Appendix A Answer Set Programming

ASP (Gelfond and Lifschitz, 1991; Marek and Truszczynski, 1999) is a knowledge representation language with roots in the research on the semantics of logic programming languages and non-monotonic reasoning. The syntax of the language is defined as follows.

Let Σ be a signature containing constant, function and predicate symbols. Terms and atoms are formed as in first-order logic. A *literal* is an atom a or its negation $\neg a$. A *rule* is a statement of the form:

$$h_1 \vee h_2 \ldots \vee h_k \leftarrow l_1, \ldots, l_m, \text{not } l_{m+1}, \ldots, \text{not } l_n \tag{A1}$$

where each h_i and l_i is a literal and *not* is called *default negation* operator. The intuitive meaning of A1 is given in terms of a rational agent reasoning about its own beliefs and it is summarized by the statement "a rational agent that believes $l_1, \ldots, l_m$ and has no reason to believe $l_{m+1}, \ldots, l_n$, must believe one of $h_1, \ldots, h_k$." If $m = n = 0$, symbol $\leftarrow$ is omitted and the rule is a *fact*. Rules of the form $\bot \leftarrow l_1, \ldots, \text{not } l_n$ are abbreviated $\leftarrow l_1, \ldots, \text{not } l_n$, and called *constraints*, intuitively meaning that $\{l_1, \ldots, \text{not } l_n\}$ must not be satisfied. A rule with variables is interpreted as a shorthand for the set of rules obtained by replacing the variables with all possible variable-free terms. A *program* is a set of rules over Σ.

Next, we define the semantics of ASP. We say that a consistent set S of literals is closed under a rule if $\{h_1, \ldots, h_k\} \cap S \neq \emptyset$ whenever $\{l_1, \ldots, l_m\} \subseteq S$ and $\{l_{m+1}, \ldots, l_n\} \cap S = \emptyset$. Set S is an answer set of a *not*-free program Π if S is the minimal set closed under its rules. The reduct, Π^S, of a program Π w.r.t. S is obtained from Π by removing every rule containing an expression "not l" s.t. $l \in S$ and by removing every other occurrence of not l. Set S is an answer set of Π if it is the answer set of Π^S.

Appendix B Proofs of Theorems

In this appendix, we provide proofs of the main results of this paper.

B.1 Proof of Theorem 1

Before we proceed to the proof of Theorem 1, we need to introduce the following notions. Let AD be an action description of $\mathcal{AL}_{IR}$, n be a positive integer, and $\Sigma(AD)$ be the signature of AD. $\Sigma^n(AD)$ denotes the signature obtained as follows:

- $const(\Sigma^n(AD)) = const(\Sigma(AD)) \cup \{0, \ldots, n\}$
- $pred(\Sigma^n(AD)) = \{holds, u, split, occurs\}$

Let

$$\alpha^n(AD) = \langle \Sigma^n(AD), \Pi^\alpha(AD) \rangle, \tag{B1}$$

where

$$\Pi^\alpha(AD) = \bigcup_{r \in AD} \alpha(r), \tag{B2}$$

and $\alpha(r)$ is defined as follows:

- $\alpha(e \text{ causes } \lambda \text{ if } l_1, \ldots, l_n)$ is

$$\chi(\lambda, I+1) \leftarrow occurs(e, I), \chi(l_1, I), \ldots, \chi(l_n, I). \tag{B3}$$

if λ is a fluent literal. If λ is of the form $u(f)$, the translation of the law is

$$u(f, I+1) \leftarrow occurs(e, I), \chi(l_1, I), \ldots, \chi(l_n, I), \text{not } split(f, I). \tag{B4}$$

$$\chi(f, I+1) \vee \chi(\neg f, I+1) \leftarrow occurs(e, I), \chi(l_1, I), \ldots, \chi(l_n, I), split(f, I). \tag{B5}$$

