Representing and Learning Grammars in Answer Set Programming: Proofs

Mark Law
Imperial College London, UK
mark.law09@imperial.ac.uk

Alessandra Russo
Imperial College London, UK
a.russo@imperial.ac.uk

Elisa Bertino
Purdue University, USA
bertino@purdue.edu

Krysia Broda
Imperial College London, UK
k.broda@imperial.ac.uk

Jorge Lobo
ICREA - Universitat Pompeo Fabra
jorge.lobo@upf.edu

Theorem 2. *For any fragment $\mathcal{F}$ of ASP that contains constraints and negation as failure, $\mathcal{F}$ BAM reduces to $\mathcal{F}$ BAS.*

Proof. Let G be an ASG in $ASG^{\mathcal{F}}$ and $\mathtt{s}$ be the string $\mathtt{s_1 \ldots s_{|s|}}$. To prove the theorem, we show that deciding whether $\mathtt{s} \in \mathcal{L}^d(G)$ reduces to $\mathcal{F}$ BAS. Let G' be the grammar constructed by extending G in the following way:

- Replace G_S with a new start terminal $\mathtt{start}$, and adding a single production rule $\mathtt{start} \rightarrow \mathtt{start}'\{\mathtt{:\text{-}\ not\ yields(1, |s|)@1.}\}$ to G_{PR} (where $\mathtt{start}'$ is the original start node of G).
- For each production rule $n \rightarrow n_1 \ldots n_k P$ in G_{PR}, add the following rules to P:
  - For each $X \in [1, |s|]$, the fact $\mathtt{yields(X, X, 0)}$.
  - For each $i \in [1, k]$ such that $n_i \in G_T$, for each $X, Y \in [1, |s|]$ such that $s_Y = n_i$, the rule $\mathtt{yields(X, Y, i)\,\text{:-}\,yields(X, Y-1, i-1)}$.
  - For each $i \in [1, k]$ such that $n_i \in G_N$, for each $X, Y, Z \in [1, |s|]$, the rule $\mathtt{yields(X, Z, i)\,\text{:-}\,yields(X, Y, i-1), yields(Y, Z)@i}$.
  - The rule $\mathtt{yields(X, Y)\,\text{:-}\,yields(X, Y, k)}$.

The extra ASP rules in the grammar restrict G so that the only possible string in $\mathcal{L}^d(G')$ is $\mathtt{s}$. This means that $\mathcal{L}(G') \neq \emptyset$ iff $\mathtt{s} \in \mathcal{L}^d(G)$. Hence $\mathcal{F}$ BAM reduces to $\mathcal{F}$ BAS. □

We say that an annotated ASP program is *groundly annotated* if all its annotations are ground.

Lemma 1. *Deciding the satisfiability of a groundly annotated ASP program reduces to deciding satisfiability of an unannotated ASP program using the same fragment of ASP.*

Proof. Let P be an annotated program. Let P' be the program constructed by replacing each annotated atom $\mathtt{p(t_1, \ldots, t_n)@[a_1, \ldots, a_m]}$ with the atom $\mathtt{p(t_1, \ldots, t_n, annotations, a_1, \ldots, a_m)}$, where $\mathtt{annotations}$ is a new constant symbol (required to differentiate $\mathtt{p(1, 1)@[1]}$ from $\mathtt{p(1)@[1, 1]}$, which will be replaced by $\mathtt{p(1, 1, annotations, 1)}$ and $\mathtt{p(1, annotations, 1, 1)}$, respectively).

P' is isomorphic to P, and is therefore satisfiable iff P is satisfiable. P' also uses the same fragment of ASP. Hence, deciding the satisfiability of an groundly annotated ASP program reduces to deciding satisfiability of an unannotated ASP program using the same fragment of ASP (note that in the propositional case P' will be ground, and so it is isomorphic to a propositional program). □

Theorem 4. *Propositional unstratified BAS is in NP.*

Proof. Let G be a propositional unstratified ASG. Let $\mathtt{max\_k}$ be the number of nodes in the body of the longest production rule in G_{PR}. Let Π be the following ASP program:

