# Towards Lightweight Completion Formulas for Lazy Grounding in Answer Set Programming

Bart Bogaerts
(Joint work with Simon Marynissen and Antonius Weinzierl)

September 14, 2020 @ NMR'20

ARTIFICIAL
INTELLIGENCE
RESEARCH GROUP

# CONCLUSION

- Ground&Solve vs Lazy Grounding: complementary strengths
- Step towards obtaining insights into where the strengths come from
- Our focus (currently) is on missed propagations that could without too much computational overhead be detected
- Specifically, focus on completion formulas (standard in G&S; not present in LG)
- We identify several mechanisms that allow adding completion formulas
- No implementation yet. Effect to be evaluated...

# CONTEXT

- Answer Set Programming: Well-established KR paradigm
- Most systems are based on Ground & Solve
- Lazy Grounding:
  - Sometimes large parts of the grounding are useless
  - Interleave the two phases
  - Only ground what is required for correctness of the solver

# BOTTOM-UP GROUNDING

$$\mathrm{p}(\mathrm{X}) \leftarrow \mathrm{q}(\mathrm{X}, \mathrm{Y}) \land \mathrm{r}(\mathrm{Y}, \mathrm{Z}) \land \neg \mathrm{s}(\mathrm{X}, \mathrm{Y}, \mathrm{Z}).$$

# LAZY GROUNDING

$$\mathrm{p}(\mathrm{X}) \leftarrow \mathrm{q}(\mathrm{X}, \mathrm{Y}) \wedge \mathrm{r}(\mathrm{Y}, \mathrm{Z}) \wedge \neg \mathrm{s}(\mathrm{X}, \mathrm{Y}, \mathrm{Z}).$$

# LAZY GROUNDING

+ Avoids grounding useless parts

$$1 \leq \{p(1..100)\} \leq 1.$$

$$r(X_1, X_2, \ldots, X_k) \leftarrow p(X_1), p(X_2), \ldots p(X_k)$$

- Bad search performance

  Hypothesis: missed propagation!

  In particular: missing completion formulas

# COMPLETION FORMULAS

$$\mathrm{p}(1) \leftarrow \mathrm{q}(1,1) \wedge \mathrm{r}(1,2) \wedge \neg \mathrm{s}(1,1,2).$$
$$\mathrm{p}(1) \leftarrow \mathrm{q}(1,2) \wedge \mathrm{r}(2,2) \wedge \neg \mathrm{s}(1,2,2).$$
$$\mathrm{p}(1) \leftarrow \mathrm{q}(1,2) \wedge \mathrm{r}(2,3) \wedge \neg \mathrm{s}(1,2,3).$$

# PROBLEM

- How to recognize that all rules deriving an atom are "seen"?
- Five ideas all based on an analysis of the non-ground program. Based on idea of "bound":
  - $\mathrm{X}$ and $\mathrm{Y}$ sets of variables in $\mathrm{r}$
  - mapping $\mathrm{f} : \mathrm{sub}(\mathrm{X}) \rightarrow 2^{\mathrm{sub}(\mathrm{Y})}$ is a bound if it "captures" relevant substitutions
  - denoted $\mathrm{f} : \mathrm{X} \curvearrowright \mathrm{Y}$
  - Mostly interested in small bounds (preferably functional)

# USING BOUNDS

Let $\mathrm{h}$ be a ground atom. Let $\mathrm{r}_1, \ldots, \mathrm{r}_\mathrm{n}$ be the rules in $\mathcal{P}$ whose head unifies with $\mathrm{h}$. Let $\sigma_\mathrm{i}$ denote the most general unifier of $\mathrm{h}$ and $head(\mathrm{r}_\mathrm{i})$. If there is a bound $\mathrm{f}_\mathrm{i} : var(head(\mathrm{r}_\mathrm{i})) \curvearrowright var(\mathrm{r}_\mathrm{i})$ for all i, then

$$\neg \mathrm{h} \vee \bigvee_{1 \leq \mathrm{i} \leq \mathrm{n}} \bigvee_{\tau \in \mathrm{f}_\mathrm{i}(\sigma_\mathrm{i})} \beta(\tau(\mathrm{r}_\mathrm{i}))$$

holds in all answer sets of $\mathcal{P}$.

# IDEA 1: NON-PROJECTIVE RULES

$$move(\mathrm{X}, \mathrm{Y}, \mathrm{XX}, \mathrm{YY}) \leftarrow valid(\mathrm{X}, \mathrm{Y}, \mathrm{XX}, \mathrm{YY}), \neg other(\mathrm{X}, \mathrm{Y}, \mathrm{XX}, \mathrm{YY}).$$

# IDEA 2: FUNCTIONAL DEPENDENCIES

$$\begin{aligned}\{gt(\mathrm{A},\mathrm{X},\mathrm{U})\} \leftarrow\, & elem(\mathrm{A},\mathrm{X}), comUnit(\mathrm{U}),\\ & comUnit(\mathrm{U}_1), \mathrm{U}_1 = \mathrm{U}+1, rule(\mathrm{A}),\\ & \mathrm{U} < \mathrm{X}.\end{aligned}$$

# IDEA 3: DOMAIN PREDICATES

$$colored(\mathrm{N}) \leftarrow assign(\mathrm{N}, \mathrm{C}), color(\mathrm{C})$$

# IDEA 4: COMBINING BOUNDS

- e.g. $f_1 : X \curvearrowright Y_1$ and $f_2 : X \curvearrowright Y_2$ gives rise to bound : $X \curvearrowright Y_1 \cup Y_2$
- e.g., composition : $X \curvearrowright Y$ and : $Y \curvearrowright Z$

$$h(X) \leftarrow X + 1 = Z, Z = U.$$

# IDEA 5: BOUNDS ON ARGUMENT POSITIONS

$$colored(\mathrm{N}) \leftarrow assign(\mathrm{N}, \mathrm{C}) \sout{\wedge\, color(\mathrm{C})}$$

# IMPLEMENTATION & EXPERIMENTS

# CONCLUSION

- Ground&Solve vs Lazy Grounding: complementary strengths
- Step towards obtaining insights into where the strengths come from
- Our focus (currently) is on missed propagations that could without too much computational overhead be detected
- Specifically, focus on completion formulas (standard in G&S; not present in LG)
- We identify several mechanisms that allow adding completion formulas
- No implementation yet. Effect to be evaluated...