- $\alpha(l_0 \text{ if } l_1, \ldots, l_n)$ is

$$\chi(l_0, T) \leftarrow \chi(l_1, T), \ldots, \chi(l_n, T). \tag{B6}$$

- $\alpha(e \text{ impossible\_if } l_1, \ldots, l_n)$ is

$$\leftarrow \chi(l_1, T), \ldots, \chi(l_n, T), occurs(e, T).$$

Let also

$$\Phi^n(AD) = \langle \Sigma^n(AD), \Pi^\Phi(AD) \rangle, \tag{B7}$$

where

$$\Pi^\Phi(AD) = \Pi^\alpha(AD) \cup \Pi \tag{B8}$$

and Π contains the following rules:

$$\chi(F, I+1) \leftarrow \chi(F, I), \text{not } \chi(\neg F, I+1), \text{not } u(F, I+1). \tag{B9}$$

$$\chi(\neg F, I+1) \leftarrow \chi(\neg F, I), \text{not } \chi(F, I+1), \text{not } u(F, I+1). \tag{B10}$$

$$u(F, I+1) \leftarrow u(F, I), \text{not } \chi(F, I+1), \text{not } \chi(\neg F, I+1). \tag{B11}$$

Π also contains the following rules:

$$\leftarrow \chi(F, I), u(F, I). \tag{B12}$$

$$\leftarrow \chi(\neg F, I), u(F, I). \tag{B13}$$

When we refer to a single action description, we drop argument AD from the above expressions.

For the rest of this section, we will focus on ground programs. In order to keep notation simple, we will use α^n and Φ^n to denote the ground versions of the programs previously defined.

The following notation will be useful in our further discussion. Given a time point t, a state σ, and a compound action a, let

$$\begin{aligned} \chi(\sigma, t) &= \{\chi(l, t) \mid l \in \sigma \cap Lit\} \cup \\ &\quad \{u(f, t) \mid u(f) \in \sigma\} \\ occurs(a, t) &= \{occurs(e, t) \mid e \in a\} \end{aligned} \tag{B14}$$

These sets can be viewed as the representation of σ and a in ASP. Let also

$$split(q_t, t) = \{split(f, t) \mid f \in q_t\}$$

which represents a set of fluents to which reasoning by cases should be applied according to a qualifier q_t.

For any action description AD, state σ_0, and qualified action sequence $s = \langle a_0/q_0, \ldots, a_{n-1}/q_{n-1} \rangle$, let $\Phi^n(\sigma_0, s)$ denote

$$\Phi^n \cup \{occurs(a_i, i) \mid a_i \text{ is in } s\} \cup \{split(q_i, i) \mid q_i \text{ is in } s\} \tag{B15}$$

Where possible, we drop the first argument, and denote the program by $\Phi^n(\sigma_0, s)$. Also, for convenience, we write $\Phi^1(\sigma_0, a_0, q_0)$ when $n = 1$.

An important property of Cn_Z that we will use later is:

Lemma 3
For every fluent f, $u(f) \in Cn_Z$ *iff* $u(f) \in S$.

Proof
The thesis follows trivially from the observation that proper extended literals do not occur in state constraints. $\square$

The following lemma will be helpful in proving the main result of this section. It states the correspondence between (single) transitions of the transition diagram and answer sets of the corresponding ASP program.

Lemma 4
Let AD be an action description and $\mathcal{T}(AD)$ be the transition diagram it describes. Then, $\langle \sigma_0, a_0, \sigma_1 \rangle \in \mathcal{T}(AD)$ iff $\sigma_1 = \{l \mid \chi(l,1) \in A\} \cup \{u(f) \mid u(f,1) \in A\}$ for some qualifier q_0 and some answer set A of $\Phi^1(\sigma_0, a_0, q_0)$.

Proof
Let us define

$$Y_{\sigma_0,a_0,q_0} = \chi(\sigma_0, 0) \cup occurs(a_0, 0) \cup split(q_0, 0) \tag{B16}$$

and

$$\Phi^1(\sigma_0, a_0, q_0) = \Phi^1 \cup Y_{\sigma_0,a_0,q_0}$$

Left-to-right. Let us construct the qualifier q_0 as:

$$\begin{aligned} q_0 = \{f \mid\ & e \text{ causes } u(f) \text{ if } \Gamma \in AD, \\ & e \in a_0, \Gamma \subseteq \sigma_0, \text{ and} \\ & u(f) \notin \sigma_1\} \end{aligned} \tag{B17}$$

The set q_0 is an ASP representation of a qualifier q_0 in a qualified action sequence.

Let us show that, if $\langle \sigma_0, a_0, \sigma_1 \rangle \in \mathcal{T}(AD)$, then

$$A = Y_{\sigma_0,a_0,q_0} \cup \chi(\sigma_1, 1) \tag{B18}$$

is an answer set of $\Phi^1(\sigma_0, a_0, q_0)$. Notice that $\langle \sigma_0, a_0, \sigma_1 \rangle \in \mathcal{T}(AD)$ implies that σ_1 is a state. Herein, we refer to $\Phi^1(\sigma_0, a_0, q_0)$ as P.