- $\mathtt{node(G_S)@[]}$.
- For each $D \in [1, .., d]$, for each $X_1, X_2, \ldots, X_D \in [1, \mathtt{max\_k}]$:
  - If $D = \mathtt{max\_k}$ for each production PR with a non-empty body, the rule:
    $\mathtt{:\text{-}\,pr(PR_{ID})@[X_1, X_2, \ldots, X_D]}$.
  - For each production rule $PR = \mathtt{n} \rightarrow \mathtt{n_1 \ldots n_k}\ P$, for each $i \in [1, k]$:
    - ∗ The rule:
      $\mathtt{node(n_i)@[X_1, X_2, \ldots, X_{D-1}, i]\,\text{:-}}$
      $\mathtt{pr(PR_{ID})@[X_1, X_2, \ldots, X_{D-1}]}$.
    - ∗ For each rule $\mathtt{h@a_0}$:- $b_1@a_1, \ldots, b_i@a_i$, not $b_{i+1}@a_{i+1}, \ldots,$ not $b_j@a_j$ in P:
      $\mathtt{h@[X_1, X_2, \ldots, X_D, a_0]\,\text{:-}}$
      $\mathtt{b_1@[X_1, X_2, \ldots, X_D, a_1]}$,
      $\ldots$,
      $\mathtt{b_i@[X_1, X_2, \ldots, X_D, a_i]}$,
      $\mathtt{not\ b_{i+1}@[X_1, X_2, \ldots, X_D, a_{i+1}]}$,
      $\ldots$,
      $\mathtt{not\ b_j@[X_1, X_2, \ldots, X_D, a_j]}$.
  - Let $PR^1, \ldots, PR^m$ be the set of production rules in G_{PR} for n.
    For each node $n \in G_N$, for each $i \in [1, m]$, the rule:

$\mathtt{pr(PR^{i}_{ID})@[X_1, X_2, \ldots, X_D]:-}$
$\quad\mathtt{node(n)@[X_1, X_2, \ldots, X_D]}$
$\quad\mathtt{not\ pr(PR^{1}_{ID})@[X_1, X_2, \ldots, X_D],}$
$\quad\ldots$
$\quad\mathtt{not\ pr(PR^{i-1}_{ID})@[X_1, X_2, \ldots, X_D],}$
$\quad\mathtt{not\ pr(PR^{i+1}_{ID})@[X_1, X_2, \ldots, X_D],}$
$\quad\ldots$
$\quad\mathtt{not\ pr(PR^{m}_{ID})@[X_1, X_2, \ldots, X_D].}$

For each node $n \in G_N$, the rule:
$\mathtt{:-node(n)@[X_1, X_2, \ldots, X_D]}$
$\quad\mathtt{not\ pr(PR^{1}_{ID})@[X_1, X_2, \ldots, X_D],}$
$\quad\ldots$
$\quad\mathtt{not\ pr(PR^{m}_{ID})@[X_1, X_2, \ldots, X_D].}$

Π is satisfiable iff $\mathcal{L}^d(G) \neq \emptyset$. Hence, deciding whether $\mathcal{L}^d(G) \neq \emptyset$ reduces to the NP problem of deciding the satisfiability of a propositional ASP program consisting of normal rules and constraints. Hence, propositional unstratified BAS is a member of NP. □

Theorem 7. *First order stratified BAS is in EXP.*

Proof. Let G be a first order stratified ASG.

Let $\mathtt{max\_k}$ be the number of nodes in the body of the longest production rule in G_{PR}. Let $pt\_size = \Sigma_{i=0}^{d} max\_k^i$. Note that this is polynomial in the size of G (the value of d is constant).

A parse tree PT is represented as an atom $pt(pr^1_{id}, \ldots, pr^{pt\_size}_{id})$, where for each node $n \in PT$, $pr^{f(trace(n))}_{id} = rule(n)_{id}$, where $f([t_1, \ldots, t_m]) = 1 + \Sigma_{i=1}^{m} max\_k^{m-i} t_i$. For any trace t not present in the parse tree $pr^{f(t)}_{id} = 0$.

