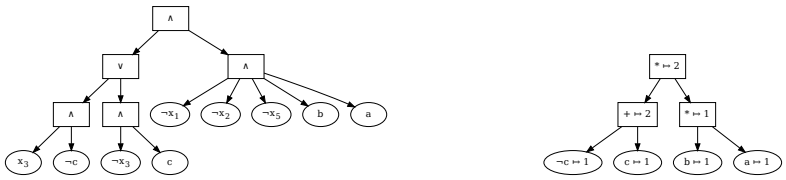

Fig. 2: Φ_{Π_1} (left) and $\tau(\mathcal{G}_{\Pi_1})$ labeled with evaluations (right).

A Additional Examples

Example 9. Compare Figures 1 and 2 (right) for an example of absorption. In terms of Algorithm 1, we see that the two most left remaining $\wedge$-nodes had removed each child corresponding to auxiliary variables, so that both had only one child $\neg c$ and c, respectively, left. Accordingly, both $\vee$-nodes were absorbed by their only child, respectively, as depicted in Figure 2 (right). Figure 2 (right) illustrates that we can compress $\mathcal{G}_{\Pi_1}$ with $|\mathcal{G}_{\Pi_1}| = 13 = |\varphi_{\Pi_1}|$ to $|\tau(\mathcal{G}_{\Pi_1})| = 6$.

B Omitted Supplemental Material

Lemma 1. $\mathcal{M}((\Phi_\Pi)^L) = \mathcal{S}(\Pi^L)$ for program Π and assumptions L.

Proof. We first establish the following claim:

$$\mathrm{comp}(\Pi^L) = \mathrm{comp}(\Pi \cup \mathrm{ic}(L)) = \mathrm{comp}(\Pi) \wedge \bigwedge_{\ell \in L} \ell \tag{1}$$

By definition, $\mathrm{comp}(\Pi^L) = \mathrm{comp}(\Pi \cup \mathrm{ic}(L))$. This further evaluates to $\mathrm{comp}(\Pi) \cup \mathrm{ic}(L)$. Since $\bot$ evaluates to false always and

$$\mathrm{comp}(\{\bot \leftarrow B(r)^+, \neg B(r)^- \mid r \in \Pi, H(r) = \bot\}) = \bot \leftrightarrow \bigvee_{r \in \Pi, H(r) = \bot} BF(r),$$

we have that

$$\begin{aligned}\mathcal{M}(\bot \leftrightarrow \bigvee_{r \in \Pi, H(r) = \bot} BF(r)) &= \mathcal{M}(\bigwedge_{r \in \Pi, H(r) = \bot} \bot \leftrightarrow BF(r)),\\ &= \mathcal{M}(\bigwedge_{r \in \Pi, H(r) = \bot} \neg BF(r)).\end{aligned}$$

As a result,

$$\begin{aligned}\mathcal{M}(\mathrm{comp}(\Pi^L \setminus \mathrm{ic}(L)) \cup \mathrm{ic}(L)) &= \mathcal{M}(\mathrm{comp}(\Pi^L \setminus \mathrm{ic}(L)) \cup \bigcup_{\ell \in L} \mathrm{comp}(\mathrm{ic}(\ell)) \\ &= \mathcal{M}(\mathrm{comp}(\Pi) \wedge \bigwedge_{\ell \in L} \mathrm{comp}(\mathrm{ic}(\ell))) \\ &= \mathcal{M}(\mathrm{comp}(\Pi) \wedge \bigwedge_{\ell \in L} \neg BF(\mathrm{ic}(\ell))) \\ &= \mathcal{M}(\mathrm{comp}(\Pi) \wedge \bigwedge_{\ell \in L} \ell).\end{aligned}$$

In consequence, Equation 1 holds.

It remains to show that conditioning $(\Phi_\Pi)^L$ in the sd-DNNF Φ_Π preserves all models according to Π under the set L of assumptions. By definition of conditioning, it holds that $\mathcal{M}((\Phi_\Pi)^L) = \mathcal{M}(\Phi_\Pi \wedge \bigwedge_{\ell \in L} \ell)$. By assumption, it is true that $\mathcal{M}(\Phi_\Pi \wedge \bigwedge_{\ell \in L} \ell) = \mathcal{M}(\mathrm{comp}(\Pi) \wedge \bigwedge_{\ell \in L} \ell)$. From Equation 1, we obtain that $\mathcal{M}(\mathrm{comp}(\Pi) \wedge \bigwedge_{\ell \in L} \ell) = \mathcal{M}(\mathrm{comp}(\Pi^L))$. By definition, $\mathcal{M}(\mathrm{comp}(\Pi^L)) = \mathcal{S}(\Pi^L)$. In consequence, we established that $\mathcal{M}((\Phi_\Pi)^L) = \mathcal{S}(\Pi^L)$. Hence, the Lemma sustains. □

Lemma 2. Let Π be a program, Φ_Π an sd-DNNF of $\mathrm{comp}(\Pi)$ after a transformation that preserves the number of models, but introduces auxiliary variables, and $\mathcal{G}_\Pi$ its counting graph. Then, $\mathrm{val}(\tau(\mathcal{G}_\Pi)) = \mathrm{val}(\mathcal{G}_\Pi)$ and $\tau(\mathcal{G}_\Pi)$ can be constructed in time $\mathcal{O}(2 \cdot |\Phi_\Pi|)$.

