Alternative Characterizations for Program Equivalence under Answer-Set Semantics: Preliminary Report*

Martin Gebser[1], Torsten Schaub[1], Hans Tompits[2], and Stefan Woltran[2]

[1] Institut für Informatik, Universität Potsdam,
August-Bebel-Straße 89, D-14482 Potsdam, Germany
{gebser,torsten}@cs.uni-potsdam.de
[2] Institut für Informationssysteme 184/3, Technische Universität Wien,
Favoritenstraße 9-11, A-1040 Vienna, Austria
{tompits,stefan}@kr.tuwien.ac.at

Abstract. Logic programs under answer-set semantics constitute an important tool for declarative problem solving. In recent years, two research issues received growing attention. On the one hand, concepts like loops and elementary sets have been proposed in order to extend Clark's completion for computing answer sets of logic programs by means of propositional logic. On the other hand, different concepts of program equivalence, like strong or uniform equivalence, have been studied in the context of program optimization and modular programming. In this paper, we bring these two lines of research together and provide alternative characterizations for different conceptions of equivalence in terms of unfounded sets, along with the related concepts of loops and elementary sets. Our results yield new insights into the model theory of equivalence checking. We further exploit these characterizations to develop novel encodings of program equivalence in terms of propositional logic.

1 Introduction

The increasing success of answer-set programming [1] as a tool for declarative problem solving has produced the need to optimize logic programs in various ways, while leaving their semantics unaffected. Different scenarios have led to different criteria of when a program's semantics is preserved. Formally, this is reflected by different definitions of program equivalence (see below). For instance, in solving, one is usually interested in program modifications preserving all answer sets, while program optimization requires a stronger definition, guaranteeing that replacing one subprogram by another preserves answer sets, no matter how the encompassing program looks like.

In what follows, we elaborate upon the model theory underlying program equivalence, dealing primarily with the well-known concepts of SE- and UE-models [2, 3]. In particular, we provide a new perspective on these semantic structures by using *unfounded sets* [4] and related constructs like *elementary sets* [5] and *loops* [6, 7]. Recall that SE- and UE-models are defined as pairs (X, Y), where Y is a model of a given logic program P and X is a model of the reduct P^Y. The major difference between this

* Partially supported by the Austrian Science Fund (FWF) under project P15068-INF.

characterization and our approach is that we refer to $(Y \setminus X)$ rather than to X itself. As it turns out, an explicit reference to the reduct and its models is not required, rather, the respective unfoundedness property possessed by $(Y \setminus X)$ allows us to characterize and distinguish SE- and UE-models.

2 Background

A propositional *disjunctive logic program* is a finite set of rules of the form

$$a_1 \vee \cdots \vee a_k \leftarrow a_{k+1}, \ldots, a_m, \mathit{not}\, a_{m+1}, \ldots, \mathit{not}\, a_n, \tag{1}$$

where $1 \leq k \leq m \leq n$ and every a_i ($1 \leq i \leq n$) is a propositional atom from some universe $\mathcal{U}$; not denotes default negation. A rule as in (1) is called a *fact* if $k = n = 1$; it is said to be *positive* if $m = n$. For a rule r, $H(r) = \{a_1, \ldots, a_k\}$ is the *head* of r, $B(r) = \{a_{k+1}, \ldots, a_m, \mathit{not}\, a_{m+1}, \ldots, \mathit{not}\, a_n\}$ is the *body* of r, $B^+(r) = \{a_{k+1}, \ldots, a_m\}$ is the *positive body* of r, and $B^-(r) = \{a_{m+1}, \ldots, a_n\}$ is the *negative body* of r. We sometimes denote a rule r by $H(r) \leftarrow B(r)$.

The (*positive*) *dependency graph* of a program P is $(\mathcal{U}, \{(a, b) \mid r \in P, a \in H(r), b \in B^+(r)\})$. A nonempty set $U \subseteq \mathcal{U}$ is a *loop* of P if the subgraph of the dependency graph of P induced by U is strongly connected. Similar to Lee [7], we consider every singleton as a loop. A program P is *tight* [8, 9] if every loop of P is a singleton.

As usual, an interpretation Y is a set of atoms (i.e., a subset of $\mathcal{U}$). For a rule r, we write $Y \models r$ iff $H(r) \cap Y \neq \emptyset$, $B^+(r) \not\subseteq Y$, or $B^-(r) \cap Y \neq \emptyset$. An interpretation Y is a *model* of a program P, denoted by $Y \models P$, iff $Y \models r$ for every $r \in P$. The *reduct* of P with respect to Y is $P^Y = \{H(r) \leftarrow B^+(r) \mid r \in P, B^-(r) \cap Y = \emptyset\}$. An interpretation Y is an *answer set* of P iff Y is a minimal model of P^Y.

