Proof of the Soundness and Completeness of ILASP2

Mark Law, Alessandra Russo and Krysia Broda

July 3, 2015

Abstract

In this document we provide the proofs of soundness and completeness of the ILASP2 (Inductive Learning of Answer Set Programs) algorithm which were omitted from the paper *Learning Weak Constraints in Answer Set Programming*. ILASP2 can learn ASP programs with normal rules, constraints, choice rules and weak constraints.

1 Introduction

In the next section, we recall the necessary definitions from the paper and give some extra definitions omitted from the paper. In section 3 we introduce some extra notation used in this document. In section 4 we give some lemmas necessary for the proofs.

The main content in the document is in sections 5, 6 and 7 where we describe various parts of our meta representation (how we represent orderings, the construction of T_{meta} and the construction of $VR_{meta}(T)$) and prove necessary results about them. In section 8, we give the proof of the soundness and completeness of our algorithm, $ILASP2$, and in section 8, give an example encoding.

The second part of the document concerns some theoretical properties of our learning framework. In section 9, we prove sufficient and necessary conditions for there to be a solution to an ILP_{LOAS} task and in section 10, we prove the complexity of deciding the existence of a solution to a task.

2 Definitions

Definition 2.1 [1]**.** A *weak constraint* is of the form: $:\sim\ b_1, \ldots, b_n,$ not $c_1, \ldots,$ not $c_m.[w@l, t_1, \ldots t_m]$ where $b_1, \ldots, b_n,\ c_1, \ldots, c_m$ are atoms, w and l are terms specifying the *weight* and the *level*, and $t_1, \ldots, t_m$ are terms. A weak constraint W is said to be *safe* if every variable occurring anywhere in W occurs in at least one positive literal in the body of W.

Definition 2.2. Given a program P and an answer set A of P, $weak(P, A)$ is the list of ground terms $(w, l, t_1, \ldots, t_n)$ for which there exist at least one weak constraint $:\sim body.[w@l, t_1, \ldots, t_n]$ in $ground(P)$ such that $body$ is satisfied by A.

Definition 2.3. Given a program P and an answer set A in $AS(P)$, $weak(P, A) = \{(w, l, t_1, \ldots, t_o) \mid :\sim b_1, \ldots, b_n,$ $not\ c_1, \ldots, not\ c_m.[w@l, t_1, \ldots, t_o] \in ground(P)$ and A extends $\langle\{b_1, \ldots, b_n\}, \{c_1, \ldots, c_m\}\rangle$ and $w \in \mathbb{Z}\}$. For each level l, $P^l_A = \sum_{(w,l,t_1,\ldots,t_o)\in weak(P,A)} w$.

Definition 2.4. For any $A_1, A_2 \in AS(P)$, A_1 *dominates* A_2 (written $A_1 \succ_P A_2$) iff $\exists l$ such that $P^l_{A_1} < P^l_{A_2}$ and $\forall m > l, P^m_{A_1} = P^m_{A_2}$. An answer set $A \in AS(P)$ is *optimal* if it is not dominated by any other $A_2 \in AS(P)$.

As we only use choice rules rather than programs allowing aggregates in the body of a rule, we are able to present the slightly simplified semantics of ASP presented in [2]. For the subset of ASP programs that we consider, this semantics is equivalent to the full semantics given in [3].

Definition 2.5. The reduct of a program P with respect to an interpretation I, is constructed in the following 4 steps.

1. Remove any rule whose body contains `not a` for some $\texttt{a} \in I$ and remove any negative literals from the remaining rules.

2. For any constraint R, $\texttt{:-}body(R)$, replace R with $\bot \texttt{:-} body^+(R)$.

3. For any choice rule R, $\texttt{l}\{\texttt{h}_1;\ldots;\texttt{h}_n\}\texttt{u} \texttt{:-} body(R)$ such that $l \leq |I \cap \{h_1, \ldots, h_n\}| \leq u$, replace R with the set of rules $\{\texttt{h}_i \texttt{:-} body^+(R) \mid h_i \in I \cap \{h_1 \ldots h_n\}\}$.

4. For any remaining choice rule R, $\texttt{l}\{\texttt{h}_1;\ldots;\texttt{h}_n\}\texttt{u} \texttt{:-} body(R)$, replace R with the constraint $\bot \texttt{:-} body^+(R)$.

P^X is a definite logic program, containing one additional atom $\bot$ which cannot appear in an interpretation of P. The idea is that if $P^X \models \bot$ then X is not an *Answer Set*.

Definition 2.6. Given any program P, X is an Answer Set of P if and only if $X = M((ground(P))^X)$ (where $M(P)$ denotes the least Herbrand model of P).

Definition 2.7 [4]**.** A *Learning from Answer Sets* task is a tuple $T = \langle B, S_{LAS}(M_h, M_b), E^+, E^- \rangle$ where B is the background knowledge, $S_{LAS}(M_h, M_b)$ is the search space defined by a language bias $M = \langle M_h, M_b \rangle$, E^+ and E^- are sets of partial interpretations called, respectively, positive and negative examples. An hypothesis H is an *inductive solution* of T, written $H \in ILP_{LAS}(T)$, if and only if $H \subseteq S_{LAS}(M_h, M_b)$; $\forall e^+ \in E^+ \; \exists A \in AS(B \cup H)$ such that A extends e^+; and finally, $\forall e^- \in E^- \; \nexists A \in AS(B \cup H)$ such that A extends e^-.

Definition 2.8. A mode bias with ordering is a tuple $M = \langle M_h, M_b, M_o, M_w, l_{max} \rangle$, where M_h and M_b are respectively head and body declarations, M_o is a set of mode declarations for body literals in weak constraints, M_w is a set of integers and l_{max} is a positive integer. The search space S_M is the set of rules R that satisfy one of the conditions:

- $R \in S_{LAS}(M_h, M_b)$.
- R is a safe weak constraint $:\sim b_1, \ldots, b_i, \texttt{not } b_{i+1}, \ldots, \texttt{not } b_j.[w@l, t_1, \ldots, t_n]$ such that $\forall k \in [1, j]$ b_k is compatible with M_o; $t_1, \ldots, t_n$ is the set of terms in $b_1, \ldots, b_j$; $w \in M_w$, $l \in [0, l_{max})$.

Definition 2.9. An *ordering example* is a tuple $o = \langle e_1, e_2 \rangle$ where e_1 and e_2 are partial interpretations. An ASP program P *bravely respects* o iff $\exists A_1, A_2 \in AS(P)$ such that A_1 extends e_1, A_2 extends e_2 and $A_1 \succ_P A_2$. P *cautiously respects* o iff $\forall A_1, A_2 \in AS(P)$ such that A_1 extends e_1 and A_2 extends e_2, it is the case that $A_1 \succ_P A_2$.

Definition 2.10. A *Learning from Ordered Answer Sets* task is a tuple $T = \langle B, S_M, E^+, E^-, O^b, O^c \rangle$ where B is an ASP program, called the background knowledge, S_M is the search space defined by a mode bias with ordering M, E^+ and E^- are sets of partial interpretations called, respectively, positive and negative examples, and O^b and O^c are sets of ordering examples over E^+ called brave and cautious orderings. A hypothesis $H \subseteq S_M$ is in $ILP_{LOAS}(T)$, the *inductive solutions* of T, if and only if:

1. Let M_h and M_b be as in M and H' be the subset of H with no weak constraints. $H' \in ILP_{LAS}(\langle B, S_{LAS}(M_h, M_b), E^+, E^- \rangle)$
2. $\forall o \in O^b$ $B \cup H$ bravely respects o
3. $\forall o \in O^c$ $B \cup H$ cautiously respects o

Definition 2.11. Let $T = \langle B, S_M, E^+, E^-, O^b, O^c \rangle$ be an ILP_{LOAS} task. Any $H \subseteq S_M$ is a *positive hypothesis* iff $\forall e \in E^+ \; \exists A \in AS(B \cup H)$ such that A extends e, and $\forall o \in O^b$ $H \cup B$ bravely respects o. The set of positive hypotheses of T is denoted $\mathcal{P}(T)$.

Definition 2.12. A positive hypothesis H is a *violating hypothesis* of $T = \langle B, S_M, E^+, E^-, O^b, O^c \rangle$, written $H \in \mathcal{V}(T)$, iff at least one of the following cases is true:

- $\exists e^- \in E^-$ and $\exists A \in AS(B \cup H)$ such that A extends e^-. In this case we call A a violating interpretation of T and write $\langle H, A \rangle \in \mathcal{VI}(T)$.
- $\exists A_1, A_2 \in AS(B \cup H)$ and $\exists \langle e_1, e_2 \rangle \in O^c$ such that A_1 extends e_1, A_2 extends e_2 and $A_1 \not\succ_P A_2$ with respect to $B \cup H$. In this case, we call $\langle A_1, A_2 \rangle$ a *violating pair* of T and write $\langle H, \langle A_1, A_2 \rangle \rangle \in \mathcal{VP}(T)$.

Definition 2.13. Let T be an ILP_{LOAS} task, VI and VP (resp.) be sets of violating interpretations and pairs of interpretations, and B be the background knowledge. Any $H \in \mathcal{P}(T)$ is a *remaining hypothesis* of T with respect to $VI \cup VP$ iff $VI \cap AS(B \cup H) = \emptyset$ and $\forall \langle I_1, I_2 \rangle \in VP$ if $I_1, I_2 \in AS(B \cup H)$ then $I_1 \succ_{B \cup H} I_2$. A remaining hypothesis H is a *remaining violating hypothesis* iff $\exists R$ such that $\langle H, R \rangle \in \mathcal{VI}(T) \cup \mathcal{VP}(T)$.

We will call those violating hypotheses which are not remaining violating hypotheses, *known violating hypotheses* (they are already ruled out by some known violating reason).

Finally, Algorithm 1 is our algorithm, ILASP2.

Algorithm 1 ILASP2

```
procedure ILASP2(T)
    VR = []
    solution = solve(T_meta ∪ VR_meta(T))
    while solution ≠ nil && solution.optimality%2 == 0 do
        A = solution.answer_set
        if v_i ∈ A then
            VR += M_vi^{-1}(A)
        else if ∃t1, t2 such that v_p(t1, t2) ∈ A then
            VR += M_vp^{-1}(A)
        end if
        solution = solve(T_meta ∪ VR_meta(T))
    end while
    return {M_hyp^{-1}(A) | A ∈ AS*(T_meta ∪ VR_meta(T))}
end procedure
```

3 Extra notation

Definition 3.1. For any ASP program P, predicate name *pred* and term `term` we will write $reify(P, pred, term)$ to mean the program constructed by replacing every atom $\mathtt{a} \in P$ by $\mathtt{pred(a, term)}$. We will use the same notation for sets of literals/partial interpretations, so for a set S: $reify(S, pred, term) = \{\mathtt{pred(atom, term)} : \mathtt{atom} \in S\}$.

Definition 3.2. For any ASP program P and any atom `a`, $append(P, a)$ is the program constructed by appending `a` to every rule in P.

Definition 3.3. Given a set of rules S_M such that all rules $R \in S_M$ have identifiers R_{id} and any program P such that:

1. Some rules in P contain the atom $\mathtt{in\_hyp(R_{id})}$ for some rules $R \in S_M$
2. No rule contains more than one instance of *in_hyp* (or a non ground atom with the predicate *in_hyp*).
3. P contains the choice rule $\{\mathtt{in\_hyp(R_{id})} : R \in S_M)\}$.
4. No other rule contains an *in_hyp* atom in the head.

For any $H \subseteq S_M$, $P[H]$ is constructed in three steps:

1. Remove the choice rule $\{\mathtt{in\_hyp(R_{id})} : R \in S_M)\}$.
2. Remove any rule with $\mathtt{in\_hyp(R_{id})}$ in its body st $R \notin H$.
3. Remove the *in_hyp* atoms from the remaining rules.

Definition 3.4. Given a set of rules S_M such that all rules $R \in S_M$ have identifiers R_{id} and an interpretation I:

$\mathcal{M}^{-1}_{hyp}(I) = \{R \mid \mathtt{in\_hyp(R_{id})} \in I, R \in S_M\}$

Definition 3.5. Given an interpretation I:

$\mathcal{M}_{vi}^{-1}(I) = \{\mathtt{atom} \mid \mathtt{in\_as(atom, n)} \in I\}$

Definition 3.6. Given an interpretation I and two ground terms $\mathtt{t_1}$ and $\mathtt{t_2}$:

$\mathcal{M}_{vp}^{-1}(I, t_1, t_2) = \langle\{\mathtt{atom} \mid \mathtt{in\_as(atom, t_1)} \in I\}, \{\mathtt{atom} \mid \mathtt{in\_as(atom, t_2)} \in I\}\rangle$

Definition 3.7. Given an interpretation I which contains the atom $\mathtt{v\_p(t_1, t_2)}$ for some ground terms $\mathtt{t_1}$ and $\mathtt{t_2}$:

$\mathcal{M}_{vp}^{-1}(I) = \mathcal{M}_{vp}^{-1}(I, t_1, t_2)$.

If I contains more than one atom $v\_p(t1, t2)$, then $\mathcal{M}_{vp}^{-1}(I)$ will choose a pair of terms; it is not gauranteed which. If I contains no such atom, then $\mathcal{M}_{vp}^{-1}(I)$ is undefined.

Definition 3.8. For any ASP program P, $normal(P)$ is the set of all normal rules in P.

Definition 3.9. For any ASP program P, $constraints(P)$ is the set of all (hard) constraints in P.

Definition 3.10. For any ASP program P, $choice(P)$ is the set of all choice rules in P.

Definition 3.11. For any ASP program P, $weak(P)$ is the set of all weak constraints in P.

Definition 3.12. For any ASP program P, $non\_weak(P) = normal(P) \cup constraints(P) \cup choice(P)$.

4 Lemmas

In this document $ground(P)$ refers to all ground instances of rules in P, rather than the "relevant" grounding produced by ASP solvers.

Lemma 4.1. Let P and Q be ASP program such that $atoms(ground(P)) \cap atoms(ground(Q)) = \emptyset$.

Then $AS(P \cup Q) = \{A_1 \cup A_2 \mid A_1 \in AS(P), A_2 \in AS(Q)\}$.

Hence $P \cup Q$ is satisfiable if and only if P and Q are both satisfiable.

Corollary 4.2. Let $last\_terms(P)$ be a function which extracts the last argument from each atom in a program P.

Let P and Q be ASP programs (which contain no atoms of arity 0) st each term in $last\_terms(P)$ and $last\_terms(Q)$ is ground and $last\_terms(P) \cap last\_terms(Q) = \emptyset$

Then $AS(P \cup Q) = \{A_1 \cup A_2 \mid A_1 \in AS(P), A_2 \in AS(Q)\}$.

Hence $P \cup Q$ is satisfiable if and only if P and Q are both satisfiable.

Corollary 4.3. Let p_1 and p_2 be two distinct predicate names and $Terms_1$ and $Terms_2$ be disjoint sets of ground terms. Let P be any ASP program.

Let $Q = append(reify(P, p_1, X), p_2(X))$

Then $AS(Q \cup \{\mathtt{p_2(t)} \mid \mathtt{t} \in Terms_1 \cup Terms_2\}) = \{A_1 \cup A_2 \mid i \in \{1, 2\}, A_i \in AS(Q \cup \{\mathtt{p_2(t)} \mid \mathtt{t} \in Terms_i\})\}$.

Lemma 4.4. For any program $P \cup Q$ in which the atom $\mathtt{a}$ does not occur:

$AS(append(P, a) \cup Q \cup \{\mathtt{a.}\}) = \{A \cup \{\mathtt{a.}\} : A \in AS(P \cup Q)\}$

Lemma 4.5. For any ASP program P any partial interpretation $E = \langle E^{inc}, E^{exc}\rangle$ and any ground atom a which does not appear in P or E.

$AS(P \cup \{\mathtt{a}\ \mathtt{:-} \bigwedge_{\mathtt{lit} \in \mathtt{E^{inc}}} \mathtt{lit}, \bigwedge_{\mathtt{lit} \in \mathtt{E^{exc}}} \mathtt{not\ lit.} \quad \mathtt{:-\ not\ a.}\}) = \{A \cup \{\mathtt{a}\} \mid A \in AS(P) \text{ st } A \text{ extends } E\}$.

Lemma 4.6. Let R be the rule $\mathtt{h\ :\!-\ b_1, \ldots, b_n}, \#\text{sum}\{\mathtt{s_1 = w_1, \ldots, s_m = w_m}\} < \mathtt{0}$, where $\mathtt{h}$, each of the $\mathtt{b_i}$'s and $\mathtt{s_i}$'s are ground atoms and each $\mathtt{w_i}$ is an integer.

For any set of (ground) facts F (such that $\mathtt{h} \notin F$):

$\mathtt{h} \in M(R \cup F)$ (the only Answer Set of this program) iff $\forall 0 < i \leq n$: $b_i \in F$ and $\sum_{s_j \in F}(w_j) < 0$

Corollary 4.7. Let R be the rule $\mathtt{h} \texttt{:-} \mathtt{b_1}, \ldots, \mathtt{b_n}, \#\mathrm{sum}\{\mathtt{s_1} = \mathtt{w_1}, \ldots, \mathtt{s_m} = \mathtt{w_m}\} < \mathtt{0}$, where $\mathtt{h}$, each of the $\mathtt{b_i}$'s and $\mathtt{s_i}$'s are atoms and each $\mathtt{w_i}$ is an integer.

For any set of (ground) facts F (such that $\mathtt{h} \not\in F$):

1. If $\left(\sum_{s \in F, \exists \theta st\ s = \mathtt{s_i}\theta} \mathtt{w_i}\right) < 0$ then $AS(F \cup R) = AS(F \cup \{\mathtt{h} \texttt{:-} \mathtt{b_1}, \ldots, \mathtt{b_n}\})$
2. Otherwise $AS(F \cup R) = AS(F)$.

Lemma 4.8. Given a set of rules S_M such that all rules $R \in S_M$ have identifiers R_{id} and any program P such that:

1. Some rules in P contain the atom $\mathtt{in\_hyp(R_{id})}$ for some rules $R \in S_M$
2. No rule contains more than one instance of *in_hyp* (or a non ground atom with the predicate *in_hyp*).
3. P contains the choice rule $\{\mathtt{in\_hyp(R_{id})} : R \in S_M)\}$.
4. No other rule contains an *in_hyp* atom in the head.

Given any $H \subseteq S_M$,

$AS(P[H]) = \{A \backslash \{\mathtt{in\_hyp(R_{id})} \mid R \in H\} \mid A \in AS(P) \text{ st } \mathcal{M}^{-1}_{hyp}(A) = H\}$.

Part I

ILASP2

5 Repesentation of weak constraints

Before defining the main meta translations, we describe how we represent weak constraints. Essentially, we translate them to normal rules such that each head corresponds to an element which could occur in the set $weak(P, A)$ (which is used to define the semantics of weak constraints). We then add rules to determine which interpretations dominate other interpretations. In this section we define this representation and prove some of the properties we require in later sections.

5.1 Meta Level Representation

Definition 5.1. Let p_1 and p_2 be distinct predicate names and t be a term. Given R as a weak constraint $:\sim \mathtt{b_1}, \ldots, \mathtt{b_m}, \text{not } \mathtt{c_1}, \ldots, \text{not } \mathtt{c_1}.[\mathtt{wt@lev}, \mathtt{t_1}, \ldots, \mathtt{t_n}]$, $meta_{weak}(R, p_1, p_2, t)$ is the rule:

```
w(wt, lev, args(t1, ... ,tn), t) :-
       p2(t), p1(b1,t), ... p1(bm, t),
       not p1(c1,t), ... , not p1(cl,t).
```

For a set of weak constraints W, $meta_{weak}(W, p_1, p_2, t) = \{meta_{weak}(R, p_1, p_2, t) \mid R \in W\}$.

Example 5.2. Let W be the set of weak constraints:

```
:~ p(X, Y), not q(X).[2@1, X, Y]
:~ p(X, Y).[-1@2, X]
```

Then $meta_{weak}(W, in\_as, as, V)$ is the program:

```
w(2, 1, args(X, Y), V) :- as(V), in_as(p(X, Y), V), not in_as(q(X), V).
w(-1, 2, args(X), V) :- as(V), in_as(p(X, Y), V).
```

The intuition of this meta encoding (when used with these two predicates) is that for each answer set V of some program P, if in_as defines the elements of V, then $\mathtt{w(W, L, args(t_1, \ldots, t_n), V)}$ is true if and only if $(W, L, t_1, \ldots, t_n) \in weak(P, V)$.

Now that we have defined the predicate w to represent $weak(P, A)$ for each answer set A, we can use some additional rules to determine, given two interpretations, whether one dominates another.

Definition 5.3. Given any two terms $t1$ and $t2$, $dominates(t1, t2) = dom_1(t1, t2) \cup dom_2(t1, t2)$ where:

$dom_1(t1, t2)$:

```
dom_lv(t1,t2,L) :- lv(L), #sum{w(W,L,A,t1)=W, w(W,L,A,t2)=-W} < 0.
non_dom_lv(t1,t2,L) :- lv(L), #sum{w(W,L,A,t2)=W, w(W,L,A,t1)=-W} < 0.
```

$dom_2(t1, t2)$:

```
non_bef(t1,t2,L) :- lv(L), lv(L2), L < L2, non_dom_lv(t1,t2,L2).
dom(t1,t2) :- dom_lv(t1,t2,L), not non_bef(t1,t2,L).
```

The intuition is that $dom(as1, as2)$ (where $as1$ and $as2$ represent two answer sets) should be true if and only if $as1$ dominates $as2$.