Proof. Let $\mathcal{G}_\Pi$ be the counting graph of an sd-DNNF that is equivalent to the CNF that has been constructed from $\mathrm{comp}(\Pi)$ using a transformation that preserves the number of models, which usually is the Tseitin transformation. We show that the value $\mathrm{val}(N)$ of each node N of $\mathcal{G}_\Pi$, which is not removed in $\tau(\mathcal{G}_\Pi)$, does not change, since for N and its respective children $\mathit{children}(N)$ in Algorithm 1 we modify only literals that occur in the program Π. By $N_\tau \in \tau(\mathcal{G}_\Pi)$ we denote the modified version of N, and by $\mathit{children}_\tau(N)$ we denote the children of N in $\tau(\mathcal{G}_\Pi)$. We distinguish the cases:

1. Suppose N is a literal node. Let ℓ denote the corresponding literal. If $\ell \notin \mathcal{L}(\Pi)$, then N is removed in $\tau(\mathcal{G}_\Pi)$, thus by contraposition, we know that, if N is not removed in $\tau(\mathcal{G}_\Pi)$, then $\ell \in \mathcal{L}(\Pi)$. Assume $\ell \in \mathcal{L}(\Pi)$. Then $N = N_\tau$. Therefore, $\mathrm{val}(N) = \mathrm{val}(N_\tau) \in \{0, 1\}$.
2. Suppose N is not a literal node. Then, since N is an $\wedge$- or an $\vee$-node, we know that $|\mathit{children}(N)| \geq 2$. However, in general $0 \leq |\mathit{children}_\tau(N)| \leq |\mathit{children}(N)|$.
   (a) Assume $|\mathit{children}_\tau(N)| = 0$. Then, in Algorithm 1, N will be ignored and thus not belong to $\tau(\mathcal{G}_\Pi)$.
   (b) Assume $|\mathit{children}_\tau(N)| = 1$. Then, in Algorithm 1, N will be absorbed by its only child. Thus, N does not belong to $\tau(\mathcal{G}_\Pi)$.
   (c) Assume $|\mathit{children}_\tau(N)| \geq 2$. Then in Algorithm 1, N will be evaluated on $\mathit{children}_\tau(N)$, which means N_τ will be contained in $\tau(\mathcal{G}_\Pi)$. We now

need to show that val(N) on *children*(N) corresponds to val(N) on *children*$_\tau$(N), i.e., val(N) = val(N_τ). By assumption (number of models is preserved), we have a bijection between $M(\Phi_\Pi)$ and $\mathcal{S}(\Pi)$ which ignores auxiliary variables. Therefore, we can simply set the values of children *children*(N) that have been removed or absorbed due to Cases 2a, 2b, or 2c – as a consequence of removing auxiliary variables – to the corresponding neutral element of the value of N.

i. Assume N is an $\wedge$-node. Accordingly, in Algorithm 1, N will be evaluated on *children*$_\tau$(N) such that in the product corresponding to val(N), the value of each removed branch (removed child), due to removing auxiliary variables, corresponds to the neutral element of multiplication, i.e., 1. Therefore, we conclude that val(N) = val(N_τ).
ii. Assume N is an $\vee$-node. Again, accordingly, in Algorithm 1, N will be evaluated on *children*$_\tau$(N) such that in the sum corresponding to val(N), the value of each removed branch (removed child), due to removing auxiliary variables, corresponds to the neutral element of addition, i.e., 0. Therefore, val(N) = val(N_τ), which concludes the proof.

Inspecting Algorithm 1, we see that we require two traversals of the original counting graph, one from Lines 2–8 and another one in Line 9 where we remove the nodes that do not belong to the CCG. Runtime follows from the fact that we need to traverse Φ_Π twice. □

Lemma 3. For any given program Π where $C_i \in \mathit{cycles}(\Pi)$ and $1 \leq i \leq n$, we have that $\mathcal{AS}(\Pi) = \mathcal{S}(\Pi \cup \{\lambda(C_1), \ldots, \lambda(C_n)\})$.

