This document provides the proofs which were omitted from the paper *Inductive Learning of Answer Set Programs*. In the first section, we recall the necessary definitions from the paper. In section 2 we introduce some extra notation which serves only to simplify the proofs. In section 3 we give some lemmas necessary for the proofs; and finally, in section 4 we give the proofs.

1 Definitions

Definition 1.1 corresponds to definition 4 from the paper.

Definition 1.1. A *Learning from Answer Sets* task is a tuple $T = \langle B, S_M, E^+, E^- \rangle$ where B be is the background knowledge, S_M the search space defined by a language bias M, E^+ and E^- are sets of partial interpretations called, respectively, the positive and negative examples. A hypothesis $H \in ILP_{LAS}(T)$, the set of *inductive solutions* of T if and only if:

1. $H \subseteq S_M$
2. $\forall e^+ \in E^+ \; \exists A \in AS(B \cup H)$ such that A extends e^+
3. $\forall e^- \in E^- \; \nexists A \in AS(B \cup H)$ such that A extends e^-

We write $ILP^n_{LAS}(T)$ to mean the set of all inductive solutions of length n.

Definition 1.2 corresponds to definition 6 from the paper.

Definition 1.2. Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} task. An hypothesis $H \in positive\_solutions(T)$, called the set of *positive inductive solutions* of T, if and only if $H \subseteq S_M$ and $\forall e^+ \in E^+ \; \exists A \in AS(B \cup H)$ such that A extends e^+.

Definition 1.3 corresponds to definition 7 from the paper.

Definition 1.3. Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} task. An hypothesis $H \in violating\_solutions(T)$, called the set of *violating inductive solutions* of T, if and only if $H \in positive\_solutions(H)$ and $\exists e^- \in E^- \; \exists A \in AS(B \cup H)$ such that A extends e^-.

We will write $positive\_solutions^n(T)$ and $violating\_solutions^n(T)$ to denote the positive and violating solutions of length n.

Definition 1.4 corresponds to definition 8 from the paper.

Definition 1.4. Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} learning task and $n \in \mathbb{N}$. Let R_{id} be a unique identifier for each rule $R \in S_M$ and let e^+_{id} be a unique identifier for each positive example $e^+ \in E^+$. The learning task T is represented as the ASP *task program* $T^n_{meta} = meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(E^-) \cup meta(Aux, n)$ where each of these five "meta" components are as follows:

1. $meta(B)$ is generated from B by replacing every atom A with the atom $e(A, X)$, and by adding the condition $ex(X)$ to the body of each rule.
2. $meta(S_M)$ is generated from S_M by replacing every atom A with the atom $e(A, X)$, and by adding the two conditions $active(R_{id})$ and $ex(X)$ to the body of the rule R that matches the correct rule identifier R_{id}.

3. $meta(E^+)$ includes for every $e^+ = \langle\{li_1,\ldots,li_h\},\{le_1,\ldots,le_k\}\rangle \in E^+$ the rules
   - $ex(ex^+_{id})$
   - $\leftarrow$ not $example\_covered(ex^+_{id})$
   - $example\_covered(e^+_{id}) \leftarrow e(li_1, ex^+_{id}), \ldots, e(li_h, ex^+_{id}),$ not $e(le_1, ex^+_{id}), \ldots,$ not $e(le_k, ex^+_{id})$

4. $meta(E^-)$ includes for every $e^- = \langle\{li_1,\ldots,li_h\},\{le_1,\ldots,le_k\}\rangle \in E^-$ the rule
   - $violating \leftarrow e(li_1, negative), \ldots, e(li_h, negative),$ not $e(le_1, negative), \ldots,$ not $e(le_k, negative)$

5. $meta(Aux, n)$ includes the ground facts $length(R_{id}, |R|)$ for every rule $R \in S_M$ and the rule n #sum$\{active(R) = X : length(R, X)\}n$ to impose that the total length of the (active) hypothesis has to be n.

Definition 1.5 corresponds to definition 9 from the paper.

Definition 1.5. Let hypothesis $H = \{R_1, \ldots, R_h\}$. We denote with $constraint(H)$ the rule $\leftarrow active(R_{id1}), \ldots, active(R_{idh})$, where R_{id1},... R_{idh} are the unique identifiers of rules $R_1, \ldots, R_h$ in H.

For any set of active ids A, $meta^{-1}(A) = \{R \in S_M : active(R_{id}) \in A\}$ ($meta^{-1}$ converts the Answer Sets of T^n_{meta} back to hypotheses).

2 Extra notation

This section gives some definitions which weren't in the paper. The only purpose of these definitions is to give some notation which simplifies the proofs.