5.2 Properties

As the meta level programs considered in this section are each clearly stratified, for each program P there is a unique answer set (equal to the minimal Herbrand model $M(P)$). For the rest of this section, therefore, we shall refer to $M(P)$ rather than $AS(P)$.

This section is devoted to proving the following lemma (which proves that our representation of weak constraints is correct).

Lemma 5.4. Let $t1$ and $t2$ be two distinct terms and L be a set of integers. Let I_1 and I_2 be interpretations and P be an ASP program.

For any predicates p_1 and p_2 (not used in the rest of the program):

$$M\left(\begin{array}{l} dominates(t_1,t_2)\cup \\ \{\texttt{lv(l)}.\mid l\in L\}\cup \end{array} \left\{\begin{array}{l} meta_{weak}(weak(P),p_1,p_2,t_i) \\ \quad\cup\, reify(I_1,p_1,t_1) \\ \quad\cup\,\{\texttt{p}_2(\texttt{t}_\texttt{i})\} \end{array}\right\}\right) \text{ contains the atom } \texttt{dom}(\texttt{t}_1,\texttt{t}_2) \text{ if and only if } I_1 \succ_P I_2.$$

In order to do this, we need to first prove several intermediate lemmas. Some of the proofs of these lemmas have been omitted from the main document, but can be found in the appendix.

Lemma 5.5. Let l be a constant, t_1 and t_2 be two distinct ground terms and *head* be an atom.

Let R be the rule $\texttt{head :- b}_1,\ldots,\texttt{b}_\texttt{n}, \#\texttt{sum}\{\texttt{w(W,l,A,t}_1) = \texttt{W}, \texttt{w(W,l,A,t}_2) = -\texttt{W}\} < 0$ and F be a set of (ground) facts of the predicate $w/4$ and (where *head* has a different predicate name to w)

For $i\in\{1,2\}$, let $\mathcal{S}_i = (\sum_{w(weight,l,args,t_i)\in F} weight)$

1. If $\mathcal{S}_1 \geq \mathcal{S}_2$ then $M(F\cup R) = M(F)$
2. If $\mathcal{S}_1 < \mathcal{S}_2$ then $M(F\cup R) = F\cup\{\texttt{head :- b}_1,\ldots,\texttt{b}_\texttt{n}\}$)

Lemma 5.6. Let t_1 and t_2 be two distinct terms and L be a set of integers. F is a set of (ground) facts of the predicate $w/4$.

$$\begin{array}{l} M(dom_1(t_1,t_2)\cup\{\texttt{lv(l)}\mid l\in L\}\cup F) \\ \quad = M\left(\left\{\begin{array}{l} \texttt{dom\_lv(t}_1,\texttt{t}_2,\texttt{l) :- }\#\texttt{sum}\{\texttt{w(W,L,A,t}_1) = \texttt{W}, \texttt{w(W,L,A,t}_2) = -\texttt{W}\} < 0. \\ \texttt{non\_dom\_lv(t}_1,\texttt{t}_2,\texttt{l) :- }\#\texttt{sum}\{\texttt{w(W,L,A,t}_2) = \texttt{W}, \texttt{w(W,L,A,t}_1) = -\texttt{W}\} < 0. \end{array}\right\}\cup F\right)\cup\{\texttt{lv(l)}\mid l\in L\} \end{array}$$

Proof. Follows from repeated use of lemma **??** over the facts in $\{\texttt{lv(l)}\mid l\in L\}$. ☐

Proposition 5.7. Let t_1 and t_2 be two distinct terms and L be a set of integers.

Let W_1 be a set of facts which are ground instances of the form $\texttt{w(W,L,A,t}_1)$ and W_2 be a set of facts which are ground instances of the form $\texttt{w(W,L,A,t}_2)$.

For $i\in\{1,2\}$ and $l\in L$, let $\mathcal{S}^l_i = (\sum_{\texttt{w(weight,l,args,t}_\texttt{i})\in W_i} weight)$

$$\begin{array}{l} M(dom_1(t_1,t_2)\cup\{\texttt{lv(l)}\mid l\in L\}\cup W_1\cup W_2) \\ \quad = \{\texttt{dom\_lv(t}_1,\texttt{t}_2,\texttt{l)}\mid l\in L, \mathcal{S}^l_1 < \mathcal{S}^l_2\}\cup\{\texttt{non\_dom\_lv(t}_1,\texttt{t}_2,\texttt{l)}\mid l\in L, \mathcal{S}^l_2 > \mathcal{S}^l_2\}\cup W_1\cup W_2\cup\{\texttt{lv(l)}\mid l\in L\} \end{array}$$

Proof. Follows directly from lemma 5.5 and lemma 5.6. ☐

Recall that for any interpretation I, program P and integer l, P^l_I is the sum of the weights w such that $(w,l,\ldots)\in weak(P,I)$. Recall also that this is used to determine which interpretations dominate each other. Lemma 5.8 shows that if we represent $weak(W,I_1)$ and $weak(W,I_2)$ as facts, we can use dom_1 to capture at each level whether I_1 or I_2 dominates at that level.

Lemma 5.8. Let t_1 and t_2 be two distinct terms and L be a set of integers. Let I_1 and I_2 be interpretations and W a set of weak constraints.

For $i\in\{1,2\}$ let $W_i = \{\texttt{w(wt,l,args(a}_1,\ldots,\texttt{a}_\texttt{n}),\texttt{t}_\texttt{i})\mid (wt,l,a_1,\ldots,a_n)\in weak(W,I_i)\}$

$$\begin{array}{l} M(dom_1(t_1,t_2)\cup\{\texttt{lv(l)}\mid l\in L\}\cup W_1\cup W_2) \\ \quad = \{\texttt{dom\_lv(t}_1,\texttt{t}_2,\texttt{l)}\mid l\in L, W^l_{I_1} < W^l_{I_2}\}\cup\{\texttt{non\_dom\_lv(t}_1,\texttt{t}_2,\texttt{l)}\mid l\in L, W^l_{I_1} > W^l_{I_2}\}\cup W_1\cup W_2\cup\{\texttt{lv(l)}\mid l\in L\} \end{array}$$

Proof.

For $i \in \{1,2\}$ and $l \in L$, let $\mathcal{S}_i^l = (\sum_{\mathtt{w(weight,l,args,t_i)} \in W_i} weight)$

$M(dom_1(t_1,t_2) \cup \{lv(l) \mid l \in L\} \cup W_1 \cup W_2)$

$= \{\mathtt{dom\_lv(t_1,t_2,l)} \mid l \in L, \mathcal{S}_1^l < \mathcal{S}_2^l\} \cup \{\mathtt{non\_dom\_lv(t_1,t_2,l)} \mid l \in L, \mathcal{S}_1^l > \mathcal{S}_2^l\} \cup W_1 \cup W_2 \cup \{\mathtt{lv(l)} \mid l \in L\}$ (by proposition 5.7)

$= \{\mathtt{dom\_lv(t1,t2,l)} \mid l \in L, (\sum_{w(wt,l,a_1,\ldots,a_n,t_1) \in weak(W,I_1)} wt) < (\sum_{w(wt,l,a_1,\ldots,a_n,t_2) \in weak(W,I_2)} wt)\}$
$\cup \{\mathtt{non\_dom\_lv(t_1,t_2,l)} \mid l \in L, (\sum_{w(wt,l,a_1,\ldots,a_n,t_1)) \in weak(W,I_1)} wt) > (\sum_{w(wt,l,a_1,\ldots,a_n,t_2)) \in weak(W,I_2)} wt)\}$
$\cup W_1 \cup W_2 \cup \{\mathtt{lv(l)} \mid l \in L\}$ (by definition of W_1 and W_2)

$= \{\mathtt{dom\_lv(t_1,t_2,l)} \mid l \in L, W_{I_1}^l < W_{I_2}^l\} \cup \{\mathtt{non\_dom\_lv(t_1,t_2,l)} \mid l \in L, W_{I_1}^l > W_{I_2}^l\} \cup W_1 \cup W_2 \cup \{\mathtt{lv(l)} \mid l \in L\}$ (by definition of $W_{I_1}^l$ and $W_{I_2}^l$)

□

Lemma 5.9. Let t_1 and t_2 be two distinct terms. Let I_1 and I_2 be interpretations and W a set of weak constraints and L be the set of levels in W.

For $i \in \{1,2\}$, let $W_i = \{w(weight, l, args(a_1,\ldots,a_n), t_i) \mid (weight, l, a_1, \ldots, a_n) \in weak(P, I_i)\}$

$I_1 \succ_W I_2$ if and only if $dom(t_1,t_2) \in M(dominates(t_1,t_2) \cup \{lv(l) \mid l \in L\} \cup W_1 \cup W_2)$

Proof.

Let $M_{dom} = M(dominates(t_1,t_2) \cup \{\mathtt{lv(l)} \mid l \in L\} \cup W_1 \cup W_2))$

$M_{dom} = M(dom_1(t_1,t_2) \cup dom_2(t_1,t_2) \cup \{\mathtt{lv(l)} \mid l \in L\} \cup W_1 \cup W_2)$ (by definition of *dominates*).

$= M(dom_2(t_1,t_2) \cup M(dom_1(t_1,t_2) \cup \{\mathtt{lv(l)} \mid l \in L\} \cup W_1 \cup W_2))$ (by the splitting set theorem).

$= M(dom_2(t_1,t_2) \cup \{\mathtt{dom\_lv(t_1,t_2,l)} \mid l \in L, (W_{I_1}^l) < (W_{I_2}^l)\} \cup \{\mathtt{non\_dom\_lv(t_1,t_2,l)} \mid l \in L, (W_{I_1}^l) > (W_{I_2}^l)\}$
$\cup W_1 \cup W_2 \cup \{\mathtt{lv(l)} \mid l \in L\})$ (by lemma 5.8).

Assume $\mathtt{dom(t1,t2)} \in M_{dom}$

$\Leftrightarrow \exists l \in L$ such that $\mathtt{dom\_lv(t_1,t_2,l)} \in M_{dom}$ and $\mathtt{non\_bef(t_1,t_2,l)} \notin M_{dom}$

$\Leftrightarrow \exists l \in L$ such that $(W_{I_1}^l) < (W_{I_2}^l)$ and $\mathtt{non\_bef(t_1,t_2,l)} \notin M_{dom}$

$\Leftrightarrow \exists l \in L$ such that $(W_{I_1}^l) < (W_{I_2}^l)$ and $\nexists l_2 \in L$ such that $l < l_2$ and $\mathtt{non\_dom\_lv(t_1,t_2,l_2)} \in M_{dom}$

$\Leftrightarrow \exists l \in L$ such that $(W_{I_1}^l) < (W_{I_2}^l)$ and $\nexists l_2 \in L$ such that $l < l_2$ and $(W_{I_1}^{l_2}) > (W_{I_2}^{l_2})$

$\Leftrightarrow \exists l \in L$ such that $(W_{I_1}^l) < (W_{I_2}^l)$ and $\forall l_2 \in L$ such that $l < l_2$, $(W_{I_1}^{l_2}) \leq (W_{I_2}^{l_2})$

$\Leftrightarrow \exists l \in L$ such that $(W_{I_1}^l) < (W_{I_2}^l)$ and $\forall l_2 \in L$ such that $l < l_2$, $(W_{I_1}^{l_2}) = (W_{I_2}^{l_2})$ (as L is finite)

$\Leftrightarrow I_1 \succ_W I_2$ by the definition of $\succ$

□

Lemma 5.10. Let t be a ground term, W be a set of weak constraints and I be an interpretation. Let p_1 and p_2 be predicate names.

$M(meta_{weak}(W, p_1, p_2, t) \cup reify(I, p_1, t) \cup \{\mathtt{p_2(t)}\})$
$= \{\mathtt{w(wt, l, args(a_1, \ldots, a_n), t)} \mid (wt, l, a_1, \ldots, a_n) \in weak(P, I)\} \cup reify(I, p_1, t) \cup \{\mathtt{p_2(t)}\}$

Proof. Let $F = reify(I, pred_1, t) \cup \{pred_2(t)\}$

$M(meta_{weak}(W, p_1, p_2, t) \cup reify(I, p_1, t) \cup \{\mathtt{p_2(t)}\})$

$$= M\left(\left\{ \begin{array}{l} \mathtt{w(wt,l,args(a_1,\ldots,a_n),t)\text{:-}\,p_2(t),} \\ \quad \mathtt{p_1(b_1,t),\ldots,p_1(b_m,t),}\text{not } \mathtt{p_1(c_1,t),\ldots,}\text{not } \mathtt{p_1(c_1,t)} \end{array} \middle| \begin{array}{l} :\sim \mathtt{b_1,\ldots,b_m,}\text{not } \mathtt{c_1,\ldots,} \\ \quad \text{not } \mathtt{c_1.[wt@l,a_1,\ldots,a_n]} \end{array} \in W \right\} \cup F\right)$$

$$= \left\{ \mathtt{w(wt,l,args(a_1,\ldots,a_n),t)} \middle| \begin{array}{l} :\sim \mathtt{b_1,\ldots,b_m,}\text{not } \mathtt{c_1,\ldots,} \\ \quad \text{not } \mathtt{c_1.[wt@l,a_1,\ldots,a_n]} \end{array} \in W, \begin{array}{l} F \models \mathtt{p_1(b_1,t),\ldots,p_1(b_m,t),} \\ \text{not } \mathtt{p_1(c_1,t),\ldots,}\text{not } \mathtt{p_1(c_1,t)} \end{array} \right\} \cup F$$

$$= \left\{ \mathtt{w(wt, l, args(a_1, \ldots, a_n), t)} \middle| \begin{array}{l} :\sim \mathtt{b_1, \ldots, b_m}, \text{not } \mathtt{c_1}, \ldots, \\ \quad \text{not } \mathtt{c_1}.[\mathtt{wt@l, a_1, \ldots, a_n}] \end{array} \in W, \begin{array}{l} reify(I, p_1, t) \models \mathtt{p_1(b_1, t)}, \ldots, \mathtt{p_1(b_m, t)}, \\ \text{not } \mathtt{p_1(c_1, t)}, \ldots, \text{not } \mathtt{p_1(c_1, t)} \end{array} \right\} \cup F$$

$$= \left\{ \mathtt{w(wt, l, args(a_1, \ldots, a_n), t)} \middle| \begin{array}{l} :\sim \mathtt{b_1, \ldots, b_m}, \text{not } \mathtt{c_1}, \ldots, \\ \quad \text{not } \mathtt{c_1}.[\mathtt{wt@l, a_1, \ldots, a_n}] \end{array} \in W, \begin{array}{l} I \models b_1 \ldots, b_m, \\ \text{not } c_1, \ldots, \text{not } c_l \end{array} \right\} \cup F$$

$$= \{\mathtt{w(wt, l, args(a_1, \ldots, a_n), t)} | (wt, l, a_1, \ldots, a_n) \in weak(W, I)\} \cup F$$

□

We have now proved the properties required to prove lemma **??**

Lemma 5.4. Let $t1$ and $t2$ be two distinct terms and L be a set of integers. Let I_1 and I_2 be interpretations and P be an ASP program.

For any predicates p_1 and p_2 (not used in the rest of the program):

$$M\left(\begin{array}{l} dominates(t_1, t_2) \cup \\ \{\mathtt{lv(l)}. \mid l \in L\} \cup \end{array} \left\{ \begin{array}{l} meta_{weak}(weak(P), p_1, p_2, t_i) \\ \quad \cup\, reify(I_1, p_1, t_1) \\ \quad \cup\, \{\mathtt{p_2(t_i)}\} \end{array} \right\} \right)$$ contains the atom $\mathtt{dom(t_1, t_2)}$ if and only if $I_1 \succ_P I_2$.

Proof.

The program can be split into 3:

$Q_1 = meta_{weak}(weak(P), p_1, p_2, t_1) \cup reify(I_1, p_1, t_1) \cup \{\mathtt{p_2(t_1)}\}$

$Q_2 = meta_{weak}(weak(P), p_1, p_2, t_2) \cup reify(I_2, p_1, t_2) \cup \{\mathtt{p_2(t_2)}\}$

$Q_3 = dominates(t_1, t_2) \cup \{\mathtt{lv(l)} \mid l \in L\}$

Such that $M(Q) = M((M(Q_1) \cup M(Q_2)) \cup Q_3)$ (By the splitting set theorem).

$\Leftrightarrow$ For $i \in \{1, 2\}$, $M(Q_i) = \{\mathtt{w(wt, l, args(a_1, \ldots, a_n), t_i)} \mid (wt, l, a_1, \ldots, a_n) \in weak(weak(P), I_i)\}$ $\cup\, reify(I_i, p_1, t_i) \cup \{\mathtt{p_2(t_i)}\}$ (by lemma 5.10)

$\Leftrightarrow$ $Q_1 \cup Q_2 \cup Q_3$ has just one answer set A and A contains $\mathtt{dom(t_1, t_2)}$ if and only if $I_1 \succ_{weak(P)} I_2$ (by lemma 5.9).

As only weak constraints effect $\succ$, this means that A contains $\mathtt{dom(t_1, t_2)}$ if and only if $I_1 \succ_P I_2$.

□

6 Encoding the search for positive solutions: T_{meta}

Throughout this section we will refer to the ILP_{LOAS} task $T = \langle B, S_M, E^+, E^-, O^b, O^c \rangle$.

6.1 Meta Level Representation

Definition 6.1. The meta translation of the background knowledge B, written $meta(B)$, is the program:
$append(reify(non\_weak(B), in\_as, X), as(X)) \cup meta_{weak}(weak(B), in\_as, as, X)$.

Example 6.2. Consider the program B:

```
p(V) :- q(V), not r(V).
r(V) :- q(V), not p(V).
:~ p(V).[2@1, V]
```

$meta(B)$ is the program:

```
in_as(p(V), X) :- in_as(q(V), X), not in_as(r(V), X).
in_as(r(V), X) :- in_as(q(V), X), not in_as(p(V), X).
w(2, 1, args(V), X) :- in_as(p(V), X), as(X).
```

Definition 6.3. The meta translation of S_M written $meta(S_M)$ is the program:

$\{append(append(reify(R, in\_as, X), as(X)), in\_h(R_{id})) \mid R \in non\_weak(S_M)\}$
$\cup \{append(W, in\_h(W_{id})) \mid W \in meta_{weak}(weak(S_M), in\_as, as, X)\}$
$\cup \{:\sim \mathtt{in\_h}(\mathtt{R_{id}}).[2 * |\mathtt{R}|@0, \mathtt{R_{id}}] \mid R \in S_M\}$
$\cup \{\ \{\mathtt{in\_h}(\mathtt{R_{id}}) : \mathtt{R} \in \mathtt{S_M}\}.\ \}$

Example 6.4. Let S_M be:

```
p(V) :- not q(V), r(V).
:~ p(V).[2@1,V]
```

Then $meta(S_M)$ is the program:

```
in_as(p(V), X) :- not in_as(q(V), X), in_as(r(V), X), as(X), in_h(r1).
w(2, 1, args(V), X) :- in_as(p(V), X), as(X), in_h(r2).
{ in_h(r1), in_h(r2) }.
:~ in_h(r1).[6@0]
:~ in_h(r2).[2@0]
```

This definition will most likely move to the first proof section.

Definition 6.5. Given any term t and any positive example e, $cover(e, t)$ is the program:

$\{cov(t) \leftarrow \bigwedge_{l \in e^{inc}} l \wedge \bigwedge_{l \in e^{inc}} \text{not } l;\ \leftarrow \text{not } cov(t)\}$

Definition 6.6. For any $e \in E^+$, $meta(e)$ is the program: $cover(reify(e, in\_as, e_{id}), e_{id}) \cup \{\mathtt{as}(\mathtt{e_{id}}).\}$. Furthermore, we write $meta(E^+)$ to denote the program $\bigcup_{e \in E^+} meta(e)$.

Example 6.7. Consider $E^+ = \left\{ \begin{array}{l} \langle \{p, q\}, \{r\} \rangle, \\ \langle \{q, r\}, \{p\} \rangle \end{array} \right\}$. $meta(E^+)$ is the program:

```
cov(e1) :- in_as(p, e1), in_as(q, e1), not in_as(r, e1).
:- not cov(e1).
as(e1).
cov(e2) :- in_as(q, e2), in_as(r, e2), not in_as(p, e2).
:- not cov(e2).
as(e2).
```

Definition 6.8. For any $e \in E^-$, $meta(e)$ is the program: $\{\texttt{v\_i :-} \bigwedge_{l \in e^{inc}} \texttt{in\_as(l,n)} \wedge \bigwedge_{l \in e^{exc}} \text{not } \texttt{in\_as(l,n)}\}$. Furthermore, $meta(E^-)$ is the program $\{\texttt{violating :- v\_i.} \quad :\sim \text{not } \texttt{violating}.[1@0, \texttt{violating}]\} \cup \bigcup_{e \in E^-} meta(e) \cup \{\texttt{as(n)}\}$.