Proof. Recall that $\mathcal{AS}(\Pi) \subseteq \mathcal{S}(\Pi)$. However, supported models – in particular those that are not answer sets – might contain a cycle $C = \{c_0, \ldots, c_m\} \in \mathit{cycles}(\Pi)$ without external support from $ES(C) = \{s_0, \ldots, s_k\}$, which are precisely those supported models we exclude by adding a rule

$$\bot \leftarrow c_0, \ldots, c_m, \neg s_0, \ldots, \neg s_k$$

in the form of unsupported constraints $\lambda(C)$ to Π for each $C \in \mathit{cycles}(\Pi)$. This ensures that cyclic atoms are not present without external support in any supported model, which provides us with supported models that are answer sets. □

Theorem 1. Let Π be a program, $\mathit{cycles}(\Pi) = \{C_1, \ldots, C_n\}$, and further $U \coloneqq \{\lambda(C_1), \ldots, \lambda(C_n)\}$ be the set of all unsupported constraints of Π. Then, $|\mathcal{S}(\Pi^L \cup U)| = \sum_{i=0}^{n} (-1)^i \sum_{\Gamma \in \Lambda_i(\Pi)} |\mathcal{S}(\Pi^L \cup \Gamma)|$.

Proof (Sketch). We proceed by induction on $|\mathit{cycles}(\Pi)|$.

Induction Base Case: We assume that $|\mathit{cycles}(\Pi)| = 0$. Then, since Π admits no positive cycle in $DP(\Pi)$, we have $\mathcal{AS}(\Pi^L) = \mathcal{S}(\Pi^L)$, and therefore $|\mathcal{AS}(\Pi^L)| = |\mathcal{S}(\Pi^L)|$.

Induction Hypothesis (IH): We assume that the proposition holds for every program Π with $|cycles(\Pi)| < o$.

Induction Step: We need to show that the result holds for a program Π with $|cycles(\Pi)| = o + 1$. Let $C' \in cycles(\Pi)$ be a cycle. We define $U_o \coloneqq \{\lambda(C_1), \ldots, \lambda(C_o)\}$ for any $\{C_1, \ldots, C_o\} \subseteq cycles(\Pi)$ such that $|U_o| = o$ with $C_i \neq C'$ for $C_i \in \{C_1, \ldots, C_o\}$. Then, by IH, we have that

$$x \coloneqq |\mathcal{S}(\Pi^L \cup U_o)| = \sum_{i=0}^{o}(-1)^i \sum_{\Gamma \in \Lambda_i(\Pi), \lambda(C') \notin \Gamma} |\mathcal{S}(\Pi^L \cup \Gamma)|$$

To x, the formula $\sum_{i=0}^{o+1}(-1)^i \sum_{\Gamma \in \Lambda_i(\Pi)} |\mathcal{S}(\Pi^L \cup \Gamma)|$ adds $|\mathcal{S}(\Pi \cup \lambda(C'))|$. However, this formula then subtracts supported models satisfying both constraints $\{\lambda(C'), \lambda(C'')\}$ with one of the cycles $\lambda(C'') \in U_o$ twice, which require to be added back. Thus, we proceed by adding back supported models satisfying unsupported constraints of C' with two other cycles, which again have to be subtracted in the next step. In turn, the application of the inclusion-exclusion principle ensures that

$$\sum_{i=0}^{o+1}(-1)^i \sum_{\Gamma \in \Lambda_i(\Pi)} |\mathcal{S}(\Pi^L \cup \Gamma)| = x + \sum_{i=0}^{o+1}(-1)^i \sum_{\Gamma \in \Lambda_i(\Pi), \lambda(C') \in \Gamma} |\mathcal{S}(\Pi^L \cup \Gamma)|.$$

□

Lemma 4. Let Π be a program and L be assumptions. If $a_i^L = a_{i+1}^L$ for some integer $i \geq 0$, then $a_i^L = |\mathcal{AS}(\Pi^L)|$.

Proof. Suppose $a_i^L = a_{i+1}^L$, then $\sum_{\Gamma \in \Lambda_{i+1}(\Pi)} |\mathcal{S}(\Pi^L \cup \Gamma)| = 0$. It easy to see that therefore no further combination of unsupported constraints with set L of assumptions where we combine unsupported constraints of cycles that occur in subsets of $cycles(\Pi)$ with cardinality $j > i + 1$ points to any supported model. In other words, we have $\forall j > i : \sum_{\Gamma \in \Lambda_j(\Pi)} |\mathcal{S}(\Pi^L \cup \Gamma)| = 0$, which concludes the proof. □

Corollary 3. Let $n = |cycles(\Pi)|$. Then $a_n^L = |\mathcal{AS}(\Pi^L)|$ for program Π and assumptions L.

Proof. Note that $a_n^L = \sum_{i=0}^{n}(-1)^i \sum_{\Gamma \in \Lambda_i(\Pi)} |\mathcal{S}(\Pi^L \cup \Gamma)|$. The result follows from Theorem 1 by using Lemma 3. □