Definition 2.1. Given a rule R and a constant c, we write $e(R, c)$ to denote the rule constructed by replacing every atom A in R with $e(A, c)$.

For any ASP program P and constant $const$ we will write $e(P, const)$ to mean the program constructed by replacing every atom $A \in P$ by $e(A, const)$. We will use the same notation for sets of literals/partial interpretations, so for a set S: $e(S, const) = \{e(A, const) : A \in S\}$.

Definition 2.2. 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 2.3. Given a program P and a positive example $e^+ = \langle E^{inc}, E^{exc}\rangle$ the expansion of P wrt e^+ is written $e^+[P]$ and constructed as follows:

$$append(e(B \cup H, e^+_{id}), ex(e^+_{id})) \cup \{ex(ex^+_{id}). \quad example\_covered(ex^+_{id}) \leftarrow \bigwedge_{lit \in ex^+_{inc}} e(lit, ex^+_{id}) \wedge \bigwedge_{lit \in ex^+_{exc}} \text{not } e(lit, ex^+_{id}).$$
$$\leftarrow \text{not } example\_covered(ex^+_{id}).\}$$

Definition 2.4. Given a program P and the set of all negative examples E^-

$$negative[P, E^-] = \{\leftarrow \text{not } violating. \quad ex(negative).\} \cup append(e(B \cup H, negative), ex(negative)) \cup \bigcup_{e^- \in E^-} \{violating \leftarrow \bigwedge_{lit \in ex^-_{inc}} e(lit, negative) \wedge \bigwedge_{lit \in ex^-_{exc}} \text{not } e(lit, negative)\}.$$

Definition 2.5. For any ILP_{LAS} task T and hypothesis $H \subseteq S_M$:

$T_{meta}[H] = meta(B) \cup meta(S_M) \cup meta(E^+) \cup meta(E^-) \cup \{active(R_{id}) : R \in H\}$.

(This is T^n_{meta} without $meta(Aux, n)$ in addition to one fact $active(R_{id})$ for each rule $R \in H$)

3 Lemmas

Lemma 3.1. For any ASP program P, $AS(ground(P)) = AS(P)$.

Lemma 3.2. For any ASP program P, such that P contains no rule with the predicate *active* in the head, and any sum rule S: n #sum $\{active(r_1) = w_1, ..., active(r_m) = w_m\}n$ (where the r_i's are constants and the w_i's are integers).

For any subset X of $[1, m]$ st $n = \sum_{i \in X} w_i$, then $AS(P \cup \{active(r_i) : i \in X\}) = \{A \in AS(P \cup S) : A \cap \{active(1), ..., active(m)\} = X\}$.

Corollary 3.3. For any hypothesis $H \subseteq S_M$ st $|H| = n$:

$\exists A \in AS(T^n_{meta})$ st $H = meta^{-1}(A) \Leftrightarrow T_{meta}[H]$ is satisfiable.

Lemma 3.4. Let P be any ground ASP program and C be any constraint $\leftarrow b_1 \wedge ... \wedge b_n \wedge \text{not } c_1 \wedge ... \wedge \text{not } c_m$,

$AS(P \cup C) = \{A \in AS(P) : (\exists i \in [1, n] \text{ st } b_i \notin A) \vee (\exists i \in [1, m] \text{ st } c_i \in A)\}$.

Lemma 3.5. For any ASP program P: $AS(e(P, const)) = e(AS(P), const)$.

Lemma 3.6. For any program $P \cup Q$ in which the atom a does not occur:

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

Lemma 3.7. 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.

$\exists A \in AS(P)$ st A extends E iff $P \cup \{a \leftarrow \bigwedge_{lit \in E^{inc}} lit \wedge \bigwedge_{lit \in E^{exc}} \text{not } lit. \quad \leftarrow a.\}$ is satisfiable.

Lemma 3.8. For any ILP_{LAS} task $T = \langle B, S_M, E^+, E^- \rangle$:

$H \in positive\_solutions^n(T)$ iff $|H| = n$ and $H \subseteq S_M$ and $\bigcup_{e^+ \in E^+} [e^+[B \cup H]]$ is satisfiable.

Proof. Assume $H \in positive\_solutions^n(T)$

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\forall e^+ \in E^+ : \exists A \in AS(B \cup H)$ st A extends e^+ (by definition).

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\forall e^+ \in E^+ : \exists A \in AS(B \cup H)$ st $e(A, e^+_{id})$ extends $e(e^+, e^+_{id})$.

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\forall e^+ \in E^+ : \exists A \in AS(e(B \cup H, e^+_{id}))$ st A extends $e(e^+, e^+_{id})$ by lemma 3.5.