Example 6.9. Consider $E^- = \left\{ \begin{array}{l} \langle\{p,q\},\{r\}\rangle, \\ \langle\{q,r\},\{p\}\rangle \end{array} \right\}$. $meta(E^-)$ is the program:

```
v_i :- in_as(p, n), in_as(q, n), not in_as(r, n).
v_i :- in_as(q, n), in_as(r, n), not in_as(p, n).
as(n).
violating :- v_i.
:~ not violating.[1@0, violating]
```

Definition 6.10. Let $o = \langle e_1, e_2\rangle$ be in O^b.

$$meta(o) = dominates(o_{id1}, o_{id2}) \cup \{\texttt{as}(\texttt{o}_{\texttt{id1}}).\ \texttt{as}(\texttt{o}_{\texttt{id2}}).\ \texttt{:-}\text{not } \texttt{dom}(\texttt{o}_{\texttt{id1}}, \texttt{o}_{\texttt{id2}})\} \cup cover(reify(e_1, in\_as, o_{id1}), o_{id1})$$
$$\cup\, cover(reify(e_2, in\_as, o_{id2}), o_{id2})$$

Furthermore, $meta(O^b) = \bigcup_{o \in O^b} meta(o) \cup \{\texttt{lv(l)} \mid l \in L\}$.

Definition 6.11. Let $o = \langle e_1, e_2\rangle$ be in O^c.

$$meta(o) = dominates(e^1_{id}, e^2_{id}) \cup cover(reify(e^1, in\_as, e^1_{id}), e^1_{id}) \cup cover(reify(e^2, in\_as, e^2_{id}), e^2_{id})$$
$$\cup \{\texttt{vp}(\texttt{e}^1_{\texttt{id}}, \texttt{e}^2_{\texttt{id}})\ \texttt{:- not dom}(\texttt{e}^1_{\texttt{id}}, \texttt{e}^2_{\texttt{id}})\}$$

Furthermore, $meta(O^c) = \{\texttt{v\_p :- v\_p(T1,T2)}\} \cup \bigcup_{o \in O^c} meta(o) \cup \{\texttt{lv(l)} \mid l \in L\}$

Definition 6.12. $T_{meta} = meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(E^-) \cup meta(O^b) \cup meta(O^c)$.

Note that this definition is slightly different to the definition given in the paper. To simplify the proofs, we have partially ground the program. The task program in the paper is given in definition 6.13. The grounding of the programs are the same, and therefore have the same answer sets.

Definition 6.13. Let T be the ILP_{LOAS} task $\langle B, S_M, E^+, E^-, O^b, O^c\rangle$. Then $T_{meta} = meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(E^-) \cup meta(O^b) \cup meta(O^c)$ where each meta component is as follows:

- $meta(B) = append(reify(non\_weak(B), in\_as, X), as(X))$
  $\cup\, meta_{weak}(weak(B), in\_as, as, X)$.
- $meta(S_M) =$
  $\{append(append(reify(R, in\_as, X), as(X)), in\_h(R_{id})) \mid R \in non\_weak(S_M)\}$
  $\cup \{append(W, in\_h(W_{id})) \mid W \in meta_{weak}(weak(S_M), in\_as, as, X)\}$
  $\cup \{:\sim \texttt{in\_h}(\texttt{R}_{\texttt{id}}).[2 * |\texttt{R}|@0, \texttt{R}_{\texttt{id}}] \mid R \in S_M\}$
  $\cup \{\ \{\texttt{in\_h}(\texttt{R}_{\texttt{id}}) : \texttt{R} \in \texttt{S}_{\texttt{M}}\}.\ \}$
- $meta(E^+) = \left\{ \begin{array}{l} cover(e, e_{id}) \\ \texttt{as}(\texttt{e}_{\texttt{id}}). \end{array} \middle| \langle e^{inc}, e^{exc}\rangle \in E^+ \right\}$
- $meta(E^-) = \left\{ \begin{array}{l} \texttt{v\_i :- in\_as}(\texttt{e}^{\texttt{inc}}_1, \texttt{n}), \ldots, \texttt{in\_as}(\texttt{e}^{\texttt{inc}}_{\texttt{n}}, \texttt{n}), \\ \quad \texttt{not in\_as}(\texttt{e}^{\texttt{exc}}_1, \texttt{n}), \ldots, \\ \quad \texttt{not in\_as}(\texttt{e}^{\texttt{exc}}_{\texttt{m}}, \texttt{n}). \\ \texttt{as(n)}. \end{array} \middle| \langle e^{inc}, e^{exc}\rangle \in E^- \right\}$
  $\cup \left\{ \begin{array}{l} \texttt{violating :- v\_i.} \\ :\sim \texttt{not violating}.[1@0] \end{array} \right\}$

- $meta(O^b) = \left\{ \begin{array}{l|l} \texttt{as(o}_{\texttt{id1}}\texttt{).} \quad \texttt{as(o}_{\texttt{id2}}\texttt{).} & \\ cover(e^1, o_{id1}) & \\ cover(e^2, o_{id2}) & o = \langle e_1, e_2 \rangle \in O^b \\ dominates(o_{id1}, o_{id2}) & \\ \texttt{:- not dom(o}_{\texttt{id1}}\texttt{, o}_{\texttt{id2}}\texttt{).} & \end{array} \right\} \cup \{\texttt{lv(l).} \mid l \in L\}$
- $meta(O^c) = \left\{ \begin{array}{l|l} dominates(e_1, e_2) & \\ \texttt{v\_p(e}^1_{\texttt{id}}\texttt{, e}^2_{\texttt{id}}\texttt{) :-} & \langle e_1, e_2 \rangle \in O^c \\ \quad \texttt{not dom(e}_1\texttt{, e}_2\texttt{).} & \end{array} \right\} \cup \left\{ \begin{array}{l} \texttt{v\_p :- v\_p(T1, T2).} \\ \texttt{violating :- v\_p.} \end{array} \right\}$

Example 6.14. Let B be the program:

```
p(V) :- r(V), not q(V).
q(V) :- r(V), not p(V).

r(1).
r(2).

a :- not b.
b :- not a.
```

Let S_M be the set of rules:

```
q(1).
:~ q(V).[1@1, V, r2]
:~ b.[1@1, b, r3]
```

Let $E^+ = \left\{ \begin{array}{l} \langle \{p(2)\}, \emptyset \rangle, \\ \langle \emptyset, \{p(2)\} \rangle, \\ \langle \{a\}, \{b\} \rangle, \\ \langle \emptyset, \{a\} \rangle \end{array} \right.$

Let $E^- = \{ \ \langle \{p(1)\}, \emptyset \rangle$

Let $O^b = \{ \ \langle e^+_3, e^+_4 \rangle$

Let $O^c = \{ \ \langle e^+_1, e^+_2 \rangle$

Figure 1 shows T_{meta}.

6.2 Properties

Lemma 6.15. For any $H \subseteq S_M$,

$(meta(S_M) \cup meta(B))[H] = append(reify(non\_weak(B \cup H), in\_as, X), as(X)) \cup meta_{weak}(weak(B \cup H), in\_as, as, X)$

Proof. Follows directly from the definition of $[H]$. □

Proposition 6.16. For any $H \subseteq S_M$, partial interpretation e and term id:

$AS(append(reify(non\_weak(B \cup H), in\_as, id), as(id)) \cup meta(e))$
$\quad = \{reify(A, in\_as, id) \cup \{\texttt{as(id)}, \texttt{cov(id)}\} \mid A \in AS(B \cup H), A \text{ extends } e\}.$

Proof.

$AS(append(reify(non\_weak(B \cup H), in\_as, as(id))) \cup meta(e)).$

$\quad AS(append(reify(non\_weak(B \cup H), in\_as, as(id))) \cup \{\texttt{as(id)}\} \cup cover(reify(e, in\_as, id), id)).$

$\quad = \{A \mid A \in AS(append(reify(non\_weak(B \cup H), in\_as, id), as(id)) \cup \{\texttt{as(id)}\} \cup cover(reify(e, in\_as, id), id))\}$

```
% meta(B)
in_as(p(V),X) :- in_as(r(V),X),
  not in_as(q(V),X), as(X).
in_as(q(V),X) :- in_as(r(V),X),
  not in_as(p(V),X), as(X).

in_as(r(1),X) :- as(X).
in_as(r(2),X) :- as(X).

in_as(a,X) :- not in_as(b,X), as(X).
in_as(b,X) :- not in_as(a,X), as(X).


% meta(S_M)
in_as(q(1),X) :- as(X), in_h(r1).

w(1,1,args(V,r2),X) :- in_as(q(V),X),
  as(X), in_h(r2).

w(1,1,args(b,r3),X) :- in_as(b,X),
  as(X), in_h(r3).


% meta(E^+)
as(1).
as(2).
as(3).
as(4).

cov(1) :- in_as(p(2),1).
cov(2) :- not in_as(p(2),2).
cov(3) :- in_as(a,3), not in_as(b,3).
cov(4) :- not in_as(a,4).

:- not cov(1).
:- not cov(2).
:- not cov(3).
:- not cov(4).


% meta(E^-)
as(n).
v_i :- in_as(p(1),n).
```

```
% meta(O^b)
as(5).
as(6).

dom_lv(5,6,L) :- lv(L),
  #sum{w(W,L,A,5)=W, w(W,L,A,6)=-W} < 0.
wrong_dom_lv(5,6,L) :- lv(L),
  #sum{w(W,L,A,6)=W, w(W,L,A,5)=-W} < 0.
wrong_bef(5,6,L) :- lv(L), L < L2,
  wrong_dom_lv(5,6,L).
dom(5,6) :- dom_lv(5,6,L), not wrong_bef(5,6,L).

cov(5) :- in_as(a,5), not in_as(b,5).
cov(6) :- not in_as(a,6).

:- not cov(5).
:- not cov(6).
:- not dom(5,6).

lv(1).

% meta(O^c)

dom_lv(1,2,L) :- lv(L),
  #sum{w(W,L,A,1)=W, w(W,L,A,2)=-W} < 0.
wrong_dom_lv(1,2,L) :- lv(L),
  #sum{w(W,L,A,2)=W, w(W,L,A,1)=-W} < 0.
wrong_bef(1,2,L) :- lv(L), L < L2,
  wrong_dom_lv(1,2,L).
dom(1,2) :- dom_lv(1,2,L), not wrong_bef(1,2,L).

v_p(1,2) :- not dom(1,2).

violating :- v_p(X,Y).
v_p :- v_p(X,Y).
violating :- v_i.

0 {in_h(r1), in_h(r2), in_h(r3)} 2.

:~ in_h(r1).[2@0,r1]
:~ in_h(r2).[2@0,r2]
:~ in_h(r3).[2@0,r3]

:~ not violating.[1@0, violating]
```

Figure 1: An example of T_{meta}.

$= \left\{ A \cup \{\texttt{cov(id)}\} \;\middle|\; \begin{array}{c} A \in AS(append(reify(non\_weak(B \cup H), in\_as, id), as(id)) \cup \{\texttt{as(id)}\}), \\ A \text{ extends } reify(e, in\_as, id) \end{array} \right\}$ (by lemma 4.5).

$= \left\{ A \cup \{\texttt{as(id)}, \texttt{cov(id)}\} \;\middle|\; \begin{array}{c} A \in AS(reify(non\_weak(B \cup H), in\_as, id)), \\ A \text{ extends } reify(e, in\_as, id) \end{array} \right\}$ (by lemma 4.4).

$= \{reify(A, in\_as, id) \cup \{\texttt{as(id)}, \texttt{cov(id)}\} \mid A \in AS(B \cup H),\ A \text{ extends } e\}$

□

Corollary 6.17. For any $H \subseteq S_M$, positive example e:

$append(reify(non\_weak(B \cup H), in\_as, e_{id}), as(e_{id})) \cup meta(e)$ is satisfiable if and only if $B \cup H$ covers e.

Proposition 6.18. For any $H \subseteq S_M$:

$(meta(B) \cup meta(S_M) \cup meta(E^+))[H]$ is satisfiable if and only if $B \cup H$ covers all of the positive examples.

Proof.

Assume $B \cup H$ covers each of E^+

$\Leftrightarrow \forall e^+ \in E^+ append(reify(non\_weak(B \cup H), in\_as, e^+_{id}), as(e^+_{id})) \cup meta(e^+)$ is satisfiable (by corollary 6.17).

$\Leftrightarrow \bigcup_{e^+ \in E^+} (append(reify(non\_weak(B \cup H), in\_as, e^+_{id}), as(e^+_{id})) \cup meta(e^+))$ is satisfiable (by corollary 4.2).

$\Leftrightarrow append(reify(non\_weak(B \cup H), in\_as, X), as(X)) \cup \bigcup_{e^+ \in E^+} (meta(e^+))$ is satisfiable.

$\Leftrightarrow append(reify(non\_weak(B \cup H), in\_as, X), as(X)) \cup meta(E^+)$ is satisfiable.

$\Leftrightarrow append(reify(non\_weak(B \cup H), in\_as, X), as(X)) \cup meta_{weak}(weak(B \cup H), in\_as, as, X) \cup meta(E^+)$ is satisfiable by the splitting set theorem.

$\Leftrightarrow (meta(B) \cup meta(S_M) \cup meta(E^+))[H]$ is satisfiable. □

Proposition 6.19. For any $H \subseteq S_M$:

Let I be any interpretation.

$I \in AS(B \cup H)$ and $\exists e^- \in E^-$ st I extends e^- if and only if there is an answer set A of $(meta(B) \cup meta(S_M) \cup meta(E^-))[H]$ such that $\texttt{v\_i} \in A$ and $I = \mathcal{M}^{-1}_{vi}(A)$.

Proof. Let I be any interpretation.

Assume $I \in AS(B \cup H)$ and $\exists e^- \in E^-$ st I extends e^-

$\Leftrightarrow I \in AS(non\_weak(B \cup H))$ and $\exists e^- \in E^-$ st I extends e^- (as weak constraints do not affect answer sets).

$\Leftrightarrow \exists I \in AS(non\_weak(B \cup H))$ and $\exists e^- \in E^-$, st $reify(I, in\_as, n)$ extends $reify(e^-, in\_as, n)$

$\Leftrightarrow reify(I, in\_as, n) \in AS(reify(non\_weak(B \cup H), in\_as, n))$ and $\exists e^- \in E^-$ st $reify(I, in\_as, n)$ extends $reify(e^-, in\_as, n)$ (by lemma **??**)

$\Leftrightarrow \exists A \in AS(reify(non\_weak(B \cup H), in\_as, n)),\ \exists e^- \in E^-$ st $\mathcal{M}^{-1}_{vi}(A) = I$ and A extends $reify(e^-, in\_as, n)$

$\Leftrightarrow \exists A \in AS(reify(non\_weak(B \cup H), in\_as, n) \cup meta(e^-)),\ \exists e^- \in E^-$ st $\texttt{v\_i} \in A$ and $\mathcal{M}^{-1}_{vi}(A) = I$

$\Leftrightarrow \exists A \in AS(\bigcup_{e^- \in E^-} (meta(e^-)) \cup reify(non\_weak(B \cup H), in\_as, n))$ st $\texttt{v\_i} \in A$ and $\mathcal{M}^{-1}_{vi}(A) = I$

$\Leftrightarrow \exists A \in AS(\bigcup_{e^- \in E^-} (meta(e^-)) \cup \{\texttt{as(n)}\} \cup append(reify(non\_weak(B \cup H), in\_as, n), as(n)))$ st $\texttt{v\_i} \in A$ and $\mathcal{M}^{-1}_{vi}(A) = I$ (by lemma 4.4)

$\Leftrightarrow \exists A \in AS(append(reify(non\_weak(B \cup H), in\_as, n), as(n)) \cup meta(E^-))$ st $\texttt{v\_i} \in A$ and $\mathcal{M}^{-1}_{vi}(A) = I$

$\Leftrightarrow \exists A \in AS(append(reify(non\_weak(B \cup H), in\_as, X), as(X)) \cup meta(E^-))$ st $\texttt{v\_i} \in A$ and $\mathcal{M}^{-1}_{vi}(A) = I$ (as the ground program is the same)

$\Leftrightarrow \exists A \in AS(append(reify(non\_weak(B \cup H), in\_as, X), as(X)) \cup meta_{weak}(weak(B \cup H), in\_as, as, X) \cup meta(E^-))$ st $\texttt{v\_i} \in A$ and $\mathcal{M}^{-1}_{vi}(A) = I$ (by the splitting set theorem)

$\Leftrightarrow \exists A \in AS(\bigcup_{e^- \in E^-} ((meta(B) \cup meta(S_M) \cup meta(E^-))[H]))$ st $\texttt{v\_i} \in A$ and $\mathcal{M}_{vi}^{-1}(A) = I$

□

Proposition 6.20. For any $H \subseteq S_M$:
$AS((meta(B) \cup meta(S_M) \cup meta(O^b))[H])$ is satisfiable if and only if $B \cup H$ bravely respects every brave ordering.

Proof. For any $o = \langle o_1, o_2 \rangle \in O^b$, we define:

$P_1(o) = append(reify(non\_weak(B \cup H), in\_as, o_{id1}), as(o_{id1})) \cup cover(reify(o_1, in\_as, o_{id1}), o_{id1}) \cup \{\texttt{as(o}_{\texttt{id1}}\texttt{)}\}$

$P_2(o) = append(reify(non\_weak(B \cup H), in\_as, o_{id2}), as(o_{id2})) \cup cover(reify(o_2, in\_as, o_{id2}), o_{id2}) \cup \{\texttt{as(o}_{\texttt{id2}}\texttt{)}\}$

$$P_3(o) = \left\{ \begin{array}{l} dominates(o_{id1}, o_{id2}) \\ \cup\, meta_{weak}(weak(B \cup H), in\_as, as, o_{id1}) \\ \cup\, meta_{weak}(weak(B \cup H), in\_as, as, o_{id2}) \\ \cup\, \{\texttt{as(o}_{\texttt{id1}}\texttt{). as(o}_{\texttt{id2}}\texttt{). :- not dom(o}_{\texttt{id1}}\texttt{,o}_{\texttt{id2}}\texttt{).}\} \end{array} \middle| \; o \in O^b \right\} \cup \{\texttt{lv(l)} \mid l \in L\}$$

$P_4(A, id) = \{\texttt{in\_as(atom,id)} \mid \texttt{atom} \in A\}$

Assume $B \cup H$ bravely respects an ordering $o \in \langle o_1, o_2 \rangle$

Let a_1 and a_2 be answer sets of $B \cup H$ which bravely respect o.

$\Rightarrow a_1, a_2 \in AS(non\_weak(B \cup H))$ and a_1 extends o_1, a_2 extends o_2 and $a_1 \succ_{weak(B \cup H)} a_2$.

$\Rightarrow \forall i \in \{1, 2\}, \exists A \in AS(P_i(o))$ st $a_i = \{\texttt{atom} \mid \texttt{in\_as(atom,o}_{\texttt{id i}}\texttt{)} \in A\} \in AS(non\_weak(B \cup H))$ and a_i extends o_i and $a_1 \succ_{weak(B \cup H)} a_2$.

$\Rightarrow \exists A \in AS(P_1(o) \cup P_2(o))$ st $\forall i \in \{1, 2\}, a_i = \{\texttt{atom} \mid \texttt{in\_as(atom,o}_{\texttt{id i}}\texttt{)} \in A\} \in AS(non\_weak(B \cup H))$ and a_i extends o_i and $a_1 \succ_{weak(B \cup H)} a_2$.

$\Rightarrow \exists A \in AS(P_1(o) \cup P_2(o))$ st $\forall i \in \{1, 2\}, a_i = \{\texttt{atom} \mid \texttt{in\_as(atom,o}_{\texttt{id i}}\texttt{)} \in A\} \in AS(non\_weak(B \cup H))$ and $P_3(o) \cup P_4(a_1, o_{id1}) \cup P_4(a_2, o_{id2})$ is satisfiable by lemma **??**.

$\Rightarrow \exists A \in AS(P_1(o) \cup P_2(o))$ st $A \cup P_3(o)$ is satisfiable.

$\Rightarrow P_1(o) \cup P_2(o) \cup P_3(o)$ is satisfiable by the splitting set theorem.

Conversely, assume $\exists o \in O^b$ that is not bravely respected by $B \cup H$.

Case 1: o_1 is not extended by any Answer Set of $B \cup H$

$\Rightarrow P_1(o)$ is unsatisfiable.

$\Rightarrow (P_1(o) \cup P_2(o) \cup P_3(o))$ is unsatisfiable.

Case 2: o_2 is not extended by any Answer Set of $B \cup H$

$\Rightarrow P_2(o)$ is unsatisfiable.

$\Rightarrow (P_1(o) \cup P_2(o) \cup P_3(o))$ is unsatisfiable.

Case 3: For each pair of Answer Sets $\langle a_1, a_2 \rangle$ which extend o_1 and o_2, $a_1 \not\succ_{B \cup H} a_2$.

$\Rightarrow \forall A \in AS(P_1(o) \cup P_2(o)), \forall i \in \{1, 2\} a_i = \{\texttt{atom} \mid \texttt{in\_as(atom,o}_{\texttt{id i}}\texttt{)} \in A\} \in AS(non\_weak(B \cup H))$ and $a_1 \not\succ_{weak(B \cup H)} a_2$.

$\Rightarrow \forall A \in AS(P_1(o) \cup P_2(o)),\ a_i = \{\texttt{atom} \mid \texttt{in\_as(atom,o}_{\texttt{id i}}\texttt{)} \in A\} \in AS(non\_weak(B \cup H))$ and $P_3(o) \cup P_4(o, a_1, o_{id1}) \cup P_4(o, a_2, o_{id2})$ is unsatisfiable by lemma **??**.