Let us prove that A is the minimal set of literals closed under the rules of the reduct P^A. Let $\mathbb{N}^{\alpha^1(AD)}$ be the set of rules of $\alpha^1(AD)$ of form (B4). P^A contains:

a) set Y_{σ_0,a_0,q_0}.
b) all rules in $\alpha^1(AD) \setminus \mathbb{N}^{\alpha^1(AD)}$.
c) a rule

$$u(f,1) \leftarrow occurs(e,0), \chi(l_1,0), \ldots, \chi(l_n,0).$$

   for every fluent f such that $split(f,0) \notin A$.
d) a rule

$$\chi(l,1) \leftarrow \chi(l,0)$$

   for every fluent literal l such that $\chi(l,1) \in A$ and a rule

$$\chi(\neg l,1) \leftarrow \chi(\neg l,0)$$

   for every fluent literal $\neg l$ such that $\chi(\neg l,1) \in A$.

e) a rule

$$u(f, 1) \leftarrow u(f, 0)$$

for every fluent f such that $u(f, 1) \in A$.

Note that because A is an answer set, $\chi(f, 1) \in A \Leftrightarrow \chi(\neg f, 1) \notin A$ and $u(f, 1) \notin A$. The conditions for $\chi(\neg f) \in A$ and $u(f) \in A$ can be similarly described.

<u>A is closed under P^A</u>. We will prove it for every rule of the program.

1. Rules of groups (a), (d), and (e): obvious.
2. Rules of group (b) encoding dynamic laws of the form e causes λ if $l_1, \ldots, l_n$ when λ is a fluent literal:
$$\chi(\lambda, 1) \leftarrow occurs(e, 0), \chi(l_1, 0), \ldots, \chi(l_n, 0).$$
If $\{o(e, 0), \chi(l_1, 0), \ldots, \chi(l_n, 0)\} \subseteq A$, then, by (B18), $\{l_1, \ldots, l_n\} \subseteq \sigma_0$ and $e \in a_0$. Therefore, the preconditions of the dynamic law are satisfied by σ_0. Hence (4) implies $\lambda \in \sigma_1$. By (B18), $\chi(\lambda, 1) \in A$.
3. Rules of group (b) encoding dynamic laws of the form e causes λ if $l_1, \ldots, l_n$ when λ is of the form $u(f)$:
$$\chi(f, 1) \vee \chi(\neg f, 1) \leftarrow occurs(e, 0), \chi(l_1, 0), \ldots, \chi(l_n, 0), split(f, 0).$$
Let us suppose that $split(f, 0) \in A$. In fact, if that is not the case, then A is trivially closed under the rule. Similarly, assume $\{occurs(e, 0), \chi(l_1, 0), \ldots, \chi(l_n, 0)\} \subseteq A$. Then, by construction of Y_{σ_0, a_0, q_0}, $split(f, 0) \in split(q_0, 0)$. In turn, by construction of $split(q_0, 0)$ and from (B17) we conclude that $f \in q_0$ and that $u(f) \notin \sigma_1$. Because σ_1 is complete from (5), we conclude that either f or $\neg f$ is in σ_1. By (B18), either $\chi(f, 1) \in A$ or $\chi(\neg f, 1) \in A$.
4. Rules of group (b) encoding state constraints of the form l_0 if $l_1, \ldots, l_n$:
$$\chi(l_0, t) \quad \leftarrow \quad \chi(l_1, t), \ldots, \chi(l_n, t).$$
If $\{\chi(l_1, t), \ldots, \chi(l_n, t)\} \subseteq A$, then, by (B18), $\{l_1, \ldots, l_n\} \subseteq \sigma_t$, i.e. the preconditions of the state constraint are satisfied by σ_t. If $t = 1$, then (5) implies $l_0 \in \sigma_1$. By (B18), $\chi(l_0, t) \in A$. If $t = 0$, since states are closed under the state constraints of AD, we have that $l \in \sigma_0$. Again by (B18), $\chi(l_0, t) \in A$.
5. Rules of group (b) encoding executability conditions of the form e impossible_if $l_1, \ldots, l_n$:
$$\leftarrow \quad occurs(e, 0), \chi(l_1, 0), \ldots, \chi(l_n, 0).$$
Since $\langle \sigma_0, a_0, \sigma_1 \rangle \in \mathcal{T}(AD)$ by hypothesis, $\langle \sigma_0, a_0 \rangle$ does not satisfy the preconditions of any executability condition. Then, either $e \notin a_0$ or $l_i \notin \sigma_0$ for some i. By (B18), the body of this rule is not satisfied.
6. Rules of group (c) encoding dynamic laws when λ is of the form $u(f)$:
$$u(f, 1) \leftarrow occurs(e, 0), \chi(l_1, 0), \ldots, \chi(l_n, 0).$$
If the rule is in P^A, then $split(f, 0) \notin A$. By construction of Y_{σ_0, a_0, q_0}, $split(f, 0) \notin split(q_0, 0)$ By construction of $split(q_0, 0)$, $f \notin q_0$ and from (B17) it follows that $u(f) \in \sigma_1$. By (B18), $u(f, 1) \in A$.