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\forall e^+ \in E^+ : e(B \cup H, e^+_{id}) \cup \{\leftarrow \text{not } example\_covered(e^+_{id}).$
$example\_covered(e^+_{id}) \leftarrow \bigwedge_{lit \in e^+_{inc}} e(lit, e^+_{id}) \wedge \bigwedge_{lit \in e^+_{exc}} \text{not } e(lit, e^+_{id}).\}$ is satisfiable by lemma 3.7.

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\forall e^+ \in E^+ : append(e(B \cup H, e^+_{id}), ex(e^+_{id})) \cup \{ex(e^+_{id}). \quad \leftarrow \text{not } example\_covered(e^+_{id}).$ $example\_covered(e^+_{id}) \leftarrow \bigwedge_{lit \in e^+_{inc}} e(lit, e^+_{id}) \wedge \bigwedge_{lit \in e^+_{exc}} \text{not } e(lit, e^+_{id}).\}$ is satisfiable by lemma 3.6 (used once for each $e^+ \in E^+$).

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\forall e^+ \in E^+ : e^+[B \cup H]$ is satisfiable.

$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\bigcup_{e^+ \in E^+} e^+[B \cup H]$ is satisfiable (the individual programs have no atoms in common as every atom in each contains the relevant constant e^+_{id}).

□

Lemma 3.9. For any program P and set of examples E^-:

$\exists e^- \in E^-$ st $\exists A \in AS(P)$ st A extends e^- iff $negative[P, E^-]$ is satisfiable.

Proof. Assume $\exists e^- \in E^-$ st $\exists A \in AS(P)$ st A extends e^-

$\Leftrightarrow \exists e^- \in E^-$ st $\exists A \in AS(P)$ st $e(A, negative)$ extends $e(e^-, negative)$.

$\Leftrightarrow \exists e^- \in E^-$ st $\exists A \in AS(e(P, negative))$ st A extends $e(e^-, negative)$ by lemma 3.5.

$\Leftrightarrow \exists e^- \in E^-$ st $e(P, negative) \cup \{\leftarrow \text{not } violating.$ $violating \leftarrow \bigwedge_{lit \in e(e^-_{inc}, negative)} lit \wedge \bigwedge_{lit \in e(e^-_{exc}, negative)} \text{not } lit.\}$ is satisfiable by lemma 3.7.

$e(P, negative) \cup \bigcup_{e^- \in E^-} \{\leftarrow \text{not } violating. \quad violating \leftarrow \bigwedge_{lit \in e(e^-_{inc}, negative)} lit \wedge \bigwedge_{lit \in e(e^-_{exc}, negative)} \text{not } lit.\}$ is satisfiable (as violating already occurs in every Answer Set, so adding more rules with violating at the head will make no difference).

$\Leftrightarrow negative[P, E^-]$ is satisfiable by lemma 3.6. □

4 Proofs

Theorem 4.1 corresponds to Theorem 1 from the paper.

Theorem 4.1. *Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} learning task.*

Then $ILP_{LAS}(T) = positive\_solutions(T) \backslash violating\_solutions(T)$

Proof.

$$
\begin{aligned}
H \in ILP_{LAS}(T) \quad & \Leftrightarrow H \subseteq \quad S_M \wedge \forall e^+ \in E^+ : \; \exists A \in AS(B \cup H) \text{ st } A \text{ extends } e^+ \\
& \qquad\qquad \wedge \forall e^- \in E^- : \nexists A \in AS(B \cup H) \text{ st } A \text{ extends } e^+ \\
& \Leftrightarrow H \subseteq \quad S_M \wedge \forall e^+ \in E^+ : \exists A \in AS(B \cup H) \text{ st } A \text{ extends } e^+ \\
& \qquad\qquad \wedge \nexists e^- \in E^- \text{ st } \exists A \in AS(B \cup H) \text{ st } A \text{ extends } e^+ \\
& \Leftrightarrow H \in \quad positive\_solutions(T) \\
& \qquad\qquad \wedge \nexists e^- \in E^- \text{ st } \exists A \in AS(B \cup H) \text{ st } A \text{ extends } e^+ \\
& \Leftrightarrow H \in \quad positive\_solutions(T) \wedge H \notin violating\_solutions(T)
\end{aligned}
$$

□

Proposition 4.2 corresponds to proposition 1 from the paper.

Proposition 4.2. Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} task and $n \in \mathcal{N}$.
Then $H \in positive\_solutions^n(T)$ if and only if $\exists A \in AS(T^n_{meta})$ such that $H = meta^{-1}(A)$.