Let for each $D \in [0, d]$, let $traces(D)$ be the set of lists $\{[t_1, \ldots, t_D] | \forall i \in [1, D], t_i \in [1, \mathtt{max\_k}]\}$

Let C be the following set of rules:

- For each $m \in [0, d-1]$, each trace $T = [t_1, \ldots, t_m]$ such that $\forall i \in [1, m] : t_i \in [1, \mathtt{max\_k}]$, and each $j \in [1, \mathtt{max\_k}]$, the rule:
  $\mathtt{vio(X_1, \ldots, X_{pt\_size}):-}$
  $\quad\mathtt{p(X_1, \ldots, X_{pt\_size}),}$
  $\quad\mathtt{X_{f(T)} = 0,}$
  $\quad\mathtt{X_{f(T++[j])} \neq 0.}$
- For each $m \in [0, d-1]$, each production rule $n \to n_1 \ldots n_k\ P \in G_{PR}$ with id pr_{id}, and each $T = [t_1, \ldots, t_m]$ such that $\forall i \in [1, m] : t_i \in [1, \mathtt{max\_k}]$:
  - For each $j \in [k+1, \mathtt{max\_k}]$, the rule:
    $\mathtt{vio(X_1, \ldots, X_{pt\_size}):-}$
    $\quad\mathtt{p(X_1, \ldots, X_{pt\_size}),}$
    $\quad\mathtt{X_{f(T)} = pr_{id},}$
    $\quad\mathtt{X_{f(T++[j])} \neq 0.}$
  - For each $i \in [1, k]$, and production rule $n' \to n'_1 \ldots n'_{k'}\ P' \in G_{PR}$ with id pr'_{id} such that $n' \neq n_i$, the rule:
    $\mathtt{vio(X_1, \ldots, X_{pt\_size}):-}$
    $\quad\mathtt{p(X_1, \ldots, X_{pt\_size}),}$
    $\quad\mathtt{X_{f(T)} = pr_{id},}$
    $\quad\mathtt{X_{f(T++[i])} = pr'_{id}.}$

Consider the program $Pi \cup C \cup \{\mathtt{p(X_1, \ldots, X_{pt\_size}):-index(X_1), \ldots, index(X_{pt\_size}).}\} \cup \{\mathtt{index(i).} \mid i \in [0, \mathtt{max\_k}]\}$. The program has a single answer set which contains $\mathtt{vio(X_1, \ldots, X_{pt\_size})}$ for each $\mathtt{X_1, \ldots, X_{pt\_size}}$ iff the corresponding parse tree is not a valid parse tree for G_{CFG}.

Let Π^2 be the program consisting of Π and the following extra rules:

- For each production rule $n \to n_1 \ldots n_k\ P \in G_{PR}$ with id pr_{id}, each $m \in [1, d]$, each $T = [t_1, \ldots, t_m]$ such that $\forall i \in [1, m] : t_i \in [1, \mathtt{max\_k}]$, and each rule $R \in P$: the rule constructed by appending $p(X_1, \ldots, X_{pt\_size})$ to the body of $R@[X_1, \ldots, X_{pt\_size}] + +T$, and replacing the head of any constraints with $\mathtt{vio(X_1, \ldots, X_{pt\_size})}$.
- The rule:
  $\mathtt{non\_empty:-}$
  $\quad\mathtt{p(X_1, \ldots, X_{pt\_size}),}$
  $\quad\mathtt{not\ vio(X_1, \ldots, X_{pt\_size}).}$

The resulting program is stratified, and bravely entails $\mathtt{non\_empty}$ iff $\mathcal{L}^d(G)$ is non-empty – there must be at least one parse tree that is both valid and whose resulting ASP program is satisfiable. Thus, as the program is polynomial in the size of G, we have shown a polynomial reduction from first order BAS to an EXP-complete problem. Hence, stratified first order BAS is a member of EXP. □

Theorem 8. *First order unstratified BAS in NEXP.*

Proof. We prove the theorem by showing that an ASG G can be mapped to an ASP program P which is satisfiable iff $\mathcal{L}^d(G) \neq \emptyset$.