$\underline{A \text{ is the minimal set closed under the rules of } P^A}$. We will prove this by assuming that there exists a set $B \subseteq A$ such that B is closed under the rules of P^A, and by showing that $B = A$.

First of all,

$$Y_{\sigma_0,a_0,q_0} \subseteq B, \tag{B19}$$

since these are facts in P^A.

Let

$$\delta = \{l \mid \chi(l,1) \in B\}. \tag{B20}$$

Since $B \subseteq A$,

$$\delta \subseteq \sigma_1 \tag{B21}$$

Let W be the element of $\mathbb{E}(a_0, \sigma_0)$ satisfying (4). We will show that $\delta = \sigma_1$ by proving that

$$\delta = CN_Z(W \cup (\sigma_1 \cap \sigma_0)). \tag{B22}$$

Dynamic laws. Let d be a dynamic law of AD of the form e causes λ if $l_1, \ldots, l_n$, such that $e \in a_0$ and $\{l_1, \ldots, l_n\} \subseteq \sigma_0$. Because of (B19), $\chi(\{l_1, \ldots, l_n\}, 0) \subseteq B$ and $o(e, 0) \in B$. If λ is a fluent literal, then since B is closed under $\alpha(d)$, $\chi(\lambda, 1) \in B$, and $\lambda \in \delta$. Therefore, $W \subseteq \delta$. It can be similarly shown if λ is a properly extended literal.

Inertia. P^A contains a (reduced) inertia rule of the form

$$\chi(f,1) \leftarrow \chi(f,0). \tag{B23}$$

for every fluent $f \in \sigma_1$. Suppose $l \in \sigma_1 \cap \sigma_0$. Then, $\chi(l,0) \in Y_{\sigma_0,a_0,q_0}$, and, since B is closed under (B23), $\chi(f,1) \in B$. Therefore, $\sigma_1 \cap \sigma_0 \subseteq \delta$. The same argument applies to the other reduced inertia rules.

State constraints. Let r be a state constraints of AD of the form l_0 if $l_1, \ldots, l_n$, such that

$$\chi(\{l_1, \ldots, l_n\}, 0) \subseteq B. \tag{B24}$$

Since B is closed under $\alpha(r)$, $\chi(l_0, 1) \in B$, and $l_0 \in \delta$. Then, δ is closed under the state constraints of AD.

Summing up, (B22) holds. From (4) and (B21), we obtain $\sigma_1 = \delta$. Therefore $\chi(\sigma_1, 1) \subseteq B$.

At this point we have shown that $Y_{\sigma_0,a_0,q_0} \cup \chi(\sigma_1, 1) \subseteq B \subseteq A$.

Right-to-left. Let A be an answer set of P and let $\sigma_1 = \{l \mid \chi(l,1) \in A\} \cup \{u(f) \mid u(f,1) \in A\}$. We have to show that

$$\sigma_1 = CN_Z(W \cup (\sigma_1 \cap \sigma_0)) \text{ for some } W \in \mathbb{E}(a_0, \sigma_0) \tag{B25}$$

as well as that $\langle \sigma_0, a_0 \rangle$ respects all executability conditions and that σ_1 is consistent and complete.

σ_1 consistent. Obvious, since A is a (consistent) answer set by hypothesis.