$\Rightarrow (P_1(o) \cup P_2(o) \cup P_3(o))$ is unsatisfiable.

Hence $\forall o \in O^b$, $B \cup H$ bravely respects $o \Leftrightarrow \forall o \in O^b$, $(P_1(o) \cup P_2(o) \cup P_3(o))$ is satisfiable

$\Leftrightarrow (\bigcup_{o \in O^b} P_1(o) \cup P_2(o) \cup P_3(o)))$ is satisfiable

$\Leftrightarrow AS((meta(B) \cup meta(S_M) \cup meta(O^b))[H])$ is satisfiable (by corollary 4.3)

□

Corollary 6.21. $(meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H]$ is satisfiable if and only if H is a positive hypothesis of the task.

Proof. Assume $(meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H]$ is satisfiable.

$\Leftrightarrow$ $(meta(B) \cup meta(S_M) \cup meta(E^+))[H]$ is satisfiable and $(meta(B) \cup meta(S_M) \cup meta(O^b))[H]$ is satisfiable by corollary 4.3.

$\Leftrightarrow$ $B \cup H$ covers all the positive examples (by proposition 6.18) and $B \cup H$ bravely respects all brave ordering examples (proposition 6.20).

$\Leftrightarrow$ H is a positive hypothesis. □

Proposition 6.22. For any $H \subseteq S_M$ and any $o = \langle e^1, e^2 \rangle \in O^c$ (e^1 and e^2 are positive examples)

$$\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \\ \quad \cup\, meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \end{array}\middle| i \in \{1,2\}\right\} \cup \{lv(l) \mid l \in L\} \cup meta(o)\right) \text{ such that } vp(e^1_{id}, e^2_{id}) \in$$

S if and only if $\mathcal{M}^{-1}_{vp}(S, e^1_{id}, e^2_{id})$ is a violating pair of $B \cup H$ (which violates o).

Proof. Let $\langle A_1, A_2 \rangle$ be a violating pair of $B \cup H$ which violates o

$\Leftrightarrow$ $\forall i \in \{1,2\}, A_i \in AS(B \cup H)$ and A_i extends e^i and $A_1 \not\succ_{B \cup H} A_2$.

$\Leftrightarrow$ $\forall i \in \{1,2\}, \exists S_i \in AS(append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \cup meta(e^i))$ st $A_i = \{\texttt{atom} \mid \texttt{in\_as}(\texttt{atom}, \texttt{e}^\texttt{i}_\texttt{id}) \in S_i\}$ and $A_1 \not\succ_{B \cup H} A_2$ by proposition 6.16.

$\Leftrightarrow$ $\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \end{array}\middle| i \in \{1,2\}\right\}\right)$
st $\forall i \in \{1,2\}$, $A_i = \{\texttt{atom} \mid \texttt{in\_as}(\texttt{atom}, \texttt{e}^\texttt{i}_\texttt{id}) \in S\}$ and $A_1 \not\succ_{B \cup H} A_2$ (by corollary 4.2).

$\Leftrightarrow$ $\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \end{array}\middle| i \in \{1,2\}\right\}\right)$
st $\forall i \in \{1,2\}$, $A_i = \{\texttt{atom} \mid \texttt{in\_as}(\texttt{atom}, \texttt{e}^\texttt{i}_\texttt{id}) \in S\}$
and $\texttt{dom}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id}) \notin M\left(\left\{\begin{array}{l} meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \\ \quad \cup\, reify(A_i, in\_as, e^i_{id}) \cup \{\texttt{as}(\texttt{e}^\texttt{i}_\texttt{id})\} \end{array}\middle| i \in \{1,2\}\right\} \begin{array}{l} \cup dominates(e^1_{id}, e^2_{id}) \\ \cup \{\texttt{lv(l)} \mid l \in L\} \end{array}\right)$ (by lemma ??).

$\Leftrightarrow$ $\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \end{array}\middle| i \in \{1,2\}\right\}\right)$
st $\mathcal{M}^{-1}_{vp}(S, e^1_{id}, e^2_{id}) = \langle A_1, A_2 \rangle$
and $\texttt{dom}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id}) \notin M\left(\begin{array}{l} S \cup \{meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \mid i \in \{1,2\}\} \\ \quad \cup\, dominates(e^1_{id}, e^2_{id}) \cup \{\texttt{lv(l)} \mid l \in L\} \end{array}\right)$

$\Leftrightarrow$ $\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \\ \quad \cup\, meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \end{array}\middle| i \in \{1,2\}\right\} \cup dominates(e^1_{id}, e^2_{id}) \cup \{\texttt{lv(l)} \mid l \in L\}\right)$
st $\mathcal{M}^{-1}_{vp}(S, e^1_{id}, e^2_{id}) = \langle A_1, A_2 \rangle$
and $\texttt{dom}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id}) \notin S$ (by the splitting set theorem)

$\Leftrightarrow$ $\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \\ \quad \cup\, meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \end{array}\middle| i \in \{1,2\}\right\} \begin{array}{l} \cup dominates(e^1_{id}, e^2_{id}) \\ \cup \{\texttt{lv(l)} \mid l \in L\} \\ \cup \{\texttt{v\_p}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id})\texttt{ :- not dom}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id}).\} \end{array}\right)$
st $\mathcal{M}^{-1}_{vp}(S, e^1_{id}, e^2_{id}) = \langle A_1, A_2 \rangle$ and $\texttt{v\_p}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id}) \in S$

$\Leftrightarrow$ $\exists S \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \\ \quad \cup\, meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \end{array}\middle| i \in \{1,2\}\right\} \begin{array}{l} \cup \{\texttt{lv(l)} \mid l \in L\} \\ \cup meta(o) \end{array}\right)$
st $\mathcal{M}^{-1}_{vp}(S, e^1_{id}, e^2_{id}) = \langle A_1, A_2 \rangle$ and $\texttt{v\_p}(\texttt{e}^1_\texttt{id}, \texttt{e}^2_\texttt{id}) \in S$

□

Proposition 6.23. For any $H \subseteq S_M$:

Let $\langle I_1, I_2\rangle$ be any pair of interpretations.

H is a violating hypothesis with violating pair $\langle I_1, I_2\rangle$ if and only if $\exists A \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^c))[H])$ such that $v\_p \in A$ and $\mathcal{M}_{vp}^{-1}(A) = \langle I_1, I_2\rangle$

Proof.

$$AS\left(\left(\ meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b) \cup meta(O^c)\ \right)[H]\right) = \left\{ M\left(\begin{array}{c}(\bigcup_{i=1}^{n} A_i)\ \cup \\ \{\texttt{v\_p :- v\_p(V1, V2)}\}\end{array}\right) \middle| \begin{array}{c} A_1 \in AS((meta(o^1) \cup meta(O^b) \cup meta(B) \cup meta(S_M) \cup meta(E^+))[H]), \\ \ldots, \\ A_n \in AS((meta(o^n) \cup meta(O^b) \cup meta(B) \cup meta(S_M) \cup meta(E^+))[H]) \end{array}\right\}$$

(By repeated applications of lemma **??**)

Hence:

$\exists A \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b) \cup meta(O^c))[H])$ st $v\_p \in A$ and $\mathcal{M}_{vp}^{-1}(A) = \langle I_1, I_2\rangle$

$\Leftrightarrow \forall o \in O^c, (meta(o) \cup meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H]$ is satisfiable and $\exists o = \langle e^1, e^2\rangle \in O^c$ st $\exists A \in AS((meta(o) \cup meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H])$ st $vp(e^1_{id}, e^2_{id}) \in A$ and $\mathcal{M}_{vp}^{-1}(A) = \langle I_1, I_2\rangle$.

$\Leftrightarrow (meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H]$ is satisfiable (as $meta(o)$ is stratified and contains no atom which appears in the bodies of the rest of the program) and $\exists o = \langle e^1, e^2\rangle \in O^c$ st $\exists A \in AS((meta(o) \cup meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H])$ st $v\_p(e^1_{id}, e^2_{id}) \in A$ and $\mathcal{M}_{vp}^{-1}(A, e^1_{id}, e^2_{id}) = \langle I_1, I_2\rangle$.

$\Leftrightarrow H$ is a positive hypothesis (by corollary 6.21) and $\exists o = \langle e^1, e^2\rangle \in O^c$ st $\exists A \in AS((meta(o) \cup meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H])$ st $vp(e^1_{id}, e^2_{id}) \in A$ and $\mathcal{M}_{vp}^{-1}(A, e^1_{id}, e^2_{id}) = \langle I_1, I_2\rangle$

$\Leftrightarrow H$ is a positive hypothesis and $\exists o = \langle e^1, e^2\rangle \in O^c$ st $\exists A \in AS((meta(o) \cup meta(B) \cup meta(S_M) \cup meta(E^+) \cup \{\texttt{lv(l)} \mid l \in L\}$ st $v\_p(e^1_{id}, e^2_{id}) \in A$ and $\mathcal{M}_{vp}^{-1}(A, e^1_{id}, e^2_{id}) = \langle I_1, I_2\rangle$ (as the rest of O^b has no heads used in the body of the remaining rules).

$\Leftrightarrow H$ is a positive hypothesis and $\exists o = \langle e^1, e^2\rangle \in O^c$ st $\exists A \in AS((meta(o) \cup meta(B) \cup meta(S_M) \cup meta(e^1_{id}) \cup meta(e^2_{id}) \cup \{\texttt{lv(l)} \mid l \in L\}$ st $v\_p(e^1_{id}, e^2_{id}) \in A$ and $\mathcal{M}_{vp}^{-1}(A, e^1_{id}, e^2_{id}) = \langle I_1, I_2\rangle$ (as the rest of $meta(E^+)$ has no heads used in the body of the remaining rules).

$\Leftrightarrow H$ is a positive hypothesis and $\exists o = \langle e^1, e^2\rangle \in O^c$

$$\text{st } \exists A \in AS\left(\left\{\begin{array}{l} append(reify(non\_weak(B \cup H), in\_as, e^i_{id})) \\ \quad \cup\, meta(e^i) \\ \quad \cup\, meta_{weak}(weak(B \cup H), in\_as, as, e^i_{id}) \end{array} \middle| i \in \{1, 2\}\right\} \cup \{\texttt{lv(l)} \mid l \in L\} \cup meta(o)\right)$$

st $vp(e^1_{id}, e^2_{id}) \in A$ and $\mathcal{M}_{vp}^{-1}(A, e^1_{id}, e^2_{id}) = \langle I_1, I_2\rangle$ (as the relevant grounding of the two programs is the same).

$\Leftrightarrow H$ is a positive hypothesis and $\exists o \in O^c$ such that o $\langle I_1, I_2\rangle$ is a violating pair of H (by proposition 6.22).

$\Leftrightarrow H$ is a violating hypothesis with violating pair $\langle I_1, I_2\rangle$.

□

Theorem 6.24. *For any $H \subseteq S_M$:*

1. *$T_{meta}[H]$ is satisfiable if and only if H is a positive hypothesis of T.*
2. *$\exists A \in AS(T_{meta})[H])$ such that A contains the atom v_i if and only if H is a violating hypothesis such that $B \cup H$ has the violating interpretation $\mathcal{M}_{vi}^{-1}(A)$.*
3. *$\exists A \in AS(T_{meta})[H])$ such that A contains the atom v_p if and only if H is a violating hypothesis with violating pair $\mathcal{M}_{vp}^{-1}(A)$*

Proof. 1. $T_{meta} = meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(E^-) \cup meta(O^b) \cup meta(O^c)$

Assume T_{meta} is satisfiable.

$\Leftrightarrow (meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H]$ is satisfiable, as no head literal in $meta(O^c) \cup meta(E^-)$ unifies with any body literal in the rest of the program and $meta(O^c) \cup meta(E^-)$ is stratified.

$\Leftrightarrow H$ is a positive hypothesis by corollary 6.21.

2. Let A be an answer set of $T_{meta}[H]$ which contains $\texttt{v\_i}$ and let $V = \mathcal{M}_{vi}^{-1}(A)$.

   $\Leftrightarrow S \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b) \cup meta(O^c) \cup \{\texttt{as(n)}\})[H])$ and $\exists S_2 \in AS(S \cup meta(E^-)\backslash\{\texttt{as(n)}\})[H]$ st $\texttt{v\_i} \in S_2$ and $V = \mathcal{M}_{vi}^{-1}(S_2)$ by the splitting set theorem.

   $\Leftrightarrow \exists S \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b) \cup \{\texttt{as(n)}\})[H])$ st $\exists S_2 \in AS(S \cup meta(E^-)\backslash\{\texttt{as(n)}\})[H])$ st $\texttt{v\_i} \in S_2$ and $V = \mathcal{M}_{vi}^{-1}(S_2)$ (as O^c is stratified and contains no atom in E^-)

   $\Leftrightarrow \exists S_1 \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b)), \exists S_2 \in AS((meta(B) \cup meta(S_M) \cup \{\texttt{as(n)}\})[H])$ st $\exists S_3 \in AS(S_1 \cup S_2 \cup (meta(E^-)\backslash\{\texttt{as(n)}\}))[H]$ st $i\texttt{v\_i} \in S_3$ and $V = \mathcal{M}_{vi}^{-1}(S_3)$ (by corollary 4.3).

   $\Leftrightarrow \exists S_1 \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b))[H]), \exists S_2 \in AS((meta(B) \cup meta(S_M) \cup \{\texttt{as(n)}\})[H])$ st $\exists S_3 \in AS(S_2 \cup (meta(E^-)\backslash\{\texttt{as(n)}\}))[H]$ st $\texttt{v\_i} \in S_3$ and $V = \mathcal{M}_{vi}^{-1}(S_3)$ (as no atom in S_1 occurs in $meta(E^-)$).

   $\Leftrightarrow H$ is a positive hypothesis and $\exists S_1 \in AS((meta(B) \cup meta(S_M) \cup \{\texttt{as(n)}\})[H])$ st $\exists S_2 \in AS(S_1 \cup (meta(E^-)\backslash\{\texttt{as(n)}\})[H])$ st $\texttt{v\_i} \in S_2$ and $V = \mathcal{M}_{vi}^{-1}(S_2)$

   $\Leftrightarrow H$ is a positive hypothesis and $\exists S_1 \in AS((meta(B) \cup meta(S_M) \cup \{\texttt{as(n)}\})[H])$ st $\exists S_2 \in AS(S_1 \cup (meta(E^-))[H])$ st $\texttt{v\_i} \in S_2$ and $V = \mathcal{M}_{vi}^{-1}(S_2)$

   $\Leftrightarrow H$ is a positive hypothesis and $\exists S \in AS((meta(B) \cup meta(S_M) \cup meta(E^-))[H])$ st $\texttt{v\_i} \in S$ and $V = \mathcal{M}_{vi}^{-1}(S)$

   $\Leftrightarrow H$ is a positive hypothesis with violating interpretation V (by proposition 6.19).

   $\Leftrightarrow H$ is a violating hypothesis with violating interpretation V.

3. Let A be an answer set of $T_{meta}[H]$ which contains $\texttt{v\_p}$ and let $P = \mathcal{M}_{vp}^{-1}(A)$

   $\Leftrightarrow \exists S \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b) \cup meta(O^c))[H])$ st $\texttt{v\_p} \in S$ and $P = \mathcal{M}_{vp}^{-1}(S)$ and $(meta(B) \cup meta(S_M) \cup meta(E^-))[H]$ is satisfiable (by corollary 4.3)

   Notice that if $(meta(B) \cup meta(S_M) \cup meta(E^-))[H]$ were unsatisfiable then T_{meta} would be too (by the splitting set theorem). Hence:

   $\Leftrightarrow \exists S \in AS((meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(O^b) \cup meta(O^c))[H])$ st $\texttt{v\_p} \in S$ and $P = \mathcal{M}_{vp}^{-1}(S)$

   $\Leftrightarrow H$ is a violating hypothesis with violating pair V (by proposition 6.23).

□

7 Ruling out classes of violating hypothesis: VR_{meta}

In this section, as before we will assume an ILP_{LOAS} task $T = \langle B, S_M, E^+, E^-, O^b, O^c\rangle$. We will also assume a set of violating reasons $VR = VI \cup VP$, where VI are violating interpretations and VP are violating pairs.

7.1 Meta Level representation

Definition 7.1. Given any choice rule $R = \texttt{l}\{\texttt{h}_1, \ldots, \texttt{h}_n\}\texttt{u} \texttt{:-} \texttt{body}$, $reductify(R)$ is the program:

$$\left\{\begin{array}{c} \texttt{mmr}(\texttt{h}_1,\texttt{X}) \texttt{:-} \texttt{reify}(\texttt{body}^+,\texttt{mmr},\texttt{X}), \texttt{reify}(\texttt{body}^-,\ \texttt{not in\_vs},\texttt{X}),\\ \texttt{l}\{\texttt{in\_vs}(\texttt{h}_1,\texttt{X}),\ldots,\texttt{in\_vs}(\texttt{h}_n,\texttt{X})\}\texttt{u}, \texttt{in\_vs}(\texttt{h}_1,\texttt{X}).\\ \ldots\\ \texttt{mmr}(\texttt{h}_n,\texttt{X}) \texttt{:-} \texttt{reify}(\texttt{body}^+,\texttt{mmr},\texttt{X}), \texttt{reify}(\texttt{body}^-,\ \texttt{not in\_vs},\texttt{X}),\\ \texttt{l}\{\texttt{in\_vs}(\texttt{h}_1,\texttt{X}),\ldots,\texttt{in\_vs}(\texttt{h}_n,\texttt{X})\}\texttt{u}, \texttt{in\_vs}(\texttt{h}_n,\texttt{X}).\\ \texttt{mmr}(\bot,\texttt{X}) \texttt{:-} \texttt{reify}(\texttt{body}^+,\texttt{mmr},\texttt{X}), \texttt{reify}(\texttt{body}^-,\ \texttt{not in\_vs},\texttt{X}),\\ \texttt{u}+\texttt{1}\{\texttt{in\_vs}(\texttt{h}_1,\texttt{X}),\ldots,\texttt{in\_vs}(\texttt{h}_n,\texttt{X})\}.\\ \texttt{mmr}(\bot,\texttt{X}) \texttt{:-} \texttt{reify}(\texttt{body}^+,\texttt{mmr},\texttt{X}), \texttt{reify}(\texttt{body}^-,\ \texttt{not in\_vs},\texttt{X}),\\ \{\texttt{in\_vs}(\texttt{h}_1,\texttt{X}),\ldots,\texttt{in\_vs}(\texttt{h}_n,\texttt{X})\}\texttt{l}-\texttt{1}. \end{array}\right\}$$

Definition 7.2. Let P be an ASP program and $\texttt{t}$ be a term. $reductify(P, t)$ is the program:

$$\{\texttt{mmr}(\texttt{head},\texttt{t}) \texttt{:-} \texttt{reify}(\texttt{body}^+(\texttt{R}),\texttt{mmr},\texttt{t}), \texttt{reify}(\texttt{body}^-(\texttt{R}),\ \texttt{not in\_vs},\texttt{t}), \texttt{vs}(\texttt{t}).\ |\ R \in normal(P)\}$$

$$\cup\ \{\texttt{mmr}(\bot,\texttt{t}) \texttt{:-} \texttt{reify}(\texttt{body}^+(\texttt{R}),\texttt{mmr},\texttt{t}), \texttt{reify}(\texttt{body}^-(\texttt{R}),\ \texttt{not in\_vs},\texttt{t}), \texttt{vs}(\texttt{t}).\ |\ R \in constraint(P)\}$$

$$\cup\ \{reductify(R, t)\ |\ R \in choice(P)\}.$$

For any term $\texttt{t}$ we write:

$R_1(t) = \texttt{nas}(\texttt{t}) \texttt{:-} \texttt{in\_vs}(\texttt{ATOM},\texttt{t}),\ \texttt{not mmr}(\texttt{ATOM},\texttt{t})$

$R_2(t) = \texttt{nas}(\texttt{t}) \texttt{:-}\ \texttt{not in\_vs}(\texttt{ATOM},\texttt{t}), \texttt{mmr}(\texttt{ATOM},\texttt{t})$

We will also use the shorthand $nas\_rules(t)$ to denote $R_1(t) \cup R_2(t)$.