Let G be a first order unstratified ASG. Let Π be the following ASP program:

- $\mathtt{node(G_S)@[].}$
- For each $D \in [1, .., d]$, for each $X_1, X_2, \ldots, X_D \in [1, \mathtt{max\_k}]$:
  - If $D = \mathtt{max\_k}$ for each production PR with a non-empty body, the rule:
    $\mathtt{:-pr(PR_{ID})@[X_1, X_2, \ldots, X_D].}$
  - For each production rule $PR = \mathtt{n} \to \mathtt{n_1 \ldots n_k}\ P$, for each $i \in [1, k]$:
    - The rule:
      $\mathtt{node(n_i)@[X_1, X_2, \ldots X_{D-1}, i]:-}$
      $\quad\mathtt{pr(PR_{ID})@[X_1, X_2, \ldots X_{D-1}].}$
    - For each rule $\mathtt{h@a_0}$:- $b_1@a_1, \ldots, b_i@a_1$, $\mathtt{not}$ $b_{i+1}@a_{i+1}, \ldots,$ $\mathtt{not}$ $b_j@a_j$ in P:
      $\mathtt{h@[X_1, X_2, \ldots, X_D, a_0]:-}$
      $\quad\mathtt{b_1@[X_1, X_2, \ldots, X_D, a_1],}$
      $\quad\ldots,$
      $\quad\mathtt{b_i@[X_1, X_2, \ldots, X_D, a_i],}$
      $\quad\mathtt{not\ b_{i+1}@[X_1, X_2, \ldots, X_D, a_{i+1}],}$
      $\quad\ldots,$
      $\quad\mathtt{not\ b_j@[X_1, X_2, \ldots, X_D, a_j].}$
  - Let $PR^1, \ldots, PR^m$ be the set of production rules in G_{PR} for n.
    For each node $n \in G_N$, for each $i \in [1, m]$, the rule:

$\texttt{pr(PR}^{\texttt{i}}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]:-}$
$\quad\texttt{node(n)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]}$
$\quad\texttt{not pr(PR}^{\texttt{1}}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]},$
$\quad\ldots$
$\quad\texttt{not pr(PR}^{\texttt{i}-1}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]},$
$\quad\texttt{not pr(PR}^{\texttt{i}+1}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]},$
$\quad\ldots$
$\quad\texttt{not pr(PR}^{\texttt{m}}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]}.$

For each node $n \in G_N$, the rule:

$\texttt{:-node(n)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]}$
$\quad\texttt{not pr(PR}^{\texttt{1}}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]},$
$\quad\ldots$
$\quad\texttt{not pr(PR}^{\texttt{m}}_{\texttt{ID}}\texttt{)@[X}_1,\texttt{X}_2,\ldots,\texttt{X}_\texttt{D}\texttt{]}.$

Π is satisfiable iff $\mathcal{L}^d(G) \neq \emptyset$. Hence, deciding whether $\mathcal{L}^d(G) \neq \emptyset$ reduces to the NEXP problem of deciding the satisfiability of a first order ASP program consisting of normal rules and constraints. Hence, first order unstratified BAS is a member of NEXP. □

Theorem 9. *Horn BV is DP-hard.*

Proof. Let D be a decision problem in DP. There is a pair of decision problems D_1 and D_2 such that D_1 is in NP and D_2 is in $coNP$. There is a mapping from D_1 to deciding whether a set of propositional clauses C_1 is satisfiable and from D_2 to deciding whether a set of propositional clauses C_2 is unsatisifiable.

Let $V_1 = \{v^1_1, \ldots, v^1_n\}$ be the set of atoms in C_1 and $V_2 = \{v^2_2, \ldots, v^2_m\}$ be the set of atoms in C_2. For any clause $c \in C_1 \cup C_2$, $constraint(c)$ represents an annotated constraint form of c. For example, $v^1_1 \vee \neg v^1_2 \vee \neg v^1_3$ is represented as $\texttt{:-not\_v}^1\texttt{@1, v}^1\texttt{@2, v}^1\texttt{@3}$.