σ_1 complete. By contradiction, and without loss of generality, let f be a fluent s.t. $f \notin \sigma_1$, $\neg f \notin \sigma_1$, $u(f) \notin \sigma_1$, and $f \in \sigma_0$ (since σ_0 is complete by hypothesis, if $f \notin \sigma_0$, we can still select $\neg f$ or $u(f)$). Then, the reduct P^A contains a rule

$$\chi(f,1) \leftarrow \chi(f,0). \tag{B26}$$

Since A is closed under P^A, $\chi(f,1) \in A$ and $f \in \sigma_1$. Contradiction.

Executability conditions respected. By contradiction, assume that law r of form e impossible_if

$l_1, \ldots, l_n$ is not respected. Note that $\chi(\{l_1, \ldots, l_n\}, 0) \subseteq A$ and $occurs(e, 0) \in A$. Therefore, the body of $\alpha(r)$ is satisfied by A, and A is not a answer set.

(B25) holds. Let us construct W so that:

- $W \supseteq E(a_0, \sigma_0) \cap Lit$
- for every $u(f) \in E(a_0, \sigma_0)$:
  - — if $f \notin q_0$, then $u(f) \in W$
  - — otherwise, $f \in W$ if $\chi(f, 1) \in A$ and $\neg f \in W$ if $\chi(\neg f, 1) \in A$.

One can check that $W \in \mathbb{E}(a_0, \sigma_0)$.

Next, let us prove that $\sigma_1 \supseteq W$, i.e. that for every $\lambda \in W, \lambda \in \sigma_1$. Suppose $\lambda \in E(a_0, \sigma_0) \cap Lit$. There must exist a dynamic law d of the form (1) such that $\{l_1, \ldots, l_n\} \subseteq \sigma_0$ and $e \in a_0$. Since A is closed under (B3) of $\alpha(d)$, it follows that $\chi(\lambda, 1) \in A$. By construction of σ_1, $\lambda \in \sigma_1$.

Let us now consider the case in which $\lambda \notin E(a_0, \sigma_0) \cap Lit$. There must be a dynamic law d of the form e causes $u(f)$ if $l_1, \ldots, l_n$ such that f is the fluent that occurs in λ. It must be the case that $\{l_1, \ldots, l_n\} \subseteq \sigma_0$, and $e \in a_0$. Note that either $f \in q_0$ or $f \notin q_0$.

If $f \notin q_0$, then by construction of W it must be the case that λ is $u(f)$. Let us consider (B4) from $\alpha(d)$. Because A is closed under it, it follows that $u(f, 1) \in A$. By construction of σ_1, we conclude that $u(f) \in \sigma_1$.

Next, consider the case in which $f \in q_0$. If λ is f, then by construction of W, one can conclude that $\chi(f, 1) \in A$. It follows, then, that $f \in \sigma_1$. If λ is $\neg f$, with similar reasoning we derive that $\neg f \in \sigma_1$. This concludes the proof that $\sigma_1 \supseteq W$.

Additionally, $\sigma_1 \supseteq \sigma_1 \cap \sigma_0$ is trivially true.

Let us prove that σ_1 is closed under the state constraints of AD. Consider a state constraint s, of the form l_0 if $l_1, \ldots, l_n$, such that $\{l_1, \ldots, l_n\} \subseteq \sigma_0$. Since A is closed under $\alpha(s)$, $\chi(l_0, 1) \in A$. By construction of σ_1, $l_0 \in \sigma_1$.

Let us prove that σ_1 is the minimal set satisfying all conditions. By contradiction, assume that there exists a set $\delta \subset \sigma_1$ such that $\delta \supseteq W \cup (\sigma_1 \cap \sigma_0)$ and that δ is closed under the state constraints of AD. We will prove that this implies that A is not an answer set of P.

Let A' be the set obtained by removing from A all literals $\chi(l, 1)$ such that $l \in \sigma_1 \setminus \delta$ and all atoms of form $u(f, 1)$ such that $u(f) \in \sigma_1 \setminus \delta$. Since $\delta \subset \sigma_1$, $A' \subset A$.