Definition 7.3. $VI_{meta}(T)$ is the program:

$$\left\{\begin{array}{l} reductify(B, X) \cup nas\_rules(X) \cup\\ \{append(reductify(R, X), in\_hyp(R_{id}))\ |\ R \in S_M\} \end{array}\right\} \cup \left\{\begin{array}{l} \{\texttt{:- not nas}(\texttt{I}_{\texttt{id}}).\}\\ \cup reify(I, in\_vs, I_{id})\\ \cup\{\texttt{vs}(\texttt{I}_{\texttt{id}}).\} \end{array}\ \middle|\ I \in VI\right\}$$

Definition 7.4. For any violating pair $vp = \langle I_1, I_2\rangle$, $meta(vp)$ is the program:

$$dominates(vp_{id1}, vp_{id2}) \cup \{\texttt{lv}(\texttt{l}).\ |\ l \in L\} \cup \left\{\begin{array}{l} reify(I_1, in\_vs, vp_{id1}) \cup\\ nas\_rules(vp_{id1}) \cup\\ \{\texttt{vs}(\texttt{vp}_{\texttt{id i}}).\} \end{array}\ \middle|\ i \in \{1, 2\}\right\} \cup \left\{\begin{array}{l} \texttt{:- not nas}(\texttt{vp}_{\texttt{id1}}),\\ \quad \texttt{not nas}(\texttt{vp}_{\texttt{id2}}),\\ \quad \texttt{not dom}(\texttt{vp}_{\texttt{id1}}, \texttt{vp}_{\texttt{id2}}). \end{array}\right\}$$

Definition 7.5. $VP_{meta}(T))$ is the program:

$$reductify(B, X) \cup \left\{\begin{array}{ll} append(reductify(R, X), in\_hyp(R_{id})) & |\ R \in S_M\\ append(meta_{weak}(W, in\_vs, vs, X), in\_hyp(W_{id})) & |\ W \in weak(S_M)\\ meta_{weak}(W, in\_vs, vs, X) & |\ W \in weak(B) \end{array}\right\} \cup \{meta(vp)\ |\ vp \in VP\}$$

Definition 7.6. $VR_{meta}(T) = VI_{meta}(T) \cup VP_{meta}(T)$

Similar to the previous section, we again partially ground the program to simplify the proofs. The representation given in the paper is given in definition 7.7.

Definition 7.7. Let T be the ILP_{LOAS} task $\langle B, S_M, E^+, E^-, O^b, O^c\rangle$ and VR be the set of violating reasons $VI \cup VP$, where VI are violating interpretations and VP are violating pairs.

$VR_{meta}(T)$ is the program $meta(VI) \cup meta(VP) \cup meta(Aux)$ where the meta components are defined as follows:

- $meta(VI) = \left\{ \begin{array}{l} reify(I, in\_vs, I_{id}) \\ \texttt{:- not nas(I}_{\texttt{id}}\texttt{).} \\ \texttt{vs(I}_{\texttt{id}}\texttt{).} \end{array} \middle| I \in VI \right\}$

- $meta(VP) = \left\{ \begin{array}{l} dominates(vp_{id1}, vp_{id2}) \\ reify(I_1, in\_vs, vp_{id1}) \\ reify(I_2, in\_vs, vp_{id2}) \\ \texttt{vs(vp}_{\texttt{id1}}\texttt{).} \\ \texttt{vs(vp}_{\texttt{id2}}\texttt{).} \\ \texttt{:- not nas(vp}_{\texttt{id1}}\texttt{), not nas(vp}_{\texttt{id2}}\texttt{),} \\ \quad \texttt{not dom(vp}_{\texttt{id1}}\texttt{, vp}_{\texttt{id2}}\texttt{).} \end{array} \middle| vp = \langle I_1, I_2 \rangle \in VP \right\}$

- $meta(Aux) =$

$$\left\{ \begin{array}{l} reductify(B) \\ \quad \cup \left\{ \begin{array}{l} \texttt{nas(X) :- in\_vs(ATOM, X), not mmr(ATOM, X).} \\ \texttt{nas(X) :- not in\_vs(ATOM, X), mmr(ATOM, X).} \end{array} \right\} \\ \quad \cup \{append(reductify(R), in\_hyp(R_{id})) \mid R \in non\_weak(S_M)\} \\ \quad \cup \{append(meta_{weak}(W, in\_vs, vs, X), in\_hyp(W_{id})) \mid W \in weak(S_M)\} \\ \quad \cup \{meta_{weak}(W, in\_vs, vs, X) \mid W \in weak(B)\} \\ \quad \cup \{\texttt{lv(l).} \mid l \in L\} \end{array} \right\}$$

Example 7.8. Let B be the program:

```
p(V) :- r(V), not q(V).
q(V) :- r(V), not p(V).

r(1).
r(2).

a :- not b.
b :- not a.
```

Let S_M be the set of rules:

```
q(1).
:~ q(V).[1@1, V, r2]
:~ b.[1@1, b, r3]
```

Let $E^+ = \left\{ \begin{array}{l} \langle \{p(2)\}, \emptyset \rangle, \\ \langle \emptyset, \{p(2)\} \rangle, \\ \langle \{a\}, \{b\} \rangle, \\ \langle \emptyset, \{a\} \rangle \end{array} \right.$

Let $E^- = \{\ \langle \{p(1)\}, \emptyset \rangle$

Let $O^b = \{\ \langle e_3^+, e_4^+ \rangle$

Let $O^c = \{\ \langle e_1^+, e_2^+ \rangle$

Let $VI = \{\ \{p(1), p(2), r(1), r(2), a\}$

Let $VP = \{\ \langle \{p(2), q(1), r(1), r(2), a\}, \{q(1), q(2), r(1), r(2), a\} \rangle$

Let the set of violating reasons VR be $VI \cup VP$. Then figure 2 shows $VR_{meta}(T)$.

7.2 Properties

Lemma 7.9. Given any program P (with no weak constraints), term $\texttt{t}$ and interpretation I:

Let Q be the program $reductify(P, t) \cup reify(I, in\_vs, t) \cup \{\texttt{vs(t)}\}$.

Q is globally stratified and thus has a unique Answer Set denoted $M(Q)$.

```
% reductify(B,X)
mmr(p(V),X) :- mmr(r(V),X),
  not in_vs(q(V),X), vs(X).
mmr(q(V),X) :- mmr(r(V),X),
  not in_vs(p(V),X), vs(X).
mmr(r(1),X) :- vs(X).
mmr(r(2),X) :- vs(X).
mmr(a,X) :- not in_vs(b,X), vs(X).
mmr(b,X) :- not in_vs(a,X), vs(X).

% reductify(S_M) + in_hyp
mmr(q(1),X) :- vs(X), in_h(r1).

w(1,1,ts(V),X) :- vs(X),
  in_vs(q(V),X), in_h(r2).

w(1,1,args(b,r3),X) :- vs(X),
  in_vs(b,X), in_h(r3).

% nas_rules(X)
nas(X) :- in_vs(A,X), not mmr(A,X).
nas(X) :- not in_vs(A,X), mmr(A,X).


% VI
in_vs(p(1),v1).
in_vs(p(2),v1).
in_vs(r(1),v1).
in_vs(r(2),v1).
in_vs(a,v1).
vs(v1).
:- not nas(v1).
```

```
% VP
in_vs(p(2),v2).
in_vs(q(1),v2).
in_vs(r(1),v2).
in_vs(r(2),v2).
in_vs(a,v2).
vs(v2).

in_vs(q(1),v3).
in_vs(q(2),v3).
in_vs(r(1),v3).
in_vs(r(2),v3).
in_vs(a,v3).
vs(v3).

dom_lv(v2,v3,L) :- lv(L),
   #sum{w(W,L,A,v2)=W, w(W,L,A,v3)=-W} < 0.
wrong_dom_lv(v2,v3,L) :- lv(L),
   #sum{w(W,L,A,v3)=W, w(W,L,A,v2)=-W} < 0.
wrong_bef(v2,v3,L) :- lv(L), L < L2,
   wrong_dom_lv(1,2,L).
dom(v2,v3) :- dom_lv(v2,v3,L),
   not wrong_bef(v2,v3,L).

:- not nas(v2), not nas(v3),
   not dom(v2,v3).
```

Figure 2: An example of $VR_{meta}(T)$.

Lemma 7.10. Given any program P, term $\mathtt{t}$ and interpretation I:

Let $Q = reductify(P, t) \cup reify(I, in\_vs, t) \cup \{\mathtt{vs(t)}\}$

$M(P^I) = \{\mathtt{atom} \mid \mathtt{mmr(atom, t)} \in M(Q)\}$

Proof.

$M(Q)$

$= M(\{\mathtt{vs(t)}\} \cup reify(I, in\_vs, t) \cup reductify(P, t))$

$= M(\{\mathtt{vs(t)}\} \cup reify(I, in\_vs, t) \cup reductify(ground(P), t))$

$$= M\left(\begin{array}{l}\{\mathtt{vs(t)}\}\cup\\ reify(I, in\_vs, t)\cup\end{array}\left\{\begin{array}{l|l}\begin{array}{l}\mathtt{mmr(head,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t).}\end{array} & R \in normal(ground(P))\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t).}\end{array} & R \in constraint(ground(P))\\ reductify(R) & R \in choice(ground(P))\end{array}\right\}\right)$$

$$= M\left(\begin{array}{l}\{\mathtt{vs(t)}\}\cup\\ reify(I, in\_vs, t)\cup\end{array}\left\{\begin{array}{l|l}\begin{array}{l}\mathtt{mmr(head,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t).}\end{array} & R \in normal(ground(P))\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t).}\end{array} & R \in constraint(ground(P))\}\\ \begin{array}{l}\mathtt{mmr(h,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t),}\\ \quad\mathtt{l\{in\_vs(h_1,t),\ldots,in\_vs(h_n)\}u,\ in\_vs(h)}\end{array} & \begin{array}{l}R \in choice(ground(P)),\\ head(R) = \mathtt{l\{h_1,\ldots,h_n\}u},\\ h \in \{h_1,\ldots h_n\}\end{array}\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t),}\\ \quad\mathtt{\{in\_vs(h_1,t),\ldots,in\_vs(h_n)\}l-1}\end{array} & R \in choice(ground(P))\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t),}\\ \quad\mathtt{u+1\{in\_vs(h_1,t),\ldots,in\_vs(h_n)\}}\end{array} & R \in choice(ground(P))\end{array}\right\}\right)$$

$$= M\left(\begin{array}{l}\{\mathtt{vs(t)}\}\cup\\ reify(I, in\_vs, t)\cup\end{array}\left\{\begin{array}{l|l}\begin{array}{l}\mathtt{mmr(head,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t).}\end{array} & R \in normal(ground(P))\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t).}\end{array} & R \in constraint(ground(P))\}\\ \begin{array}{l}\mathtt{mmr(h,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t),}\\ \quad\mathtt{l\{in\_vs(h_1,t),\ldots,in\_vs(h_n)\}u.}\end{array} & \begin{array}{l}R \in choice(ground(P)),\\ head(R) = \mathtt{l\{h_1,\ldots,h_n\}u},\\ h \in \{h_1,\ldots h_n\}\cap I\end{array}\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t),}\\ \quad\mathtt{\{in\_vs(h_1,t),\ldots,in\_vs(h_n)\}l-1}\end{array} & R \in choice(ground(P))\\ \begin{array}{l}\mathtt{mmr(\bot,t)\text{ :- }reify(body^+(R),mmr,t),}\\ \quad\mathtt{reify(body^-(R),\ not\ in\_vs,t),vs(t),}\\ \quad\mathtt{u+1\{in\_vs(h_1,t),\ldots,in\_vs(h_n)\}}\end{array} & R \in choice(ground(P))\end{array}\right\}\right)$$

(because $\mathtt{in\_vs(atom,t)}$ is true if and only if $atom \in I$ as these atoms come from $reify(I, in\_vs, t)$).

$$
= M\left(\begin{array}{l}\{\texttt{vs(t)}\}\cup\\ reify(I, in\_vs, t)\cup\end{array}\left\{\begin{array}{l|l}
\begin{array}{l}\texttt{mmr(head,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t).}\end{array} & R \in normal(ground(P))\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t).}\end{array} & R \in constraint(ground(P))\}\\
\begin{array}{l}\texttt{mmr(h,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{l\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}u.}\end{array} & \begin{array}{l}R \in choice(ground(P)),\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ h \in \{h_1,\ldots h_n\}\cap I\\ l \leq |\{h_1,\ldots h_n\}\cap I| \leq u\end{array}\\
\begin{array}{l}\texttt{mmr(h,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{l\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}u.}\end{array} & \begin{array}{l}R \in choice(ground(P)),\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ h \in \{h_1,\ldots h_n\}\cap I\\ l > |\{h_1,\ldots h_n\}\cap I|\end{array}\\
\begin{array}{l}\texttt{mmr(h,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{l\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}u.}\end{array} & \begin{array}{l}R \in choice(ground(P)),\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ h \in \{h_1,\ldots h_n\}\cap I\\ |\{h_1,\ldots h_n\}\cap I| > u\end{array}\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}l}-\texttt{1}\end{array} & \begin{array}{l}R \in choice(ground(P))\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ l \leq |\{h_1,\ldots h_n\}\cap I| \leq u\end{array}\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}l}-\texttt{1}\end{array} & \begin{array}{l}R \in choice(ground(P))\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ l > |\{h_1,\ldots h_n\}\cap I|\end{array}\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}l}-\texttt{1}\end{array} & \begin{array}{l}R \in choice(ground(P))\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ |\{h_1,\ldots h_n\}\cap I| > u\end{array}\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{u}+\texttt{1\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}}\end{array} & \begin{array}{l}R \in choice(ground(P))\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ |l \leq \{h_1,\ldots h_n\}\cap I| \leq u\end{array}\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{u}+\texttt{1\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}}\end{array} & \begin{array}{l}R \in choice(ground(P))\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ |l > \{h_1,\ldots h_n\}\cap I|\end{array}\\
\begin{array}{l}\texttt{mmr(}\bot\texttt{,t):-reify(body}^{+}\texttt{(R),mmr,t),}\\ \quad\texttt{reify(body}^{-}\texttt{(R), not in\_vs,t),vs(t),}\\ \quad\texttt{u}+\texttt{1\{in\_vs(h}_1\texttt{,t),}\ldots\texttt{,in\_vs(h}_n\texttt{)\}}\end{array} & \begin{array}{l}R \in choice(ground(P))\\ head(R) = \texttt{l\{h}_1,\ldots,\texttt{h}_n\texttt{\}u},\\ |\{h_1,\ldots h_n\}\cap I| > u\end{array}
\end{array}\right\}\right)
$$

(the programs are the same as the new conditions are exhaustive)

$$= M\left(\begin{array}{l}\{\texttt{vs(t)}\}\cup\\ reify(I, in_vs, t)\cup\end{array}\left\{\begin{array}{l|l}
\texttt{mmr(head,t) :- reify(body}^+\texttt{(R),mmr,t),} & \\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & R\in normal(ground(P))\\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^+\texttt{(R),mmr,t),} & \\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & R\in constraint(ground(P))\}\\
 & R\in choice(ground(P)),\\
\texttt{mmr(h,t) :- reify(body}^+\texttt{(R),mmr,t),} & head(R)=\texttt{l}\{\texttt{h}_1,\ldots,\texttt{h}_n\}\texttt{u},\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t),} & h\in\{h_1,\ldots h_n\}\cap I\\
\quad\texttt{l\{in_vs(h}_1\texttt{,t),}\ldots\texttt{,in_vs(h}_n\texttt{)\}u.} & l\leq|\{h_1,\ldots h_n\}\cap I|\leq u\\
 & R\in choice(ground(P))\\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^+\texttt{(R),mmr,t),} & head(R)=\texttt{l}\{\texttt{h}_1,\ldots,\texttt{h}_n\}\texttt{u},\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t),} & l>|\{h_1,\ldots h_n\}\cap I|\\
\quad\texttt{\{in_vs(h}_1\texttt{,t),}\ldots\texttt{,in_vs(h}_n\texttt{)\}l}-\texttt{1} & \\
 & R\in choice(ground(P))\\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^+\texttt{(R),mmr,t),} & head(R)=\texttt{l}\{\texttt{h}_1,\ldots,\texttt{h}_n\}\texttt{u},\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t),} & |\{h_1,\ldots h_n\}\cap I|>u\\
\quad\texttt{u}+\texttt{1\{in_vs(h}_1\texttt{,t),}\ldots\texttt{,in_vs(h}_n\texttt{)\}} &
\end{array}\right\}\right)$$

(by corollary 4.7 (part 2), as we have only removed rules whose bodies were false due to the sum being false).

$$= M\left(\begin{array}{l}\{\texttt{vs(t)}\}\cup\\ reify(I, in_vs, t)\cup\end{array}\left\{\begin{array}{l|l}
 & R\in normal(ground(P))\\
\texttt{mmr(head,t) :- reify(body}^+\texttt{(R),mmr,t),} & reify(body^-(R), in_vs, t)\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & \quad\cap\, reify(I, in_vs, t)=\emptyset\\
 & R\in constraint(ground(P))\\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^+\texttt{(R),mmr,t),} & reify(body^-(R), in_vs, t)\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & \quad\cap\, reify(I, in_vs, t)=\emptyset\\
 & R\in choice(ground(P)),\\
 & reify(body^-(R), in_vs, t)\\
\texttt{mmr(h,t) :- reify(body}^+\texttt{(R),mmr,t),} & \quad\cap\, reify(I, in_vs, t)=\emptyset\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & head(R)=\texttt{l}\{\texttt{h}_1,\ldots,\texttt{h}_n\}\texttt{u},\\
 & h\in\{h_1,\ldots h_n\}\cap I\\
 & l\leq|\{h_1,\ldots h_n\}\cap I|\leq u\\
 & R\in choice(ground(P))\\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^+\texttt{(R),mmr,t),} & reify(body^-(R), in_vs, t)\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & \quad\cap\, reify(I, in_vs, t)=\emptyset\\
 & head(R)=\texttt{l}\{\texttt{h}_1,\ldots,\texttt{h}_n\}\texttt{u},\\
 & l>|\{h_1,\ldots h_n\}\cap I|\\
 & R\in choice(ground(P))\\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^+\texttt{(R),mmr,t),} & reify(body^-(R), in_vs, t)\\
\quad\texttt{reify(body}^-\texttt{(R), not in_vs,t),vs(t).} & \quad\cap\, reify(I, in_vs, t)=\emptyset\\
 & head(R)=\texttt{l}\{\texttt{h}_1,\ldots,\texttt{h}_n\}\texttt{u},\\
 & |\{h_1,\ldots h_n\}\cap I|>u
\end{array}\right\}\right)$$

(As the rules which have been removed contain the negation of at least one fact in the program).

$$
= M\left(\begin{array}{l}\{\mathtt{vs(t)}\}\cup\\ reify(I, in\_vs, t)\cup\end{array}\left\{\begin{array}{l|l}
\begin{array}{l}\mathtt{mmr(head,t)\ \text{:-}\ reify(body^{+}(R),mmr,t),}\\ \quad\mathtt{vs(t).}\end{array} & \begin{array}{l} R \in normal(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\end{array}\\
\begin{array}{l}\mathtt{mmr(\bot,t)\ \text{:-}\ reify(body^{+}(R),mmr,t),}\\ \quad\mathtt{vs(t).}\end{array} & \begin{array}{l} R \in constraint(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\end{array}\\
\begin{array}{l}\mathtt{mmr(h,t)\ \text{:-}\ reify(body^{+}(R),mmr,t),}\\ \quad\mathtt{vs(t).}\end{array} & \begin{array}{l} R \in choice(ground(P)),\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\\ head(R) = \mathtt{l}\{\mathtt{h_1}, \ldots, \mathtt{h_n}\}\mathtt{u},\\ h \in \{h_1, \ldots h_n\} \cap I\\ l \leq |\{h_1, \ldots h_n\} \cap I| \leq u\end{array}\\
\begin{array}{l}\mathtt{mmr(\bot,t)\ \text{:-}\ reify(body^{+}(R),mmr,t),}\\ \quad\mathtt{vs(t).}\end{array} & \begin{array}{l} R \in choice(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\\ head(R) = \mathtt{l}\{\mathtt{h_1}, \ldots, \mathtt{h_n}\}\mathtt{u},\\ l > |\{h_1, \ldots h_n\} \cap I|\end{array}\\
\begin{array}{l}\mathtt{mmr(\bot,t)\ \text{:-}\ reify(body^{+}(R),mmr,t),}\\ \quad\mathtt{vs(t).}\end{array} & \begin{array}{l} R \in choice(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\\ head(R) = \mathtt{l}\{\mathtt{h_1}, \ldots, \mathtt{h_n}\}\mathtt{u},\\ |\{h_1, \ldots h_n\} \cap I| > u\end{array}
\end{array}\right\}\right)
$$

(As the literals which were removed were definitely true given the facts in the program)

$$
= M\left(\begin{array}{l}reify(I, in\_vs, t)\cup\\ \{\mathtt{vs(t)}\}\cup\end{array}\left\{\begin{array}{l|l}
\mathtt{mmr(head,t)\ \text{:-}\ reify(body^{+}(R),mmr,t).} & \begin{array}{l} R \in normal(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\end{array}\\
\mathtt{mmr(\bot,t)\ \text{:-}\ reify(body^{+}(R),mmr,t).} & \begin{array}{l} R \in constraint(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\end{array}\\
\mathtt{mmr(h,t)\ \text{:-}\ reify(body^{+}(R),mmr,t).} & \begin{array}{l} R \in choice(ground(P)),\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\\ head(R) = \mathtt{l}\{\mathtt{h_1}, \ldots, \mathtt{h_n}\}\mathtt{u},\\ h \in \{h_1, \ldots h_n\} \cap I\\ l \leq |\{h_1, \ldots h_n\} \cap I| \leq u\end{array}\\
\mathtt{mmr(\bot,t)\ \text{:-}\ reify(body^{+}(R),mmr,t).} & \begin{array}{l} R \in choice(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\\ head(R) = \mathtt{l}\{\mathtt{h_1}, \ldots, \mathtt{h_n}\}\mathtt{u},\\ l > |\{h_1, \ldots h_n\} \cap I|\end{array}\\
\mathtt{mmr(\bot,t)\ \text{:-}\ reify(body^{+}(R),mmr,t).} & \begin{array}{l} R \in choice(ground(P))\\ reify(body^{-}(R), in\_vs, t)\\ \quad \cap\, reify(I, in\_vs, t) = \emptyset\\ head(R) = \mathtt{l}\{\mathtt{h_1}, \ldots, \mathtt{h_n}\}\mathtt{u},\\ |\{h_1, \ldots h_n\} \cap I| > u\end{array}
\end{array}\right\}\right)
$$