Consider the ASG learning task $T = \langle G, \emptyset, \{\text{“pos''}\}, \{\text{“neg''}\}\rangle$, where G is the following propositional Horn ASG:

$\quad\texttt{start} \rightarrow \text{“}\texttt{pos}\text{”}\ \texttt{a}_1 \ldots \texttt{a}_\texttt{n}\ \{constraint(c) | c \in C_1\}$
$\quad\texttt{start} \rightarrow \text{“}\texttt{neg}\text{”}\ \texttt{b}_1 \ldots \texttt{b}_\texttt{m}\ \{constraint(c) | c \in C_2\}$
$\quad\%\ \texttt{for each i} \in [1, \texttt{n}]$
$\quad\texttt{a}_\texttt{i} \rightarrow \{\texttt{v}^1.\}$
$\quad\texttt{a}_\texttt{i} \rightarrow \{\texttt{not\_v}^1.\}$
$\quad\%\ \texttt{for each i} \in [1, \texttt{m}]$
$\quad\texttt{b}_\texttt{i} \rightarrow \{\texttt{v}^2.\}$
$\quad\texttt{b}_\texttt{i} \rightarrow \{\texttt{not\_v}^2.\}$

Note that C_1 is satisfiable iff there is an interpretation I of the atoms in V_1 such that $\{\texttt{v@i.} | v_i \in I\} \cup \{\texttt{not\_v@i.} | v_i \notin I\}\ \{constraint(c) | c \in C_1\}$ is satisfiable. This is the case iff there is an interpretation I of the atoms in V_1 such that I is a model of C_1. The parse trees of G for the string “pos” generate the full set of interpretations of the atoms in V_1, and hence there is a parse tree PT of G for “pos” such that $G[PT]$ is satisfiable iff C_1 is satisfiable. Similarly, there is a parse tree PT of G for “neg” such that $G[PT]$ is satisfiable iff C_2 is satisfiable.

Hence, the hypothesis $H = \emptyset$ is a solution of the learning task T iff the decision problem D returns true. Hence, any decision problem in DP can be polynomially reduced to BV. Hence, BV is DP-hard. □

Theorem 10. *Unstratified BV is in DP.*

Proof. Checking whether H is a solution of a given learning task $T = \langle G, S_M, E^+, E^-\rangle$ at depth d corresponds to checking that for each positive example $s \in E^+$, $s \in \mathcal{L}^d(G)$ and for each negative example $s \in E^-$, $s \notin \mathcal{L}^d(G)$. As propositional unstratified BAM is in NP, this means that there is a pair of sets of decision problems (each in NP) $D^+ = \{D^+_1, \ldots D^+_{|E^+|}\}$ and $D^- = \{D^-_1, \ldots D^-_{|E^-|}\}$ such that $H \in ILP^d_{ASG}(T)$ iff each problem in D^+ returns yes and each problem in D^- returns no.

Each decision problem D^i_j can be mapped to a set of propositional clauses C^i_j such that C^i_j is satisfiable iff D^i_j returns yes. Without loss of generality, we can assume that the atoms used in each set of clauses are disjoint. Hence, $H \in ILP^d_{ASG}$ iff $C^+_1 \cup \ldots \cup C^+_{|E^+|}$ is satisfiable and for each $i \in [1, |E^-|]$, C^-_i is unsatisfiable. This is the case iff $C^+_1 \cup \ldots \cup C^+_{|E^+|}$ is satisfiable and $\{v_1 \vee \ldots \vee v_{|E^-|}\} \cup \left\{c \vee \neg v_i \middle| i \in [1, |E^-|], c \in C^-_i\right\}$ is unsatisfiable (where the v_i's are new atoms). Hence, deciding BV can be reduced to deciding one problem in NP and one problem in $coNP$. Hence, BV is a member of DP. □

Theorem 11. *Horn BTS is Σ^P_2-hard.*

Proof. We prove this by reducing the Σ^P_2-complete problem of deciding whether $\Phi \in QBF_{2,\exists}$, where $\Phi = \exists x_1, \ldots, \exists x_m, \forall x_{m+1}, \ldots, \forall x_n E$, where E is a disjunction $C_1 \vee \ldots \vee C_k$ of conjunctions of length 3 over the atoms (or negations of atoms) in $\{x_1, \ldots, x_m, x_{m+1}, \ldots, x_n\}$.

Let $constraint(C_j)$ be a denial representation of C_j using annotations. So, for example, $x_1 \wedge \neg x_3 \wedge x_5$ is represented as $\texttt{:- v@1, not\_v@3, v@5}$.

