoms” to describe a “model” of problem
  - pos(X, Y) :- row(X), col(Y).
- “Constraints” are rules with no “head” that must be false for the model to satisfy the problem
  - :- q(I,J), q(I1,J), I!=I1.
Solving ASP Programs

(Logic Program) \rightarrow \text{Grounder (Gringo)} \rightarrow \text{Grounded Program} \rightarrow \text{Solver (Clasp)} \rightarrow \text{Answer Sets}

(Clingo)
Grounding

- Occurs before Solving
- Instantiates all of the variables in an ASP logic program
- Can cause a dramatic increase in the size of the program (blow-up)

\[
\begin{align*}
\text{col}(1..n), \text{row}(1..n). & \quad \text{col}(1..4), \text{row}(1..4). \\
\{q(I,J) : \text{row}(I), \text{col}(J)\}. & \quad \{q(1,1)\}, \{q(1,2)\}, \\
& \quad \{q(1,3)\}, \{q(1,4)\}, \\
& \quad \{q(2,1)\}, \{q(2,2)\}, \\
& \quad \{q(2,3)\}, \{q(2,4)\}, \\
& \quad \{q(3,1)\}, \{q(3,2)\}, \\
& \quad \{q(3,3)\}, \{q(3,4)\}, \\
& \quad \{q(4,1)\}, \{q(4,2)\}, \\
& \quad \{q(4,3)\}, \{q(4,4)\}.
\end{align*}
\]
Solving with Lazy Grounding

(Clingo) → Grounder (Gringo) → Solver (Clasp) → Answer Sets

Base Program → Grounded Program → Answer Sets

Lazy Grounder

Lazy Constraints
Lazy Grounding

- Grounds during solving
- Avoids blow-up by only grounding on atoms in the current model
- 2 primary use cases
  - Instantiated rules have too low probability of being relevant to be worth grounding fully
  - Ground program is larger than computer memory
- We are only focusing on constraints to avoid the complexity of lazily instantiated atoms
Examples

Stable Marriage

- Finds matches between couples where no two individuals would find it preferable to swap partners
- If there is a low proportion of preferences the above rules is rarely invoked but still blows up
- Need to find a way to determine the cutoff of when to ground and when to use lazy grounding

Packing

- Attempts to fit a number of rectangular blocks in a rectangular space
- The collision constraint for even a small instance (10 packages) can blow-up to millions of grounded constraints
  - Every block at every position is checked with every other block at every position
- Not solvable with normal grounding
Mechanics

Separator

- Rewrites ASP programs into 2 portions
  - Main body
  - Expensive constraints

Lazy Grounder

- Grounds the main body of the program
- Solves the main body
- Reintroduces partially grounded constraints when they become relevant to solving
  - Only grounds based on current model of the program
Dynamic Lazy Grounding

Workflow

- Pull out expensive constraints
- Ground base program
- Pass data to an ML system to decide Lazy or Full grounding
- If Full: ground constraints and solve
- If Lazy: begin Lazy solve

Dynamic Benefits

- Can be used on existing programs
- Can choose to do lazy grounding based on problem instance
Conclusion / Future Work

● Done
  ○ Tested lazy grounding vs full grounding on Stable Marriage & Packing using dedicated lazy solvers
  ○ Developed preliminary constraint separator

● Next
  ○ Develop generic lazy solver
  ○ Begin machine learning for choosing Lazy vs Full
NSF Grant: Automated Optimization of Programs and Processing Tools in Answer Set Programming

- Eligibility: Undergraduate Student
- Funding Source: National Science Foundation (NSF)
- Commitment: 1 year, 10 hours/week
- Stipend: $150/week
- Application Method*: submit a Letter of Interest and Resume to Dr. Yuliya Lierler
References