(As the literals which were removed were definitely true given the facts in the program)

$$
= M\left(\left\{\begin{array}{l|l}
\texttt{mmr(head,t) :- reify(body}^{+}\texttt{(R),mmr,t).} & \begin{array}{l} R \in normal(ground(P)) \\ reify(body^{-}(R), in\_vs, t) \\ \quad \cap\, reify(I, in\_vs, t) = \emptyset \end{array} \\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^{+}\texttt{(R),mmr,t).} & \begin{array}{l} R \in constraint(ground(P)) \\ reify(body^{-}(R), in\_vs, t) \\ \quad \cap\, reify(I, in\_vs, t) = \emptyset \end{array} \\
\texttt{mmr(h,t) :- reify(body}^{+}\texttt{(R),mmr,t).} & \begin{array}{l} R \in choice(ground(P)), \\ reify(body^{-}(R), in\_vs, t) \\ \quad \cap\, reify(I, in\_vs, t) = \emptyset \\ head(R) = \texttt{l\{h}_1\texttt{,}\ldots\texttt{,h}_n\texttt{\}u}, \\ h \in \{h_1, \ldots h_n\} \cap I \\ l \leq |\{h_1, \ldots h_n\} \cap I| \leq u \end{array} \\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^{+}\texttt{(R),mmr,t).} & \begin{array}{l} R \in choice(ground(P)) \\ reify(body^{-}(R), in\_vs, t) \\ \quad \cap\, reify(I, in\_vs, t) = \emptyset \\ head(R) = \texttt{l\{h}_1\texttt{,}\ldots\texttt{,h}_n\texttt{\}u}, \\ l > |\{h_1, \ldots h_n\} \cap I| \end{array} \\
\texttt{mmr(}\bot\texttt{,t) :- reify(body}^{+}\texttt{(R),mmr,t).} & \begin{array}{l} R \in choice(ground(P)) \\ reify(body^{-}(R), in\_vs, t) \\ \quad \cap\, reify(I, in\_vs, t) = \emptyset \\ head(R) = \texttt{l\{h}_1\texttt{,}\ldots\texttt{,h}_n\texttt{\}u}, \\ |\{h_1, \ldots h_n\} \cap I| > u \end{array}
\end{array}\right\}\right) \cup reify(I, in\_vs, t) \cup \{\texttt{vs(t)}\}
$$

$$
= M(reify(\left\{\begin{array}{l|l}
\texttt{head :- body}^{+}\texttt{(R).} & \begin{array}{l} R \in normal(ground(P)) \\ body^{-}(R) \cap I = \emptyset \end{array} \\
\bot\texttt{ :- body}^{+}\texttt{(R).} & \begin{array}{l} R \in constraint(ground(P)) \\ body^{-}(R) \cap I = \emptyset \end{array} \\
\texttt{h :- body}^{+}\texttt{(R).} & \begin{array}{l} R \in choice(ground(P)), \\ body^{-}(R) \cap I = \emptyset \\ head(R) = \texttt{l\{h}_1\texttt{,}\ldots\texttt{,h}_n\texttt{\}u}, \\ h \in \{h_1, \ldots h_n\} \cap I \\ l \leq |\{h_1, \ldots h_n\} \cap I| \leq u \end{array} \\
\bot\texttt{ :- body}^{+}\texttt{(R).} & \begin{array}{l} R \in choice(ground(P)) \\ body^{-}(R) \cap I = \emptyset \\ head(R) = \texttt{l\{h}_1\texttt{,}\ldots\texttt{,h}_n\texttt{\}u}, \\ l > |\{h_1, \ldots h_n\} \cap I| \end{array} \\
\bot\texttt{ :- body}^{+}\texttt{(R).} & \begin{array}{l} R \in choice(ground(P)) \\ body^{-}(R) \cap I = \emptyset \\ head(R) = \texttt{l\{h}_1\texttt{,}\ldots\texttt{,h}_n\texttt{\}u}, \\ |\{h_1, \ldots h_n\} \cap I| > u \end{array}
\end{array}\right\}, mmr, t)) \cup reify(I, in\_vs, t) \cup \{\texttt{vs(t)}\}
$$

$$= M(reify(P^I), mmr, t) \cup reify(I, in\_vs, t) \cup \{\texttt{vs(t)}\}$$

$$= reify(M(P^I), mmr, t) \cup reify(I, in\_vs, t) \cup \{\texttt{vs(t)}\}$$

Hence $M(P^I) = \{\texttt{atom} \mid \texttt{mmr(atom,t)} \in M(Q)\}$

□

Lemma 7.11. Given any program P, term $\texttt{t}$ and interpretation I:

Let $Q = reductify(P, t) \cup reify(I, in\_vs, t) \cup \{\texttt{vs(t)}\}$

$I \in AS(P)$ if and only if $\texttt{nas(t)} \notin M(Q \cup nas\_rules(t))$.

Proof. Assume $I \notin AS(P)$

$\Leftrightarrow (\exists \mathtt{a} \in M(P^I) \text{ st } \mathtt{a} \notin I) \vee (\exists \mathtt{a} \in I \text{ st } \mathtt{a} \notin M(P^I))$.

Case 1: $\exists \mathtt{a} \in M(P^I)$ st $\mathtt{a} \notin I$

$\Leftrightarrow \exists \mathtt{a} \in \{\mathtt{atom} \mid \mathtt{mmr(atom, t)} \in M(Q)\}$ st $\mathtt{a} \notin I$ (By lemma 7.10).

$\Leftrightarrow \exists a$ st $mmr(a,t) \in M(Q)$ and $in\_vs(a,t) \notin M(Q)$ (as the in_vs atoms in $M(Q)$ are $reify(I, in\_vs, t)$ and $a \notin I$).

$\Leftrightarrow \exists R^g_1$ (a ground instance of $R_1(t)$) st $body(R^g_1)$ is satisfied by $M(Q)$.

Case 2: $\exists \mathtt{a} \in I$ st $\mathtt{a} \notin M(P^I)$

$\Leftrightarrow \exists \mathtt{a} \in I$ st $\mathtt{a} \notin \{\mathtt{atom} \mid \mathtt{mmr(atom, t)} \in M(Q)\}$ (By lemma 7.10).

$\Leftrightarrow \exists \mathtt{a}$ st $\mathtt{in\_vs(a, t)} \in M(Q)$ and $\mathtt{mmr(a, t)} \notin M(Q)$. (as the in_vs atoms in $M(Q)$ are $reify(I, in\_vs, t)$ and $a \in I$).

$\Leftrightarrow \exists R^g_2$ (a ground instance of $R_2(t)$) st $body(R^g_2)$ is satisfied by $M(Q)$.

Hence:

$I \notin AS(P)$

$\Leftrightarrow (\exists R^g_1 \in ground(R_1(t))$ st $M(Q)$ satisfies $body(R^g_1)) \vee (\exists R^g_2 \in ground(R_2(t))$ st $M(Q)$ satisfies $body(R^g_2))$.

$\Leftrightarrow \exists R^g \in ground(nas\_rules(t))$ st $M(Q)$ satisfies $body(R^g)$.

$\Leftrightarrow nas(t) \in M(M(Q) \cup nas\_rules(t))$

$\Leftrightarrow nas(t) \in M(Q \cup nas\_rules(t))$ as $nas(t)$ occurs nowhere in Q. □

Lemma 7.12. Let $H \subseteq S_M$

$VI_{meta}(T)[H]$ is satisfiable if and only if $\forall I \in VI, I \notin AS(B \cup H)$.

Proof. Assume $VI_{meta}(T)[H]$ is satisfiable.

$$\Leftrightarrow reductify(B \cup H, X) \cup nas\_rules(X) \cup \left\{ \begin{array}{c} \{\mathtt{:\text{-}\ not\ nas(I_{id}).\ \ vs(I_{id}).}\} \\ reify(I, in\_vs, I_{id}) \end{array} \middle| I \in VI \right\} \text{ is satisfiable.}$$

$$\Leftrightarrow \left\{ \begin{array}{c} \{\mathtt{:\text{-}\ not\ nas(I_{id}).\ \ vs(I_{id}).}\} \\ reify(I, in\_vs, I_{id}) \\ reductify(B \cup H, I_{id}) \\ nas\_rules(I_{id}) \end{array} \middle| I \in VI \right\} \text{ is satisfiable (as the relevant grounding is the same).}$$

$$\Leftrightarrow \forall I \in VI : \left\{ \begin{array}{c} \{\mathtt{:\text{-}\ not\ nas(I_{id}).\ \ vs(I_{id}).}\} \\ reify(I, in\_vs, I_{id}) \\ reductify(B \cup H, I_{id}) \\ nas\_rules(I_{id}) \end{array} \right\} \text{ is satisfiable (by corollary 4.2).}$$

$$\Leftrightarrow \forall I \in VI : nas(I_{id}) \in M\left(\left\{ \begin{array}{c} \{\mathtt{vs(I_{id}).}\} \\ reify(I, in\_vs, I_{id}) \\ reductify(B \cup H, I_{id}) \\ nas\_rules(I_{id}) \end{array} \right\}\right) \text{ is satisfiable (by lemma ??).}$$

$\Leftrightarrow \forall I \in VI$: $I \notin AS(B \cup H)$ (by lemma 7.11).

□

Lemma 7.13. Let $\mathtt{t_1}$ and $\mathtt{t_2}$ be two distinct terms and L be a set of integers. Let I_1 and I_2 be interpretations.

$$dominates(t_1, t_2) \cup \{\mathtt{lv(l)} \mid l \in L\} \cup \left\{ \begin{array}{l} meta_{weak}(weak(B \cup H), in\_vs, vs, t_i) \\ reify(I_i, in\_vs, t_i) \\ nas\_rules(t_i) \\ reductify(normal(B \cup H), t_i) \\ \{\mathtt{vs(vp_{id\ i}).}\} \end{array} \middle| i \in \{1,2\} \right\} \cup \left\{ \begin{array}{l} \mathtt{:\text{-}\ not}\ nas(t_1), \\ \quad\ \ \mathtt{not}\ nas(t_2), \\ \quad\ \ \mathtt{not}\ dom(t_1, t_2) \end{array} \right\}$$

is satisfiable if and only if $(I_1 \notin AS(B \cup H) \vee I_2 \notin AS(B \cup H) \vee I_1 \succ_{B \cup H} I_2)$.

Proof.

For $i \in \{1,2\}$ Let $Q_i = \left\{\begin{array}{l} reify(I_i, in\_vs, t_i) \\ \{\mathtt{vs(t_i)}\} \\ nas\_rules(t_i) \\ reductify(normal(B \cup H), t_i) \end{array}\right\}$

Let $Q_3 = dominates(t_1, t_2) \cup \{\mathtt{lv(l)} \mid l \in L\} \cup \left\{\begin{array}{l} meta_{weak}(weak(B \cup H), in\_vs, vs, t_i) \\ reify(I_i, in\_vs, t_i) \\ \{\mathtt{vs(t_i)}\} \end{array}\middle| i \in \{1,2\}\right\}$

$AS(Q_1 \cup Q_2 \cup Q_3) = \left\{A_1 \cup A_2 \middle| \begin{array}{c} A_1 \in AS(Q_1 \cup Q_2), \\ A_2 \in AS(Q_3) \end{array}\right\}$ as the only atoms in both $ground(Q_1 \cup Q_2)$ and $ground(Q_3)$ are facts in both programs.

$\therefore M(Q_1 \cup Q_2 \cup Q_3) = M(Q_1 \cup Q_2) \cup M(Q_3)$

Hence $\left\{\begin{array}{l} \mathtt{:-\ not}\ nas(t_1), \\ \quad \mathtt{not}\ nas(t_2), \\ \quad \mathtt{not}\ dom(t_1, t_2) \end{array}\right\} \cup Q$ is satisfiable

$\Leftrightarrow \left\{\begin{array}{l} \mathtt{:-\ not}\ nas(t_1), \\ \quad \mathtt{not}\ nas(t_2), \\ \quad \mathtt{not}\ dom(t_1, t_2) \end{array}\right\} \cup M(Q_1 \cup Q_2) \cup M(Q_3)$ is satisfiable

$\Leftrightarrow \mathtt{nas(t_1)} \in M(Q_1 \cup Q_2) \lor \mathtt{nas(t_2)} \in M(Q_1 \cup Q_2) \lor \mathtt{dom(t_1, t_2)} \in M(Q_3)$.

$\Leftrightarrow \mathtt{nas(t_1)} \in M(Q_1) \lor \mathtt{nas(t_2)} \in M(Q_2) \lor \mathtt{dom(t_1, t_2)} \in M(Q_3)$ (by corollary 4.2).

$\Leftrightarrow I_1 \notin AS(B \cup H) \lor I_2 \notin AS(B \cup H) \lor \mathtt{dom(t_1, t_2)} \in M(Q_3)$ (by lemma 7.11).

$\Leftrightarrow I_1 \notin AS(B \cup H) \lor I_2 \notin AS(B \cup H) \lor I_1 \succ_{B \cup H} I_2$ (by lemma **??**)

□

Lemma 7.14. Let $H \subseteq S_M$

$VP_{meta}(T)[H]$ is satisfiable if and only if $\forall \langle I_1, I_2 \rangle \in VP : I_1 \notin AS(B \cup H) \lor I_2 \notin AS(B \cup H) \lor I_1 \succ_{B \cup H} I_2$.

Proof.

Assume $VP_{meta}(T)[H]$ is satisfiable.

$\Leftrightarrow \{meta(vp) \mid vp \in VP\} \cup reductify(normal(B \cup H), X) \cup meta_{weak}(weak(B \cup H), in\_vs, vs, X)$

$\Leftrightarrow \left\{ meta(vp) \cup \left\{\begin{array}{l} meta_{weak}(weak(B \cup H), in\_vs, vs, vp_{id\ i}) \\ reductify(normal(B \cup H), vp_{id\ i}) \\ \{\mathtt{vs(vp_{id\ i}).}\} \end{array}\middle| i \in \{1,2\}\right\} \middle| vp \in VP \right\}$ is satisfiable.

(by partial grounding and removing rules whose bodies can't be satisfied).

$\Leftrightarrow \forall vp \in VP, meta(vp) \cup \left\{\begin{array}{l} meta_{weak}(weak(B \cup H), in\_vs, vs, vp_{id\ i}) \\ reductify(normal(B \cup H), vp_{id\ i}) \\ \{\mathtt{vs(vp_{id\ i}).}\} \end{array}\middle| i \in \{1,2\}\right\}$ is satisfiable.

$\Leftrightarrow \forall \langle I_1, I_2 \rangle \in VP : I_1 \notin AS(B \cup H) \lor I_2 \notin AS(B \cup H) \lor I_1 \succ_{B \cup H} I_2$ (by lemma 7.13). □

Theorem 7.15. *Let* $H \subseteq S_M$

$VR_{meta}(T)[H]$ *is satisfiable if and only if* H *is not a known violating hypothesis.*

Proof.

Assume H is not a known violating hypothesis.

$\Leftrightarrow \forall I \in VI, I \notin AS(B \cup H)$ and $\forall \langle I_1, I_2 \rangle \in VP : I_1 \notin AS(B \cup H) \lor I_2 \notin AS(B \cup H) \lor I_1 \succ_{B \cup H} I_2$.

$\Leftrightarrow$ (by lemma 7.12) $VI_{meta}(T)[H]$ is satisfiable and (by lemma 7.14) $VP_{meta}(T)[H]$ is satisfiable.

$\Leftrightarrow (VI_{meta}(T) \cup VP_{meta}(T))[H]$ is satisfiable.

□

8 Proof of the soundness and completeness of ILASP2

In this section we use the results that we have already proved in order to prove the soundness and completeness of $ILASP2$.

Lemma 8.1. For any task T, any hypothesis $H \subseteq S_M$ and any set of violating reasons VR:

$$AS(T_{meta}[H] \cup VR_{meta}(T)[H]) = \left\{ A_1 \cup A_2 \middle| \begin{array}{c} A_1 \in AS(T_{meta}(H), \\ A_2 \in AS(VR_{meta}(T)[H]) \end{array} \right\}$$

Proof.

The only predicate names which appear in both $T_{meta}[H]$ and $VR_{meta}(T)[H]$ are dom, lv, dom_lv, $weak$, $wrong\_bef$ and $wrong\_dom\_lv$.

Other than lv, as these are all parameterised by unique ids, none of the atoms in $T_{meta}[H]$ unify with any atom in $VR_{meta}(T)[H]$. The atoms lv only appear as facts in both programs and are the same facts in both.

Hence $AS(T_{meta}[H] \cup VR_{meta}(T)[H]) = \left\{ A_1 \cup A_2 \middle| \begin{array}{c} A_1 \in AS(T_{meta}[H]), \\ A_2 \in AS(VR_{meta}(T)[H]) \end{array} \right\}$ □

Theorem 8.2. *For any task T, any hypothesis $H \subseteq S_M$ and any set of violating reasons VR :*

$(T_{meta} \cup VR_{meta}(T))[H]$ is satisfiable if and only if H is a remaining hypothesis of T wrt VR.

Proof.

Assume $(T_{meta} \cup VR_{meta}(T))[H]$ is satisfiable

$\Leftrightarrow T_{meta}[H] \cup VR_{meta}(T)[H]$ is satisfiable

$\Leftrightarrow T_{meta}[H]$ satisfiable and $VR_{meta}(T)[H]$ is satisfiable (by lemma 8.1).

$\Leftrightarrow H$ is a positive hypothesis of T and H is a remaining hypothesis of T wrt VR (by theorem 6.24 and theorem 7.15). □

Corollary 8.3. For any task T, any hypothesis $H \subseteq S_M$ and any set of violating reasons VR:

$\exists A \in AS(T_{meta} \cup VR_{meta}(T))$ such that $meta^{-1}_{hyp}(A) = H$ if and only if H is a remaining positive hypothesis.

Proof. Follows from theorem 8.2 and lemma 4.8. □

Lemma 8.4. For any task T, any hypothesis $H \subseteq S_M$ and any set of violating reasons VR:

$\forall A \in AS(T_{meta} \cup VR_{meta}(T))$

$P^0_A = |\mathcal{M}^{-1}_{hyp}(A)| + v$ (where v is 0 if A contains $violating$ and 1 otherwise).

Proof.

Let $P = T_{meta} \cup VR_{meta}(T)$

$weak(P) = \{:\sim \texttt{in\_hyp}(\texttt{R}_{\texttt{id}}).[2 * |\texttt{R}|@0, \texttt{R}_{\texttt{id}}] \mid R \in S_M\} \cup \{:\sim \texttt{not violating}.[1@0, \texttt{violating}]\}$

$\Rightarrow \forall A \in AS(P)$, $W \in weak(P, A)$ iff $W = (w, 0, t)$ st $(t = violating \wedge w = 1) \vee (\exists R \in S_M$ st $t = R_{id} \wedge w = |R|)$

$\Rightarrow P^0_A = \left(\sum_{R \in S_M, in\_hyp(R_{id}) \in A} |R| \right) + v$ where v is 0 if $violating \in A$ and 1 otherwise.

(as 0 is the only level with any weak constraints we can just call this the optimality)

$\Rightarrow P^0_A = |\mathcal{M}^{-1}_{hyp}(A)| + v$ (where v is 0 if A contains $violating$ and 1 otherwise).

□

Theorem 8.5. *Given any hypothesis $H \subseteq S_M$:*

$\exists A \in AS(T_{meta} \cup VR_{meta})$ *such that* $\texttt{v\_i} \in A$ *and* $H = \mathcal{M}^{-1}_{hyp}(A)$

$\Leftrightarrow$ *H is a remaining violating hypothesis and $\mathcal{M}^{-1}_{vi}(A)$ is a violating interpretation of $B \cup H$.*

Proof.

Let $A \in AS(T_{meta} \cup VR_{meta})$ and assume $H = \mathcal{M}^{-1}_{hyp}(A)$, $\texttt{v\_i} \in A$ and $V = \mathcal{M}^{-1}_{vi}(A)$.

$\Leftrightarrow \exists A \in AS((T_{meta} \cup VR_{meta})[H])$ st $\texttt{v\_i} \in A$ and $V = \mathcal{M}^{-1}_{vi}(A)$ (by lemma 4.8)

$\Leftrightarrow VR_{meta}(T)[H]$ is satisfiable and $\exists A \in T_{meta}(H)$ st $v\_i \in A$ and $\mathcal{M}^{-1}_{vi}(A) = V$

(by lemma 8.1 and as no instance of *in_as* occurs in $VR_{meta}(T)[H]$).

$\Leftrightarrow H$ is a remaining hypothesis of T (by theorem 7.15) and a violating hypothesis with violating interpretation V

(by theorem 6.24).