Consider the ASG learning task $T = \langle G, S_M, \{\text{“pos''}\}, \{\text{“neg''}\}\rangle$, where G is the following propositional Horn ASG:

$\quad 1 : \texttt{start} \rightarrow \texttt{x}_1 \ldots \texttt{x}_\texttt{m}\ \text{“}\texttt{pos}\text{”}\ \{\}$
$\quad 2 : \texttt{start} \rightarrow \texttt{x}_1 \ldots \texttt{x}_\texttt{n}\ \text{“}\texttt{neg}\text{”}\ \{constraint(C_j) | j \in [1, k]\}$

$\quad\%\ \texttt{for each i} \in [1, \texttt{m}]$
$\quad\texttt{a}_\texttt{i} : \texttt{start} \rightarrow \texttt{x}_\texttt{i}\ \text{“}\texttt{neg}\text{”}\ \{$
$\qquad\texttt{:- v@1.}$
$\qquad\texttt{:- not\_v@1.}$
$\quad\}$
$\quad\texttt{b}_\texttt{i} : \texttt{x}_\texttt{i} \rightarrow \{$
$\qquad\texttt{:- v, not\_v.}$
$\quad\}$

$\quad\%\ \texttt{for each i} \in [\texttt{m}+1, \texttt{n}]$
$\quad\texttt{c}_\texttt{i} : \texttt{x}_\texttt{i} \rightarrow \{\texttt{v.}\}$
$\quad\texttt{d}_\texttt{i} : \texttt{x}_\texttt{i} \rightarrow \{\texttt{not\_v.}\}$

The hypothesis space of the task, $S_M = \{\langle \texttt{v.}, i\rangle, \langle \texttt{not\_v.}, b_i\rangle | b_i \in [1, m]\}$, means that each hypothesis represents an assignment to the existential variables in Φ. Note that the a_i rules mean that in order to cover the negative example, every inductive solution must contain at least one of the facts $\texttt{v}$ or $\texttt{not\_v}$ in each production rule b_i. The constraint in each production rule b_i means that none of the b_i production rules can contain both $\texttt{v}$ and $\texttt{not\_v}$. Hence the only possible inductive solutions contain exactly one of $\langle \mathtt{v.}, b_i\rangle$ or $\langle \mathtt{not\_v.}, b_i\rangle$ for each $i \in [1, m]$.

Let θ be an assignment to $\{x_1, \ldots, x_m\}$ such that $\forall x_{m+1}, \ldots, \forall x_n E$. Then the hypothesis corresponding to θ cannot accept the negative example with the second production rule (as every assignment to $\{x_{m+1}, \ldots, x_n\}$ must violate at least one of the constraints in the ASP of the second production rule). Conversely let θ be an assignment to $\{x_1, \ldots, x_m\}$ such that $\neg\forall x_{m+1}, \ldots, \forall x_n E$. Then the hypothesis corresponding to θ accepts the negative example with the second production rule (as there is at least one assignment to $\{x_{m+1}, \ldots, x_n\}$ which does not satisfy any of the conjunctions in E, and thus does not violate any of the constraints in the ASP of the second production rule).

Hence, T is satisfiable at depth d (for any $d \geq 1$) iff Φ is valid (i.e. iff $\Phi \in QBF_{2,\exists}$). Hence, as deciding whether $\Phi \in QBF_{2,\exists}$ is Σ_2^P-complete, deciding BTS is Σ_2^P-hard. $\square$

Theorem 13. *Let T be an ASG learning task.* $ILP^d_{ASG}(T) = \left\{H^{ASG} \mid H \in ILP^{context}_{LAS}(LAS(T, d))\right\}$

Proof. Let T be the ASG learning task $\langle G, S_M, E^+, E^-\rangle$.

Let $LAS(T, d) = \langle B_{LAS}, S_M^{LAS}, E^+_{LAS}, E^-_{LAS}\rangle$. Given any hypothesis $H \subseteq S_M$, we write H^{LAS} to denote the hypothesis $\{\langle R_X(PR_{id}) \in H | PR_{id}, R\rangle \in S_M\}$.