Proof. Assume $H \in positive\_solutions^n(T)$
$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\bigcup_{e^+ \in E^+} [e^+[B \cup H]]$ is satisfiable by lemma 3.8.
$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\bigcup_{e^+ \in E^+} [e^+[B \cup H]] \cup append(e(B \cup H, negative), ex(negative))$ is satisfiable (as none of the bodies of these new rules can be true - $ex(negative)$ does not appear at the head of any rule).
$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $\bigcup_{e^+ \in E^+} [e^+[B \cup H]] \cup append(e(B \cup H, negative), ex(negative)) \cup \bigcup_{e^- \in E^-} \{violating \leftarrow \bigwedge_{lit \in e^-_{inc}} e(lit, negative) \wedge \bigwedge_{lit \in e^-_{exc}} \text{not } e(lit, negative).\}$ is satisfiable (as $violating$ does not appear in the body of any other rule (can seen by splitting the program on every literal other than $violating$)).
$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $ground(T_{meta}[H])$ is satisfiable by lemma 3.6 (we use lemma 3.6 once for each $R \in H$ to add $active(R_{id})$ as a fact, and append it to every rule of the form $e(R, c)$ for some constant c).
$\Leftrightarrow H \subseteq S_M$ and $|H| = n$ and $T_{meta}[H]$ is satisfiable by lemma 3.1.
$\Leftrightarrow \exists A \in AS(T^n_{meta})$ st $H = meta^{-1}(A)$ by corollary 3.3.

□

Proposition 4.3 corresponds to proposition 2 from the paper.

Proposition 4.3. Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} task and $n \in \mathcal{N}$.
Let P be the ASP program $T^n_{meta} \cup \{\leftarrow not\ violating;\ \ ex(negative)\}$.
Then $H \in violating\_solutions^n(T)$ if and only if $\exists A \in AS(P)$ such that $H = meta^{-1}(A)$.

Proof. Assume $H \in violating\_solutions^n(T)$
$\Leftrightarrow H \in positive\_solutions^n(T)$ and $\exists e^- \in E^-$ st $\exists A \in AS(B \cup H)$ st A extends E.
$\Leftrightarrow |H| = n$ and $H \subseteq S_M$ and $\bigcup_{e^+ \in E^+} (e^+[B \cup H])$ is satisfiable and $\exists e^- \in E^-$ st $\exists A \in AS(B \cup H)$ st A extends E lemma 3.8.
$\Leftrightarrow |H| = n$ and $H \subseteq S_M$ and $\bigcup_{e^+ \in E^+} (e^+[B \cup H])$ is satisfiable and $negative[B \cup H]$ is satisfiable lemma 3.9.
$\Leftrightarrow |H| = n$ and $H \subseteq S_M$ and $\bigcup_{e^+ \in E^+} (e^+[B \cup H]) \cup negative[B \cup H]$ is satisfiable (as the two programs share no atoms).
$\Leftrightarrow ground(T_{meta}[H]) \cup \{\leftarrow \text{not } violating.\quad ex(negative).\}$ is satisfiable by lemma 3.6 (we use lemma 3.6 once for each $R \in H$ to add $active(R_{id})$ as a fact and also append it to every rule of the form $e(R, c)$ for some constant c).
$\Leftrightarrow T_{meta}[H] \cup \{\leftarrow \text{not } violating.\quad ex(negative).\}$ is satisfiable (by lemma 3.1).
$\Leftrightarrow \exists A \in AS(T^n_{meta} \cup \{\leftarrow \text{not } violating.\quad ex(negative).\})$ st $H = meta^{-1}(A)$ (by corollary 3.3).

□

Proposition 4.4 corresponds to proposition 3 from the paper.

Proposition 4.4. Let $T = \langle B, S_M, E^+, E^- \rangle$ be an ILP_{LAS} task and $n \in \mathcal{N}$.
Let $P = T^n_{meta} \cup \{constraint(V) : V \in violating\_solutions^n(T)\}$.
Then a hypothesis $H \in ILP^n_{LAS}(T)$ if and only if $\exists A \in AS(P)$ such that $H = meta^{-1}(A)$.

Proof. Assume $H \in ILP^n_{LAS}(T)$ (then $n = |H|$)

$\Leftrightarrow H \in positive\_solutions^n(T)$ and $H \notin violating\_solutions^n(T)$ by proposition 4.1

$\Leftrightarrow \exists A \in AS(T^n_{meta})$ st $H = meta^{-1}(A)$ and $H \notin violating\_solutions^n(T)$ by proposition 4.2

$\Leftrightarrow \exists A \in AS(T^n_{meta})$ st $H = meta^{-1}(A)$ and H is does not satisfy the body of any constraint $\{constraint(V) : V \in violating\_solutions^n(T)\}$.

$\Leftrightarrow \exists A \in AS(P)$ st $H = meta^{-1}(A)$ by lemma 3.4.

□