Exact ASP Counting with Compact Encodings

Mohimenul Kabir[1], Supratik Chakraborty[2], Kuldeep S Meel[3]

[1]National University of Singapore, [2]Indian Institute of Technology, Bombay, [3]University of Toronto

Problem Statement

Answer Set Programming (ASP)
A rule-based language for problem encoding

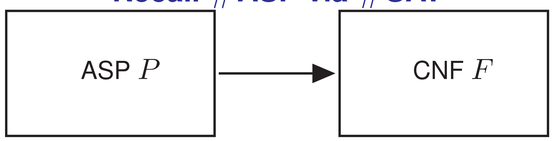
$$h \leftarrow b_1, \ldots, b_k, \sim b_{k+1}, \ldots, \sim b_{k+m}$$

Answer Set Counting (#ASP)
Given a normal program P, counts the number of answer sets of P, which is denoted as $\mathsf{CntAS}(P)$ and $\mathrm{AS}(P)$ denotes answer sets of P
Applications: Wide range of applications in probabilistic inference, network reliability, planning, navigation, etc.
Existing Techniques for #ASP include (i) enumeration (ii) dynamic programming on tree representation (iii) CNFization+#CNF
Main Contribution: A tool for #ASP, named sharpASP

Unit Propagation

- Simplifies a Boolean formula via (i) removing falsified literals (ii) removing satisfied clauses and (iii) performing inference
- If $M \models F$, then $F|_M = \emptyset$ (denotes the unit propagation of $F \wedge M$)

Recall #ASP via #SAT



- Clark Completion $\mathsf{Comp}(P)$ preserves the answer sets of P but the converse is not true for cyclic programs
  cyclic $\Rightarrow$ exists some cyclic relations between program atoms
- Existing encodings to preserve one-to-one correspondence include
  - Loop formula + $\mathsf{Comp}(P)$
  - Unfolding + $\mathsf{Comp}(P)$
  - Level Ranking + $\mathsf{Comp}(P)$
- One-to-one encodings hurt the scalability of counting algorithms

Source Code

`https://github.com/meelgroup/sharpASP`

Methodology

Key Observation

- Answer Set Definition: "*Each atom of an answer set must be justified.*"
- Under Clark's completion, all non-cyclic atoms are justified.
- Key Insight: "*It suffices to justify cyclic atoms only*".

Checking Justification

"copy atom": introduce a new atom $v^\star$ for each cyclic atom v
Purpose of "Copy atom" $v^\star$: Checking justification of atom v
Construct a Boolean formula $\mathsf{Copy}(P)$ as follows:

- for each cyclic atom v, add a clause

$$\neg v^\star \vee v$$

- for each rule $v \leftarrow a_1, \ldots a_k, b_1, \ldots b_m, \sim c_1, \ldots \sim c_n \in P$, where v, a_i are cyclic atoms and none of b_i is a cyclic atom, add a clause

$$\neg a_1^\star \vee \ldots \neg a_k^\star \vee \neg b_1 \vee \ldots \neg b_m \vee c_1 \vee \ldots c_n \vee v^\star$$

High-level Idea: If v is 1 and justified, then $v^\star$ unit propagates to 1

Alternative Answer Set Definition

$M \in \mathrm{AS}(P)$ if and only if $M \models \mathsf{Comp}(P)$ and $\mathsf{Copy}(P)|_M = \emptyset$

Counting Answer Sets

Notation: $P = (\underbrace{\mathsf{Comp}(P)}_{F}, \underbrace{\mathsf{Copy}(P)}_{G})$

$$\mathsf{CntAS}(F, G) = \mathsf{CntAS}(F|_{\neg x}, G|_{\neg x}) + \mathsf{CntAS}(F|_x, G|_x),$$
for non-copy variable x

$$\mathsf{CntAS}(\bot, G) = 0$$

$$\mathsf{CntAS}(\emptyset, G) = \begin{cases} 1 & \text{if } G = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

An Example

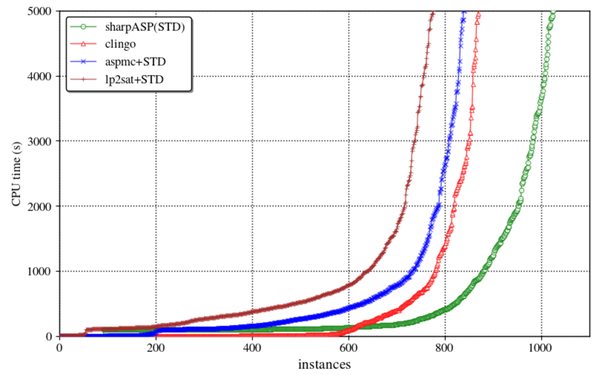
Consider program $P = \{r_1 \equiv a \leftarrow b.\ r_2 \equiv b \leftarrow a.\ r_3 \equiv s \leftarrow\sim a.\}$.
$\mathsf{Comp}(P) = \{(a \leftrightarrow b) \wedge (b \leftrightarrow a) \wedge (s \leftrightarrow \neg a)\}$
$\mathsf{Copy}(P) = \{\neg a^\star \vee a, \neg b^\star \vee b, \neg a^\star \vee b^\star, \neg b^\star \vee a^\star\}$.

- For answer set $M = \{s\}$, $M \models \mathsf{Comp}(P)$ and $\mathsf{Copy}(P)|_M = \emptyset$
- For non-answer set $M = \{a, b\}$, $M \models \mathsf{Comp}(P)$ but $\mathsf{Copy}(P)|_M = \{\neg a^\star \vee b^\star, \neg b^\star \vee a^\star\} \neq \emptyset$, since none of $a^\star$ and $b^\star$ unit propagates in $\mathsf{Copy}(P)$

Empirical Evaluation

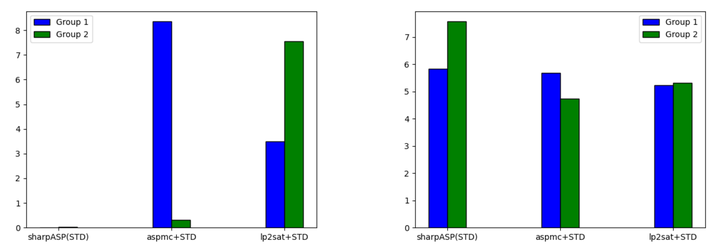
	clingo	ASProb	aspmc +STD	lp2sat +STD	sharpASP (STD)
Hamil. (405)	230	0	167	112	300
Reach. (600)	318	149	421	471	463
aspben (465)	321	39	252	193	260
Total (1470)	869	188	840	776	**1023**
PAR-2	4285	8722	4572	5084	**3373**

The performance of sharpASP vis-a-vis other ASP counters in terms of the number of instances counted within a time limit of 5000 seconds and a memory limit of 8GB, and the last row shows the PAR-2 scores.



The runtime performance of sharpASP vis-a-vis other ASP counters

Ablation Study



BCP time (seconds) #Decisions (10-base log)

Group 1: sharpASP outperforms and Group 2: does not outperform
sharpASP spends less time in BCP but makes more decisions.

Concluding Remarks

- We propose an ASP counter that counts answer sets using an alternative answer set definition, avoiding the one-to-one corresponding encoding, leveraging a #SAT-like technique.