Since $\delta \supseteq W \cup (\sigma_1 \cap \sigma_0)$, for every extended fluent literal $\lambda \in \sigma_1 \setminus \delta$ it must be true that $\lambda \notin \sigma_0$ and $\lambda \notin W$. From Lemma 3, we conclude that λ must be a fluent literal. Therefore there must exist (at least) one state constraint λ if $l_1, \ldots, l_n$ such that $\{l_1, \ldots, l_n\} \subseteq \sigma_1$ and $\{l_1, \ldots, l_m\} \not\subseteq \delta$. Hence, A' is closed under the rules of P^A. This proves that A is not an answer set of P. Contradiction. $\square$

Corollary 2

Let AD be an action description and $\mathcal{T}(AD)$ be the transition diagram it describes. Then, $\langle \sigma_0, a_0, \sigma_1, \ldots, a_{n-1}, \sigma_n \rangle$ is a path of $\mathcal{T}(AD)$ iff there exist qualifiers $q_0, q_1, \ldots, q_{n-1}$ and an answer set A of $\Phi^n(\sigma_0, \langle a_0/q_0,\ a_1/q_1,\ \ldots,\ a_{n-1}/q_{n-1} \rangle)$ such that, for every $1 \leq i \leq n$, $\sigma_i = \{l \mid \chi(l, i) \in A\} \cup \{u(f) \mid u(f, i) \in A\}$.

Proof

The thesis can be easily proven by induction from Lemma 4. $\square$

Theorem 1

Let I be a consistent set of fluent literals, F be a set of fluents, and s be a qualified action sequence. A path π is a model of $[\gamma(I,F),\, s]$ iff there exists an answer set of $\Pi_{AD}(I,F,s)$ that encodes π.

Proof
The proof leverages Corollary 2 and the Splitting Set Lemma (Lifschitz and Turner, 1994). First of all, note that it is possible to split $\Pi_{AD}(I,F,s)$ in such a way that the bottom corresponds to rules $[\mathbf{g_1}]$, $[\mathbf{g_2}]$, $[\mathbf{g_3}]$ (see Section 5.1) together with facts encoding I and F, as well as rules encoding the state constraints for time step 0. One can check that the answer sets of the bottom encode the completion $\gamma(I,F)$, and that every element of $\gamma(I,F)$ is a state of $\tau(AD)$.

The thesis follows from the application of Corollary 2 to each $\sigma_0 \in \gamma(I,F)$, after noticing the correspondence between the top of $\Pi_{AD}(I,F,s)$ and program $\Phi^n(\sigma_0,s)$. $\square$

B.2 Proof of Theorem 2

Theorem 2
A source $\mathcal{S}$ is a match for a query $\boldsymbol{q}$ iff `FindMatch`(I,$\aleph$,$\boldsymbol{q}$)$\neq\ \bot$. The semantic score of $\mathcal{S}$ is $||$`FindMatch`(I,$\aleph$,$\boldsymbol{q}$)$||$.

Proof
We begin by showing that the algorithm terminates. This follows simply from the consideration that, in the worst case, the algorithm proceeds to a systematic enumeration of the subsets of $\mathcal{F}$ and of the extensions of $\aleph$ (refer to steps (3), (5), and (7)), which are clearly finite, and terminates when all have been enumerated (step (6)).

Next, we demonstrate that if $\Pi_{AD}(I,\mathcal{F}\setminus\mathcal{D},\aleph^\times)$ has at least one answer set, then step (1) of the algorithm finds $\varepsilon(I,\aleph)$, i.e. that $I' = \varepsilon(I,\aleph)$. Note that the existence of an answer set is verified at step (2).

Left-to-right. Let A be an answer set of $\Pi_{AD}(I,\mathcal{F}\setminus\mathcal{D},\aleph^\times)$. From Theorem 1, it follows that A encodes a model π_A of $[\gamma(I,\mathcal{F}\setminus\mathcal{D}),\aleph^\times]$. By construction of $\gamma(I,\mathcal{F}\setminus\mathcal{D})$,

$$\text{there exists } I' \in I[\mathcal{F}\setminus\mathcal{D}] \text{ such that } \pi_A \text{ is a model of } [\gamma(I'),\aleph^\times]. \tag{B27}$$

Note that $l \in I'$ iff $l \in I$ or $I \in \{l' \,|\, \{\chi(l',0), forced(l'_f)\} \subseteq R$, where l'_f is the fluent from which l' is formed. If $l \in I$, then from Proposition 2, $l \in \varepsilon(I,\aleph)$ and the thesis is proven from the observation that the hypothesis of existence of an answer set guarantees the existence of $\varepsilon(I,\aleph)$. In the other case, it follows that $\chi(l,0) \in R$ and that $forced(l_f) \in R$. From the former and (B27), it follows that $l \in \bigcap_{Y\in I[\mathcal{F}\setminus\mathcal{D}]}\gamma(Y)$. Hence,

$$l \in \gamma(Y) \text{ for every } Y \in I[\mathcal{F}\setminus\mathcal{D}]. \tag{B28}$$