$\Leftrightarrow H$ be a remaining violating hypothesis with a violating interpretation V

□

Theorem 8.6. *Given any hypothesis $H \subseteq S_M$:*

$\exists A \in AS(T_{meta} \cup VR_{meta})$ *st $v\_p \in A$ and* $H = \mathcal{M}^{-1}_{hyp}(A)$

$\Leftrightarrow$ *H is a remaining violating hypothesis with violating pair $\mathcal{M}^{-1}_{vp}(A)$.*

Proof.

Let H be a remaining violating hypothesis of T with violating pair vp.

$\Leftrightarrow VI_{meta}(T)[H]$ is satisfiable by theorem 7.15 and $\exists A \in T_{meta}(H)$ st $\texttt{v\_p} \in A$ and $\mathcal{M}^{-1}_{vp}(A) = vp$ (by theorem 6.24).

$\Leftrightarrow \exists A \in AS((T_{meta} \cup VR_{meta})[H])$ st $\texttt{v\_p} \in A$ and $\mathcal{M}^{-1}_{vp}(A) = vp$ is a violating pair.

(by lemma 8.1 and the fact that no atom with the predicate *in_as* appears in $VR_{meta}(T)[H]$)

$\Leftrightarrow \exists A \in AS(T_{meta} \cup VR_{meta})$ st $H = \mathcal{M}^{-1}_{hyp}(A)$, $\texttt{v\_p} \in A$ and $\mathcal{M}^{-1}_{vp}(A) = vp$ (by lemma 4.8).

□

Corollary 8.7. Given any hypothesis $H \subseteq S_M$:

$\exists A \in AS(T_{meta} \cup VR_{meta})$ st $\texttt{violating} \in A$ and $H = \mathcal{M}^{-1}_{hyp}(A)$ iff H is a remaining violating hypothesis of T.

Theorem 8.8 is Theorem 1 from the paper.

Theorem 8.8. *Given an ILP_{LOAS} task and a set of violating reasons VR. Let AS be the set of optimal Answer Sets of $T_{meta} \cup VR_{meta}(T)$.*

If $\exists A \in AS$ st violating $\in A$ then the set of optimal remaining violating hypotheses VH is non empty and is exactly equal to the set $\{\mathcal{M}^{-1}_{hyp}(A) : A \in AS\}$.

If no $A \in AS$ contains violating, then the set of optimal remaining hypotheses (none of which are violating) is exactly equal to the set $\{\mathcal{M}^{-1}_{hyp}(A) : A \in AS\}$.

Proof.

1. Assume $\exists A \in AS$ such that $\texttt{violating} \in A$.

   Let *opt* be the optimality of A.

   $\Rightarrow$ *opt* is an even number.

   $\Rightarrow \forall A \in AS$, $\texttt{violating} \in A$, the optimality of $A =$ *opt* (and AS is non empty).

   $\Rightarrow \forall A \in AS$, $\mathcal{M}^{-1}_{hyp}(A)$ is a remaining violating hypothesis of length *opt*/2 (by corollary 8.7 and lemma 8.4).

$\Rightarrow \forall A \in AS$, $\mathcal{M}^{-1}_{hyp}(A)$ is an optimal remaining violating hypothesis

(if there were any more optimal hypotheses remaining there would be a more optimal answer set).

Let H be an optimal remaining violating hypothesis

$\Rightarrow \exists A \in AS(T_{meta} \cup VR_{meta})$ st $\texttt{violating} \in A$ and $H = \mathcal{M}^{-1}_{hyp}(A)$ by corollary 8.7 and $|H| = opt/2$.

$\Rightarrow \exists A \in AS(T_{meta} \cup VR_{meta})$ st $\texttt{violating} \in A$ and $H = \mathcal{M}^{-1}_{hyp}(A)$ and the optimality of $A = opt$

(by lemma 8.4).

$\Rightarrow \exists A \in AS$ st $\texttt{violating} \in A$ and $H = \mathcal{M}^{-1}_{hyp}(A)$

Hence, $\{\mathcal{M}^{-1}_{hyp}(A) \mid A \in AS\}$ is exactly the set of optimal remaining violating hypotheses.

2. Assume no $A \in AS$ contains the atom `violating`. Let *opt* be the optimality of these Answer Sets.

$\Rightarrow \forall A \in AS$, $\mathcal{M}^{-1}_{hyp}(A)$ is a remaining hypothesis of length $(opt - 1)/2$ (by corollary 8.7 and lemma 8.4).

Assume there is a remaining violating hypothesis H of shorter or equal length to these hypotheses.

$\Rightarrow \exists A \in AS(T_{meta} \cup VR_{meta})$ st $\texttt{violating} \in A$ and $H = \mathcal{M}^{-1}_{hyp}(A)$ and the optimality of $A = 2 * |H|$

(by corollary 8.7 and lemma 8.4).

Contradiction as this would mean that there is a more optimal Answer Set than those in AS.

Assume there is a remaining hypothesis H of shorter length to these hypotheses.

$\Rightarrow H$ is a positive hypothesis.

$\Rightarrow \exists A \in AS(T_{meta} \cup VR_{meta})$ st $\texttt{violating} \in A$ and $H = \mathcal{M}^{-1}_{hyp}(A)$ and the optimality of $A = 2*|H|+1$

(by corollary 8.7 and lemma 8.4).

Contradiction as this would mean that there is a more optimal Answer Set than those in AS.

Assume there is a remaining hypothesis H of equal length to these hypotheses.

$\Rightarrow H$ is a positive hypothesis.

$\Rightarrow \exists A \in AS(T_{meta} \cup VR_{meta})$ st $\texttt{violating} \in A$, $H = \mathcal{M}^{-1}_{hyp}(A)$ and the optimality of A is *opt*

(by corollary 8.7 and lemma 8.4).

$\Rightarrow H \in \{\mathcal{M}^{-1}_{hyp}(A) \mid A \in AS\}$.

Hence, $\{\mathcal{M}^{-1}_{hyp}(A) \mid A \in AS\}$ is exactly the set of optimal remaining hypotheses, none of which are violating.

□

Theorem 8.9 is Theorem 2 from the paper, it proves the soundness and completeness of our new algorithm, WCL.

Theorem 8.9. *Let T be any ILP_{LOAS} task. If the process $WCL(T)$ terminates, then $WCL(T)$ is equal to the set of optimal solutions of $ILP_{LOAS}(T)$.*

Proof. At every step through the while loop, VR is a set violating reasons of T.

Base Case: Before the loop has been entered, $VR = []$.

Inductive Hypothesis: Let VR_0 be a set of violating reasons. If $VR = VR_0$ at the start of an interation through the loop, then VR_1, the value of VR after one iteration of the loop, is still a set of violating reasons of T.

Proof of Inductive Hypothesis:

Case 1: $\texttt{v\_i} \notin A \wedge \texttt{v\_p} \notin A$.

$\Rightarrow VR_0 = VR_1$

$\Rightarrow VR_1$ is a set of violating reasons.

Case 2: $\texttt{v\_i} \in A$.

$\Rightarrow vi = \mathcal{M}_{vi}^{-1}(A)$ is a violating interpetation of $B \cup \mathcal{M}_h^{-1}(A)$ (by theorem 8.5)

$\Rightarrow VR_1 = VR_0 + vi$ is a set of violating reasons of T.

Case 3: $\texttt{v\_p} \in A$.

$\Rightarrow vp = \mathcal{M}_{vp}^{-1}(A)$ is a violating pair of $B \cup \mathcal{M}_h^{-1}(A)$ (by theorem 8.6).

$\Rightarrow VR_1 = VR_0 + vp$ is a set of violating reasons of T.

Hence at each step through the loop, VR is a set of violating reasons of T.

When $WCL(T)$ terminates, either *opt* is odd or $T_{meta} \cup VR_{meta}$ has no answer sets.

Case 1: *opt* is odd.

Each $as \in AS_{opt}(T_{meta} \cup VR_{meta}(T))$ has optimality *opt*; hence, by lemma 8.4, $v\_i \notin as, v\_p \notin as$ and $violating \notin as$. Hence by theorem 8.8, $\{\mathcal{M}_{hyp}^{-1}(as) \mid as \in AS_{opt}(T_{meta} \cup VR_{meta}(T))\}$ is the set of optimal remaining hypotheses, none of which are violating. This means that they are the optimal inductive solutions of T.

Case 2: $T_{meta} \cup VR_{meta}$ has no answer sets.

There are no remaining positive hypotheses (by corollary 8.3). Hence, as VR is a set of violating reasons, there are no inductive hypotheses. So $ILP_{LOAS}(T) = \emptyset = ILASP2(T)$.

□

Part II

Theoretical Properties

9 Sufficient and Necessary conditions for existence of solutions

In this section, we prove sufficient and necessary conditions for a Learning from Ordered Answer Sets task to have at least one solution. We consider a learning task with an unrestricted search space (hypotheses can be any set of normal rules, choice rules and hard and soft constraints).

Theorem 9.1. Let T be the ILP_{LOAS} task $\langle B, E^{+}, E^{-}, O^{b}, O^{c}\rangle$. The following conditions (in conjunction) are sufficient for there to exist at least one solution of T:

1. $\forall e \in E^{+}$, there is at least one model of B which extends e.
2. $\forall e_1 \in E^{+}$, $\nexists e_2 \in (E^{+} \cup E^{-})$ such that e_1 extends e_2.
3. There is no cyclic chain of ordering examples (in $O^{b} \cup O^{c}$) $\langle e_1, e_2\rangle, \langle e_2, e_3\rangle, \ldots, \langle e_{n-1}, e_n\rangle, \rangle, \langle e_n, e_1\rangle$

Proof. We show this by assuming that all of the conditions hold and constructing an inductive solution H.

For each positive example $e_i^{+} \in E^{+}$, we know by condition 1 that there is at least one model of B which does extends e_i^{+}. We denote this model as $M_{pos}(e_i^{+})$.

As there are no cycles in the ordering examples, there must be a mapping $||\ ||$ from positive examples to positive integers such that for each ordering example $\langle e_1, e_2\rangle \in (O^{b} \cup O^{c})$, $||e_1|| > ||e_2||$.

Let $s_1, \ldots, s_{|E^{+}|}$ be new atoms not occuring in B or the examples H be the program:

$$\left\{\ \mathtt{1\{s_1,\ldots,s_{|E^+|}\}1.}\ \right\} \cup \left\{ \begin{array}{c|c} \mathtt{:\sim\ s_i.[-1@||e_i||]} & \\ \mathtt{m_1 \texttt{ :- } s_i.} & \\ \ldots & \{m_1, \ldots, m_n\} = M_{pos}(e_i) \\ \mathtt{m_n \texttt{ :- } s_i.} & e_i = \langle\{inc_1, \ldots, inc_t\}, \{exc_1, \ldots, exc_v\}\rangle \in E^{+} \\ \mathtt{\texttt{:- } s_i, exc_1....} & \\ \mathtt{\texttt{:- } s_i, exc_v.} & \end{array} \right\}$$

For each positive example $e_i^{+} \in E^{+}$, $M_{pos}(e_i) \cup \{s_i\}$ is an answer set of $B \cup H$ and hence as $M_{pos}(e_i)$ extends e_i^{+}, H covers all the positive examples.

Every answer set of $B \cup H$ must contain exactly one s_i atom and hence cover at least one positive example (as $e_i^{inc} \subseteq M_{pos}(e_i)$ and there are constraints ruling out any atom in e^{exc} occuring together with s_i). Hence, as no positive examples extend other positive examples or negative examples, each answer set of $B \cup H$ extends exactly one positive example (corresponding to the s_i atom it contains) and no negative examples (hence, the negative examples are covered).

It remains to show that the ordering examples are all respected; in fact, we can show that all the ordering examples (brave and cautious) are cautiously respected (and hence also bravely respected as there are answer sets which cover each positive example).

For any ordering example $\langle e_1, e_2\rangle \in (O^{b} \cup O^{c})$, each answer set A_1 of $B \cup H$ that extends e_1 contains s_1 (and no other s_i atom); similarly each answer set A_2 of $B \cup H$ that extends e_2 contains s_2 and no other s_2 atom. For all levels l in the program other than $||e_1||$, $(B \cup H)^{l}_{A_1} = 0$ and for all levels l other than $||e_2||$, $(B \cup H)^{l}_{A_2} = 0$. Hence, as $||e_1|| > ||e_2||$, the first level on which A_1 and A_2 differ is $||e_1||$ for which A_1's score is -1 and A_2's is 0. Hence, A_1 dominates A_2.

So all the brave and cautious ordering examples are both bravely and cautiously respected.

Hence, H is an inductive solution of T.

□

We now give a slightly weakened condition which is necessary for there to exist solutions of a learning from ordered answer sets task.

Theorem 9.2. Let T be the ILP_{LOAS} task $\langle B, E^{+}, E^{-}, O^{b}, O^{c}\rangle$. The following conditions are all necessary for there to exist at least one solution of T:

1. $\forall e \in E^{+}$, there is at least one model of B which extends e.
2. $\forall e_1 \in E^{+}$, $\nexists e_2 \in E^{-}$ such that e_1 extends e_2.
3. There is no cyclic chain of ordering examples (in O^{c}) $\langle e_1, e_2\rangle, \langle e_2, e_3\rangle, \ldots, \langle e_{n-1}, e_n\rangle, \langle e_n, e_1\rangle$.

Proof. We prove this by assuming that there is an inductive solution $H \in ILP_{LOAS}(T)$ and show that each of the conditions (1)-(3) must hold.

1. For each positive example $e \in E^{+}$, there is an answer set A of $B \cup H$ which extends e. A is a model of $B \cup H$ (as all answer sets are models) and hence is also a model of B.
2. Assume for contradiction that there is a positive example $e^{+} = \langle e^{+}_{inc}, e^{+}_{exc}\rangle$ and a negative example $e^{-} = \langle e^{-}_{inc}, e^{-}_{exc}\rangle$ such that $e^{-}_{inc} \subseteq e^{+}_{inc}$ and $e^{-}_{exc} \subseteq e^{+}_{exc}$.

   As H is an inductive solution of T, there is an answer set A of $B \cup H$ which extends e^{+}.

   So there is an $A \in AS(B \cup H)$ such that $e^{+}_{inc} \subseteq A$ and $e^{+}_{exc} \cap A = \emptyset$. Hence $e^{-}_{inc} \subseteq A$ and $e^{-}_{exc} \cap A = \emptyset$. So A extends e^{-}. Contradiction, as H was an inductive solution and therefore $B \cup H$ must have no answer sets which extend e^{-}!
3. Again, assume for contradiction that there is such a chain. As there is at least one answer set which extends each e_i, this implies both that all answer sets of $B \cup H$ which extend e_1 dominate all answer sets which extend e_n (domination is clearly transitive) and that all answer sets which extend e_n dominate all answer sets which extend e_1.

   As there is at least one answer set A_1 extending e_1 and one answer set A_n which extends e_n, this implies that A_1 dominates A_n and A_n dominates A_1. This is impossible, and hence a contradiction.

□

10 Complexity Results

In this section, we prove the complexities of ILP_{LOAS} and ILP_{LAS} with respect to deciding, given a learning task T, whether there are any inductive solutions of T. In this report we consider the propositional case (where both the background knowledge B and the hypothesis space S_M are ground). This decision problem is NP^{NP}-complete for both ILP_{LOAS} and ILP_{LAS}.

10.1 Learning from Answer Sets with Stratified Summing Aggregates

Before proving the complexity results, we introduce a new learning task, ILP^{s}_{LAS} (Learning from Answer Sets with Stratified Summing Aggregates). First we recall the definition of aggregate stratification from [5]. We slightly simplify the definition by considering only propositional programs without disjunction.

Definition 10.1. A propositional logic program P, only containing aggregates in the bodies of rules, is *stratified on an aggregate* a if there is a level mapping $|| \; ||$ from $Atoms(P)$ to ordinals, such that for each rule $R \in P$, the following holds:

1. $\forall b \in Atoms(body(R)) : ||b|| \leq ||head(R)||$
2. If $a \in body(R)$, then $\forall b \in Atoms(a) : ||b|| < ||head(R)||$

P is said to be aggregate stratified if it is stratified on every aggregate in P.

The intuition is that aggregate stratification forbids recursion through aggregates.

Note that constraints and choice rules can be added in to any aggregate stratified program without breaking stratification so long as no atoms in the head of the choice rule are on a lower level than any atom in the body. This is illustrated by the following example.

Example 10.2. Any constraint $\texttt{:-}\,\mathtt{b_1, \ldots, b_n}$, $\texttt{not}\ \mathtt{c_1}, \ldots,$ $\texttt{not}\ \mathtt{c_m}$ can be rewritten as $\mathtt{s}\ \texttt{:-}\,\mathtt{b_1, \ldots, b_n}$, $\texttt{not}\ \mathtt{c_1}, \ldots,$ $\texttt{not}\ \mathtt{c_m}$, $\texttt{not s}$ where s is a new atom. s can then be mapped to a higher level than any other atom.

A choice rule $\mathtt{l\{h_1, \ldots, h_o\}u}\ \texttt{:-}\,\mathtt{b_1, \ldots, b_n}$, $\texttt{not}\ \mathtt{c_1, \ldots, c_m}$ can be rewritten as:

```
h1 :- b1, ..., bn, not c1, ..., cm, not h1'.
h1' :- b1, ..., bn, not c1, ..., cm, not h1.
                    ...
ho :- b1, ..., bn, not c1, ..., cm, not ho'.
ho' :- b1, ..., bn, not c1, ..., cm, not ho.

s :- b1, ..., bn, not c1, ...,, cm, {h1, ..., hn} l - 1,  not s.
s' :- b1, ..., bn, not c1, ...,, cm, u + 1 {h1, ..., hn},  not s'.
```

where $h'_1, \ldots, h'_o, s, s'$ are all new atoms. s and s' can both be given a new highest level and each h'_i can be given the same level as h_i (if they did not occur in the previous program then they should be given a new level one below s and s'). Provided the previous program was aggregate stratified, then this new one is too.

To avoid constantly using this mapping, we will refer to programs with choice rules and constraints as also being aggregate stratified.

Lemma 10.3. Deciding whether an aggregate stratified propositional program without disjunction cautiously entails an atom is is co-NP-complete [5].

Corollary 10.4. Deciding whether an aggregate stratified propositional program without disjunction bravely entails an atom is is NP-complete.

We can now introduce our extra learning task, Learning from Answer Sets with Stratified Aggregates. It is essentially the same as Learning from Answer Sets, but allowing summing aggregates in the bodies of rules, so long as they are stratified.

Definition 10.5. A *Learning from Answer Sets with Stratified Aggregates* task is a tuple $T = \langle B, S_M, E^+, E^- \rangle$ where B is program which can contain summing aggregates in the bodies of rules called the background knowledge, S_M is a set of rules which possibly contain summing aggregates called the search space and E^- are sets of partial interpretations called, respectively, positive and negative examples.

For this task to be well defined, $B \cup S_M$ must be aggregate stratified.

An hypothesis H is an *inductive solution* of T, written $H \in ILP^s_{LAS}(T)$, if and only if $H \subseteq S_M$; $\forall e^+ \in E^+$ $\exists A \in AS(B \cup H)$ such that A extends e^+; and finally, $\forall e^- \in E^-$ $\nexists A \in AS(B \cup H)$ such that A extends e^-.

Note that the condition of $B \cup S_M$ being aggregate stratified, implies that for any hypothesis $H \subseteq S_M$, $B \cup H$ is aggregate stratified.

10.2 Proof of Complexity

As we can show that ILP_{LAS} reduces to ILP_{LOAS} and ILP_{LOAS} reduces to ILP^s_{LAS} (in polynomial time), it suffices to show that ILP_{LAS} is NP^{NP}-hard (thus also proving the hardness of the other frameworks) and that ILP^s_{LAS} is a member of NP^{NP} (thus proving membership of the other frameworks). This shows that each framework is both a member of NP^{NP} and also NP^{NP}-hard, and therefore must be NP^{NP}-complete.

Lemma 10.6. Deciding whether an ILP_{LOAS} task has any solutions reduces polynomially to deciding whether ILP^s_{LAS} has any solutions.

Proof. To show this, we use part of the ILASP2 meta encoding in the background knowledge.

As we cannot use non-ground atoms in our reduction, we use a slightly different representation of *dominates*$(t1, t2)$ described in section 5.1.

Let *weak_atoms*(l, P) define a set of tuples representing the weak constraints at level l in P. Each weak constraint W is represented by a tuple $(atom^1, atom^2, wt, body)$ (where $atom^1$ and $atom^2$ are new atoms unique to W, wt is the weight of W and $body$ is the body of W.

$$dominates(t1,t2) = \left\{ \begin{array}{l|l} \texttt{dom\_lv(t1,t2,l)} \text{ :- } & \\ \quad \#\texttt{sum}\{\texttt{atom}^1_1 = \texttt{wt}_1, \ldots, \texttt{atom}^1_m = \texttt{wt}_m, & \\ \quad\quad \texttt{atom}^2_1 = \texttt{wt}_1, \ldots, \texttt{atom}^2_n = \texttt{wt}_n\} & l \text{ is a level in } B \cup H, \\ \texttt{non\_dom\_lv(t1,t2,l)} \text{ :- } & \{(atom^1_1, atom^2_1, wt^1_1, body^1_1), \ldots, \\ \quad \#\texttt{sum}\{\texttt{atom}^2_1 = \texttt{wt}_1, \ldots, \texttt{atom}^2_m = \texttt{wt}_m, & (atom^1_m, atom^2_m, wt_m, body_m)\} = weak\_atoms(l, B \cup S_M) \\ \quad\quad \texttt{atom}^1_1 = \texttt{wt}_1, \ldots, \texttt{atom}^1_n = \texttt{wt}_n\} & \\ \texttt{dom(t1,t2)} \text{ :- } \texttt{dom\_lv(t1,t2,l)}, & \\ \quad \texttt{not non\_bef(t1,t2,l)}. & \end{array} \right\}$$
$$\cup \left\{ \texttt{non\_bef(t1,t2,l}_1\texttt{)} \text{ :- } \texttt{non\_dom\_lv(t1,t2,l}_2\texttt{)}. \;\middle|\; \begin{array}{l} l_1, l_2 \text{ are levels in } B \cup S_M, \\ l_1 < l_2 \end{array} \right\}$$

This is essentially a ground version of *dominates*$(t1, t2)$.