$(*)$ First note that for any parse tree PT of $(G : H)_{CF}$ of depth d, (for any $H \subseteq S_M$) $(G : H)[PT]$ is satisfiable iff $\left\{ R_X(PR_{id}) \middle| \begin{array}{c} PR \in (G : H)_{PR}, \\ PR = n \rightarrow n_1 \ldots n_k\ P, \\ R \in P \end{array} \right\} \cup$ $\{\mathtt{pr(rule(n)_{id}, trace(n)).} | n \in PT\}$ is satisfiable, which is the case iff $B \cup H^{LAS}$ accepts $\langle\langle\emptyset, \emptyset\rangle, \{\mathtt{pr(rule(n)_{id}, trace(n)).} | n \in PT\}\rangle$.

Assume that $H \in ILP^d_{ASG}(T)$

$\Leftrightarrow$ $H \subseteq S_M, \forall s \in E^+, s \in \mathcal{L}^d(G : H)$ and $\forall s \in E^-, s \notin \mathcal{L}^d(G : H)$.

$\Leftrightarrow$ $H \subseteq S_M, \forall s \in E^+, \exists PT$ st PT is a parse tree of s for $(G : H)_{CF}$ at depth d and $(G : H)[PT]$ is satisfiable and $\forall s \in E^-, \forall PT$ st PT is a parse tree of s for $(G : H)_{CF}$ at depth d, $(G : H)[PT]$ is unsatisfiable.

$\Leftrightarrow$ $H \subseteq S_M$, $\forall s \in E^+$ st $\{PT_1, \ldots, PT_m\}$ is the set of all parse trees of s for $(G : H)_{CF}$ at depth d, $\exists i \in [1, m]$ st $B \cup H^{LAS}$ accepts $\langle\langle\emptyset, \emptyset\rangle, \{\mathtt{pr(rule(n)_{id}, trace(n)).} | n \in PT_i\}\rangle$ and $\forall s \in E^-, \forall PT$ st PT is a parse tree of s for $(G : H)_{CF}$ at depth d, $(G : H)[PT]$ is unsatisfiable. (by $(*)$).

$\Leftrightarrow$ $H \subseteq S_M$, $\forall s \in E^+$ st $\{PT_1, \ldots, PT_m\}$ is the set of all parse trees of s for $(G : H)_{CF}$ at depth d, $B \cup H^{LAS}$ accepts $\langle\langle\emptyset, \emptyset\rangle, \{\mathtt{1\{pt_1, \ldots, pt_m\}1.}\} \cup \{\mathtt{pr(rule(n)_{id}, trace(n))\text{:-}pt_i.} | i \in [1, m], n \in PT_i\}\rangle$ and $\forall s \in E^-, \forall PT$ st PT is a parse tree of s for $(G : H)_{CF}$ at depth d, $(G : H)[PT]$ is unsatisfiable.

$\Leftrightarrow$ $H \subseteq S_M$, $\forall e^+ \in E^+_{LAS}$, $B \cup H^{LAS}$ accepts e^+ and $\forall s \in E^-, \forall PT$ st PT is a parse tree of s for $(G : H)_{CF}$ at depth d, $(G : H)[PT]$ is unsatisfiable.

$\Leftrightarrow$ $H \subseteq S_M$, $\forall e^+ \in E^+_{LAS}$, $B \cup H^{LAS}$ accepts e^+ and $\forall s \in E^-, \forall PT$ st PT is a parse tree of s for $(G : H)_{CF}$ at depth d, $B \cup H^{LAS}$ does not accept $\langle\langle\emptyset, \emptyset\rangle, \{\mathtt{pr(rule(n)_{id}, trace(n)).} | n \in PT_i\}\rangle$. (by $(*)$).

$\Leftrightarrow$ $H \subseteq S_M$, $\forall e^+ \in E^+_{LAS}$, $B \cup H^{LAS}$ accepts e^+ and $\forall e^- \in E^-_{LAS}$, $B \cup H^{LAS}$ does not accept e^-.

$\Leftrightarrow$ $H^{LAS} \in ILP^{context}_{LAS}(LAS(T))$.

$\Leftrightarrow$ $H \in \left\{H^{ASG} \middle| H \in ILP^{context}_{LAS}(LAS(T, d))\right\}$

$\square$