By construction of $\Pi_{AD}(I,\mathcal{F}\setminus\mathcal{D},\aleph^\times)$, $forced(l_f) \in R$ iff $l_f \in \mathcal{F}\setminus\mathcal{D}$. By definition of forcing of a fluent, every element of $I[\mathcal{F}\setminus\mathcal{D}]$ contains either l or $\overline{l}$. From Proposition 1, $\gamma(Y)$ is consistent and includes Y. From (B28) and the fact that $l \in \gamma(Y)$, it follows that $l \in Y$. Hence, $l \in \bigcap_{Y\in I[\mathcal{F}\setminus\mathcal{D}]} Y$ and thus $l \in \varepsilon(I,\aleph)$. From the generality of l, it follows that $I' = \varepsilon(I,\aleph)$.

Right-to-left. The conclusion follows from Definition 5 and Theorem 1 in a straightforward way.

Next, we prove that the algorithm terminates at step (4c) iff $\mathcal{S}$ is a match for $\boldsymbol{q}$. From Theorem 1, Corollary 1, and from our observations about step (1), it follows that for every answer set A found at step (4), there exists a model π_A of $[\gamma(\varepsilon(I,\aleph),F),s]$ that encodes A and satisfies condition (**c1**) of Definition 6. With similar considerations, one can conclude that for every answer set B of $\Pi_{AD}(X, \emptyset, \langle\,\rangle)$ there exists a model π_B of $[\gamma(\pi_{\sigma_0} \setminus \varepsilon(I, \aleph), \emptyset), \langle\,\rangle]$, where π_{σ_0} is defined in Definition 6. Using Corollary 1, one can check that the three tests at step (4b) ensure that condition (**c2**) from Definition 6 is satisfied by π_A and π_B. Thus, if the algorithm terminates at step (4c), then $\mathcal{S}$ is a match for $\boldsymbol{q}$.

The right-to-left direction is proven by contradiction. We assume that the algorithm never reaches step (4c), and yet $\mathcal{S}$ is a match for $\boldsymbol{q}$. From Definition 6, it follows that there exist π and π' satisfying conditions (**c1**) and (**c2**). From Theorem 1 and earlier considerations, it follows that there exist answer sets A and B satisfying the conditions from step (4) of the algorithm. This means that the condition of the *if* statement at step (4c) is true, and thus the algorithm must terminate, which yields contradiction. This concludes the proof that a source $\mathcal{S}$ is a match for a query $\boldsymbol{q}$ iff `FindMatch`(I,$\aleph$,$\boldsymbol{q}$)$\neq \bot$.

Next, we demonstrate that the semantic score of $\mathcal{S}$ is $v = ||$`FindMatch`$(I,\aleph,\boldsymbol{q})||$. If the algorithm returns $\bot$, then $v = \infty$ by definition, and thus the thesis is proven. Otherwise, according to Definition 7, we need to prove that there exist F and s such that $v = \Delta(\gamma(\varepsilon(I, \aleph), F)) + \Delta(s)$ and that v is minimal among all possible choices of F and s satisfying conditions (**c1**) and (**c2**) from Definition 6. By construction of $\Pi_{AD}(I, \mathcal{F} \setminus \mathcal{D}, \aleph^{\times})$, Definition 4, and the earlier part of the present theorem, it follows that $\Delta(\gamma(\varepsilon(I, \aleph), F))$ is equal to the number of atoms of A formed by relation $forced$. Similarly, $\Delta(s)$ is equal to the number of atoms of A formed by $split$. Hence, $v = \Delta(\gamma(\varepsilon(I, \aleph), F)) + \Delta(s)$. The minimality of v is demonstrated by contradiction. Let us proceed by cases. Suppose that, when the algorithm terminates at step (4c), $F = \emptyset$ and $s = \aleph^{?}$. By Definition 4 and Definition 6, v is minimal, which yields contradiction. Suppose, then, that $F \neq \emptyset$ or $s \neq \aleph^{?}$. Because the values of the two variables are changed only by step 7, it follows that they were set at that step from the values of F' and s' determined by step 5. However, the values of those variables are selected so that $|F'| + \Delta(s')$ is minimal (step 5b). Contradiction. $\square$