Let

$$B' = \left\{ \begin{array}{l|l} \texttt{atom}^1 \text{ :- } \texttt{reify(body,in\_as,1)}. & (atom^1, atom^2, wt, body) \in weak\_atoms(l, B), \\ \texttt{atom}^2 \text{ :- } \texttt{reify(body,in\_as,2)}. & l \text{ is a level in } B \end{array} \right\}$$
$$\cup\, reify(B, in\_as, 1) \cup reify(B, in\_as, 2)$$
$$\cup \left\{ \begin{array}{l|l} append(reify(R, in\_as, 1), active(R_{id})), & R \in S_M \\ append(reify(R, in\_as, 2), active(R_{id})) & \end{array} \right\}$$
$$\cup \left\{ \begin{array}{l|l} \texttt{atom}^1 \text{ :- } \texttt{reify(body,in\_as,1),active(id)}. & (atom^1, atom^2, wt, body) \in weak\_atoms(l, weak(S_M)), \\ \texttt{atom}^2 \text{ :- } \texttt{reify(body,in\_as,2),active(id)}. & (id \text{ is a unique identifier for the weak constraint}), \\ & l \text{ is a level in } S_M, \end{array} \right\}$$
$$\cup \left\{ \begin{array}{l|l} cover(e^+_{id}, 1) & e^+ \in E^+ \\ cover(e^+_{id}, 2) & \end{array} \right\}$$
$$\cup \left\{ \begin{array}{l|l} cover(e^-_{id}, 1) & e^- \in E^- \\ cover(e^-_{id}, 2) & \end{array} \right\} \cup \left\{ \begin{array}{l} \texttt{as(1)} \\ \texttt{as(2)} \end{array} \right\} \cup dominates(1, 2)$$

$$S'_M = \left\{ \begin{array}{c} \texttt{active(id}_1\texttt{)}. \\ \ldots \\ \texttt{active(id}_{|S_M|}\texttt{)}. \end{array} \right\}$$

$E^{+'} = \{\langle\{cov(e^+_{id}, 1)\}, \emptyset\rangle | e^+ \in E^+\} \cup \{\langle\{cov(e^+_{id1}, 1), cov(e^+_{id2}, 2), dom(e^+_{id1}, e^+_{id2})\}, \emptyset\rangle | \langle e^+_{id1}, e^+_{id2}\rangle \in O^b\}$

$E^{-'} = \{\langle\{cov(e^-_{id}, 1)\}, \emptyset\rangle | e^- \in E^-\} \cup \{\langle\{cov(e^+_{id1}, 1), cov(e^+_{id2}, 2)\}, \{dom(e^+_{id1}, e^+_{id2})\}\rangle | \langle e^+_{id1}, e^+_{id2}\rangle \in O^c\}$

For any hypothesis $H' \in S'_M$, Let H be the corresponding hypothesis in S_M. The answer sets of $B' \cup H'$ correspond to the pairs of answer sets of $B \cup H$.

Each positive example $e^+ \in E^+$ is mapped to an example in $E^{+'}$ ensuring that at least one of the pairs of answer sets' first answer set covers e^+. Note that as each answer set of $B \cup H$ must be the first element of one of these pairs at least once, this is true if and only if $B \cup H$ covers each positive example.

Similarly each negative example $e^- \in E^-$ is mapped to an example in $E^{-'}$ ensuring that none of the pairs of answer sets' first answer set covers e^-. This is true if and only if $B \cup H$ does not cover any negative examples.

As $dominates(t1, t2)$ behaves similarly to $dominates(t1, t2)$ from section 5.1, the answer sets of $B' \cup H'$ corresponding to each pair $\langle A_1, A_2\rangle$ contains $dom(1, 2)$ if and only if A_1 dominates A_2 (with respect to the weak constraints in $B \cup H$.

Each brave ordering example $\langle e_1, e_2\rangle \in O^b$ is mapped to a positive example ensuring that there is a pair of answer sets $\langle A_1, A_2\rangle$ of $B \cup H$ such that A_1 covers e_1, A_2 covers e_2 and A_1 dominates A_2 with respect to the weak constraints in $B \cup H$. This is true if and only if $B \cup H$ bravely respects the ordering example.

Each cautious ordering example $\langle e_1, e_2\rangle \in O^c$ is mapped to a negative example ensuring that there is no pair of answer sets $\langle A_1, A_2\rangle$ of $B \cup H$ such that A_1 covers e_1, A_2 covers e_2 and A_1 dominates A_2 with respect to the weak constraints in $B \cup H$. This is true if and only if $B \cup H$ cautiously respects the ordering example.

Hence, H' is an inductive solution of $ILP^s_{LAS}(\langle B', S'_M, E^{+'}, E^{-'}\rangle)$ if and only if H is an inductive solution of $ILP_{LOAS}(\langle B, S_M, E^+, E^-, O^b, O^c\rangle)$.

This means that we can check the existence of solutions of any ILP_{LOAS} task by mapping if to an ILP^s_{LAS} task as above. Note that this is a well defined ILP^s_{LAS} task as B contains only stratified aggregates.

As this mapping is polynomial in size of the original task, this means that checking the existence of ILP_{LOAS} reduces polynomially to checking the existence of ILP^S_{LAS}.

□

Lemma 10.7. Deciding the existence of solutions for an ILP_{LAS} task reduces polynomially to deciding the existence of solutions for an ILP_{LOAS} task.

Proof. Take any ILP_{LAS} task $T = \langle B, S_M, E^+, E^-\rangle$. Clearly $ILP_{LAS}(T) = \emptyset$ if and only if $ILP_{LOAS}(\langle B, S_M, E^+, E^-, \emptyset, \emptyset\rangle) = \emptyset$. Hence checking the existence of a solution for T is equivalent to checking the existance of a solution to $\langle B, S_M, E^+, E^-, \emptyset, \emptyset\rangle$. □

Lemma 10.8. Let B be any ground program containing normal rules choice rules, constraints and summing aggregates in the body, S_M be a set of ground normal rules, choice rules and constraints and E^+ and E^- be any sets of partial interpretations. $B \cup S_M$ must also be aggregate stratified (ensuring that for each $H \subseteq S_M$, $B \cup H$ is aggregate stratified).

Deciding whether a given hypothesis $H \subset S_M$ is in $ILP^s_{LAS}(B, S_M, E^+, E^-)$ is a member of P^{NP}.

Proof. Checking whether H is an inductive solution of $T = \langle B, S_M, E^+, E^-\rangle$ can be done by checking for each positive example $e^+ \in E^+$, that there is an answer set A of $B \cup H$ such that A extends e^+ and for each negative example e^-, there is not any answer set of $B \cup H$ which extends e^-.

This is equivalent to checking that for each positive example $e^+ = \langle\{inc_1, \ldots, inc_n\}, \{exc_1, \ldots, exc_m\}\rangle$, $B \cup H \cup \{\mathtt{a} \leftarrow \mathtt{inc_1}, \ldots, \mathtt{inc_n},\ \mathtt{not\ exc_1}, \ldots,\ \mathtt{not\ exc_m}\} \models_b \mathtt{a}$ (where $\mathtt{a}$ is a new atom) and for each negative example $e^- = \langle\{inc_1, \ldots, inc_n\}, \{exc_1, \ldots, exc_m\}\rangle$, $B \cup H \cup \{\mathtt{a} \leftarrow \mathtt{inc_1}, \ldots, \mathtt{inc_n},\ \mathtt{not\ exc_1}, \ldots,\ \mathtt{not\ exc_m}\} \not\models_b \mathtt{a}$ (where $\mathtt{a}$ is a new atom).

As deciding whether an atom is bravely entailed by an aggregate stratified propositional program (containing normal rules, choice rules, constraints and summing aggregates in the bodys) is in NP 10.4, the property can be verified in polynomial time by a deterministic Turing machine with an oracle capable of solving problems in NP.

Hence verifying the property is in P^{NP}.

□

Due to this result on the verification of a solution, we can now show the related result for deciding the existance of a solution for a given learning task.

Lemma 10.9. Let B be any ground program containing normal rules choice rules, constraints and summing aggregates in the body, S_M be a set of ground normal rules, choice rules and constraints and E^+ and E^- be any sets of partial interpretations. $B \cup S_M$ must also be aggregate stratified.

Deciding whether $ILP^s_{LAS}(B, S_M, E^+, E^-)$ has a solution is in NP^{NP}.

Proof. A non-deterministic Turing Machine can have $|S_M|$ choices to make (corresponding to selecting each rule as part of the hypothesis). This hypothesis can then be verified in polynomial time using an NP oracle (as in lemma 10.8).

Such a Turing Machine would terminate answering yes if and only if the task is satisfiable (as there is a path through the Turing Machine which answers yes if and only if there is an hypothesis in S_M which is an inductive solution of the task).

Hence, deciding the existence of a solution for a general (ground) ILP_{LAS} task is in NP^{NP}.

□

Lemma 10.10. Let B be any ground program containing normal rules choice rules and constraints, S_M be a set of ground normal rules, choice rules and constraints and E^+ and E^- be any sets of partial interpretations.

Deciding whether $ILP_{LAS}(B, S_M, E^+, E^-)$ has a solution is $NP^{NP} - hard$.

Proof. We show this by reducing a known $NP^{NP} - complete$ problem (deciding the existence of an answer set for a ground disjunctive logic program) to an ILP_{LAS} task.

Take any ground disjunctive logic program P. We will define an ILP_{LAS} task $T(P)$ which has a solution if and only if P has an answer set.

Let *Atoms* be the set of atoms in P. Let P' be the program constructed by replacing each negative literal `not a` with the literal `not in_as(a)` (where *in_as* is a new predicate) and replacing each head $h_1 \vee \ldots \vee h_m$ with the counting aggregate $1\{h_1, \ldots, h_m\}m$ (empty heads are mapped to $1\{\}0$ - this is equivalent to $\bot$).

We define the learning task $T(P)$ as follows:

$B = P' \cup \{\ \texttt{:- a, not in\_as(a)}\ | a \in Atoms\}$
$S_M = \{in\_as(a) \mid a \in Atoms\}$
$E^+ = \{\langle \emptyset, \emptyset \rangle\}$
$E^- = \{\langle \{in\_as(a)\}, \{a\} \rangle | a \in Atoms\}$

This task has a solution if there exists an $H \subseteq S_M$ such that $B \cup H$ is satisfiable and no negative example is extended by any answer set of $B \cup H$.

$$\Leftrightarrow \exists H \subseteq S_M \text{ st } \exists A \in AS\left(\left\{ \texttt{1\{h}_1,\ldots,\texttt{h}_\texttt{m}\texttt{\}m :- b}_1,\ldots,\texttt{b}_\texttt{n} \;\middle|\; \begin{array}{c} \texttt{1\{h}_1,\ldots,\texttt{h}_\texttt{m}\texttt{\}m :- b}_1,\ldots,\texttt{b}_\texttt{n}, \\ \texttt{not in\_as(c}_1\texttt{)},\ldots,\texttt{ not in\_as(c}_\texttt{o}\texttt{)}. \\ \{in\_as(c_1),\ldots,in\_as(c_o)\} \cap H = \emptyset \end{array} \in P', \right\}\right)$$

such that $A \subseteq \{a \mid in\_as(a) \in H\}$ and no negative example is extended by any answer set of this program.

$$\Leftrightarrow \exists H \subseteq S_M \text{ st } \exists A \in AS\left(\left\{ \texttt{1\{h}_1,\ldots,\texttt{h}_\texttt{m}\texttt{\}m :- b}_1,\ldots,\texttt{b}_\texttt{n} \;\middle|\; \begin{array}{c} \texttt{1\{h}_1,\ldots,\texttt{h}_\texttt{m}\texttt{\}m :- b}_1,\ldots,\texttt{b}_\texttt{n}, \\ \texttt{not in\_as(c}_1\texttt{)},\ldots,\texttt{ not in\_as(c}_\texttt{o}\texttt{)}. \\ \{in\_as(c_1),\ldots,in\_as(c_o)\} \cap H = \emptyset \end{array} \in P', \right\}\right)$$

such that $A = \{a \mid in\_as(a) \in H\}$ and there is no strict subset of A which is also an answer set (the negative examples prevent this).

$\Leftrightarrow \exists H \subseteq S_M$ st $\{a \mid in\_as(a) \in H\}$ is a minimal model of

$$\left\{ \texttt{h}_1 \vee \ldots \vee \texttt{h}_\texttt{m}\texttt{\}m :- b}_1,\ldots,\texttt{b}_\texttt{n} \;\middle|\; \begin{array}{c} \texttt{1\{h}_1,\ldots,\texttt{h}_\texttt{m}\texttt{\}m :- b}_1,\ldots,\texttt{b}_\texttt{n}, \\ \texttt{not in\_as(c}_1\texttt{)},\ldots,\texttt{ not in\_as(c}_\texttt{o}\texttt{)}. \\ \{in\_as(c_1),\ldots,in\_as(c_o)\} \cap H = \emptyset \end{array} \in P', \right\}$$

$\Leftrightarrow \exists H \subseteq S_M$ st $\{a \mid in\_as(a) \in H\}$ is a minimal model of

$$\left\{ \mathtt{h_1} \vee \ldots \vee \mathtt{h_m}\}\mathtt{m}\,\mathtt{:-}\,\mathtt{b_1},\ldots,\mathtt{b_n} \;\middle|\; \begin{array}{c} \mathtt{1\{h_1,\ldots,h_m\}m\,:-\,b_1,\ldots,b_n,} \\ \mathtt{not\ c_1,\ldots,\ not\ c_o.} \\ \{in\_as(c_1),\ldots,in\_as(c_o)\} \cap H = \emptyset \end{array} \in P, \right\}$$

$\Leftrightarrow \exists A \subseteq \mathit{Atoms}$ st A is a minimal model of

$$\left\{ \mathtt{h_1} \vee \ldots \vee \mathtt{h_m}\}\mathtt{m}\,\mathtt{:-}\,\mathtt{b_1},\ldots,\mathtt{b_n} \;\middle|\; \begin{array}{c} \mathtt{1\{h_1,\ldots,h_m\}m\,:-\,b_1,\ldots,b_n,} \\ \mathtt{not\ c_1,\ldots,\ not\ c_o.} \\ \{c_1,\ldots,c_o\} \cap H = \emptyset \end{array} \in P, \right\}$$

$\Leftrightarrow \exists A \subseteq \mathit{Atoms}$ such that A is a minimal model of P^A

$\Leftrightarrow \exists A \subseteq \mathit{Atoms}$ such that A an answer set of P.

$\Leftrightarrow P$ is satisfiable.

Hence, deciding whether a disjunctive logic program can in general be mapped to the decision problem of checking the existence of solutions of a learning from answer sets task.

Therefore, deciding the existence of solutions of a ground ILP_{LAS} task is $NP^{NP} - hard$.

□

Theorem 10.11. Deciding the existence of ILP_{LOAS} and ILP_{LAS} tasks are both NP^{NP}-complete.

Proof. By lemma 10.10, deciding the existence of solutions for ILP_{LAS} is NP^{NP}-hard. Deciding the existence of solutions for ILP_{LAS} reduces to deciding the existence of solutions for ILP^s_{LAS} (trivially) and by lemma 10.9, deciding the existence of solutions for ILP^s_{LAS} is in NP^{NP}. Hence deciding the existence of solutions for ILP_{LAS} is NP^{NP}-complete.

By lemma 10.6, deciding the existence of solutions for an ILP_{LOAS} task polynomially reduces to deciding the existence of solutions for an ILP^s_{LAS} task; hence, deciding the existence of solutions for an ILP_{LOAS} task is in NP^{NP}. As deciding the existence of solutions of an ILP_{LAS} task is NP^{NP}-hard and ILP_{LAS} reduces trivially to an ILP_{LOAS} task (by lemma 10.7), ILP_{LOAS} is NP^{NP}-hard. Hence, deciding the existence of solutions for an ILP_{LOAS} task is also NP^{NP}-complete.

□

References

[1] F. Calimeri, W. Faber, M. Gebser, G. Ianni, R. Kaminski, T. Krennwallner, N. Leone, F. Ricca, T. Schaub, Asp-core-2 input language format (2013).

[2] M. Law, A. Russo, K. Broda, Simplified reduct for choice rules in asp., Tech. Rep. DTR2015-2, Imperial College of Science, Technology and Medicine, Department of Computing (2015).

[3] M. Gebser, R. Kaminski, B. Kaufmann, T. Schaub, Answer Set Solving in Practice, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool Publishers, 2012.

[4] M. Law, A. Russo, K. Broda, Inductive learning of answer set programs, in: Logics in Artificial Intelligence (JELIA 2014), Springer, 2014.

[5] W. Faber, G. Pfeifer, N. Leone, Semantics and complexity of recursive aggregates in answer set programming, Artificial Intelligence 175 (1) (2011) 278–298.

Appendix

A Proofs omitted from the report

Lemma 5.5. Let l be a constant, t_1 and t_2 be two distinct ground terms and *head* be an atom.

Let R be the rule $\texttt{head :- b}_1, \ldots, \texttt{b}_n, \#\texttt{sum}\{\texttt{w(W, l, A, t}_1\texttt{)} = \texttt{W}, \texttt{w(W, l, A, t}_2\texttt{)} = -\texttt{W}\} < 0$ and F be a set of (ground) facts of the predicate $w/4$ and (where *head* has a different predicate name to w)

For $i \in \{1, 2\}$, let $\mathcal{S}_i = (\sum_{w(weight,l,args,t_i) \in F} weight)$

1. If $\mathcal{S}_1 \geq \mathcal{S}_2$ then $M(F \cup R) = M(F)$
2. If $\mathcal{S}_1 < \mathcal{S}_2$ then $M(F \cup R) = F \cup \{\texttt{head :- b}_1, \ldots, \texttt{b}_n\})$

Proof. Follows from corollary 4.7.

Let $s_1 = w(W, l, A, t_1), s_2 = w(W, l, A, t_2), w_1 = W, w_2 = -W$

1. Assume $\mathcal{S}_1 \geq \mathcal{S}_2$
   $\Rightarrow \sum_{w(weight,l,args,t_1) \in F}(weight) \geq \sum_{w(weight,l,args,t_2) \in F}(weight)$
   $\Rightarrow \sum_{w(weight,l,args,t_1) \in F}(weight) - \sum_{w(weight,l,args,t_2) \in F}(weight) \geq 0$
   $\Rightarrow \sum_{w(weight,l,args,t_1) \in F}(weight) + \sum_{w(weight,l,args,t_2) \in F}(-weight) \geq 0$
   $\Rightarrow \sum_{s \in F, \exists \theta st\ s = w(W,l,A,t_1)\theta}(W\theta) + \sum_{s \in F, \exists \theta st\ s = w(W,l,A,t_2)\theta}(-W\theta) \geq 0$
   $\Rightarrow \sum_{s \in F, \exists \theta st\ s = s_1\theta}(w_1\theta) + \sum_{s \in F, \exists \theta st\ s = s_2\theta}(w_2\theta) \geq 0$
   $\Rightarrow \sum_{s \in F, \exists \theta \exists i \in \{1,2\} st\ s = s_i\theta}(w_i\theta) \geq 0$
   $\Rightarrow$ (by corollary 4.7) $AS(F \cup R) = AS(F)$
2. Assume $\mathcal{S}_1 < \mathcal{S}_2$
   $\Rightarrow \sum_{w(weight,l,args,t_1) \in F}(weight) < \sum_{w(weight,l,args,t_2) \in F}(weight)$
   $\Rightarrow \sum_{w(weight,l,args,t_1) \in F}(weight) - \sum_{w(weight,l,args,t_2) \in F}(weight) < 0$
   $\Rightarrow \sum_{w(weight,l,args,t_1) \in F}(weight) + \sum_{w(weight,l,args,t_2) \in F}(-weight) < 0$
   $\Rightarrow \sum_{s \in F, \exists \theta st\ s = w(W,l,A,t_1)\theta}(W\theta) + \sum_{s \in F, \exists \theta st\ s = w(W,l,A,t_2)\theta}(-W\theta) < 0$
   $\Rightarrow \sum_{s \in F, \exists \theta st\ s = s_1\theta}(w_1\theta) + \sum_{s \in F, \exists \theta st\ s = s_2\theta}(w_2\theta) < 0$
   $\Rightarrow \sum_{s \in F, \exists \theta \exists i \in \{1,2\} st\ s = s_i\theta}(w_i\theta) < 0$
   $\Rightarrow$ (by corollary 4.7) $AS(F \cup R) = AS(F \cup \{head \texttt{ :- } b_1, \ldots, b_n\})$

□