Two programs, P and Q, are *ordinarily equivalent* iff their answer sets coincide. Furthermore, P and Q are *strongly equivalent* [10] (resp., *uniformly equivalent* [3]) iff, for every program (resp., set of facts) R, $P \cup R$ and $Q \cup R$ are ordinarily equivalent.

For interpretations X, Y, the pair (X, Y) is an *SE-interpretation* iff $X \subseteq Y$. Given an SE-interpretation (X, Y) and a program P, (X, Y) is an *SE-model* [2] of P iff $Y \models P$ and $X \models P^Y$. An SE-model (X, Y) is a *UE-model* [3] of P iff there is no SE-model (Z, Y) of P such that $X \subset Z \subset Y$. The set of all SE-models (resp., UE-models) of P is denoted by $SE(P)$ (resp., $UE(P)$). Two programs P and Q are strongly (resp., uniformly) equivalent iff $SE(P) = SE(Q)$ (resp., $UE(P) = UE(Q)$) [2, 3].

Example 1. Consider $P = \{a \vee b \leftarrow\}$ and $Q = \{a \leftarrow \mathit{not}\, b;\ b \leftarrow \mathit{not}\, a\}$. Clearly, both programs are ordinarily equivalent as $\{a\}$ and $\{b\}$ are their respective answer sets. However, they are not strongly equivalent. Indeed, since P is positive, we have that $SE(P) = \{(a, a), (b, b), (ab, ab), (a, ab), (b, ab)\}$.[1] For Q, we have to take the reduct into account. In particular, we have $Q^{\{a,b\}} = \emptyset$, and so any interpretation is a model of $Q^{\{a,b\}}$. Hence, each pair (X, ab) with $X \subseteq \{a, b\}$ is an SE-model of Q. We thus have $SE(Q) = \{(a, a), (b, b), (ab, ab), (a, ab), (b, ab), (\emptyset, ab)\}$. That is, $SE(P) \neq SE(Q)$,

[1] Whenever convenient, we use strings like ab as a shorthand for $\{a, b\}$. As a convention, we let universe $\mathcal{U}$ be the set of atoms occurring in the programs under consideration.

so P and Q are not strongly equivalent. A witness for this is $R = \{a \leftarrow b;\ b \leftarrow a\}$, as $P \cup R$ has $\{a, b\}$ as its (single) answer set, while $Q \cup R$ has no answer set.

Concerning uniform equivalence, observe first that $UE(P) = SE(P)$. This is not the case for Q, where the SE-model $(\emptyset, ab)$ drops out since there exist further SE-models (Z, ab) of Q with $\emptyset \subset Z \subset \{a, b\}$, viz. (a, ab) and (b, ab). One can check that $(\emptyset, ab)$ is in fact the only pair in $SE(Q)$ that is no UE-model of Q. So, $UE(Q) = SE(Q) \setminus \{(\emptyset, ab)\} = SE(P) = UE(P)$. Thus, P and Q are uniformly equivalent. $\Diamond$

We conclude this section with the following known properties. First, for any program P and any interpretation Y, the following statements are equivalent: (i) $Y \models P$; (ii) $Y \models P^Y$; (iii) $(Y, Y) \in SE(P)$; and (iv) $(Y, Y) \in UE(P)$. Second, if $Y \models P$, Y is an answer set of P iff, for each SE-model (resp., UE-model) (X, Y) of P, $X = Y$.

3 Model-Theoretic Characterizations by Unfounded Sets

In this section, we exploit the notion of an unfounded set [4] and provide alternative characterizations of models for logic programs and program equivalence. Roughly speaking, the aim of unfounded sets is to collect atoms that cannot be derived from a program with respect to a fixed interpretation. Given the closed-world reasoning flavor of answer sets, such atoms are considered to be false. However, we shall relate here unfounded sets also to SE- and UE-models, and thus to concepts that do not fall under the closed-world assumption (since they implicitly deal with program extensions). For the case of uniform equivalence, we shall also employ the recent concept of elementarily unfounded sets [5], which via elementary sets decouple the idea of (minimal) unfounded sets from fixed interpretations. Finally, we link our results to loops.

Given a program P and an interpretation Y, a set $U \subseteq \mathcal{U}$ is *unfounded* [4] for P with respect to Y if, for each $r \in P$, at least one of the following conditions holds:

1. $H(r) \cap U = \emptyset$,
2. $H(r) \cap (Y \setminus U) \neq \emptyset$,
3. $B^+(r) \not\subseteq Y$ or $B^-(r) \cap Y \neq \emptyset$, or
4. $B^+(r) \cap U \neq \emptyset$.

Note that the empty set is unfounded for any program P with respect to any interpretation, since the first condition, $H(r) \cap \emptyset = \emptyset$, holds for all $r \in P$.

Example 2. Consider the following program:

$$P = \left\{ \begin{array}{llllll} r_1: & a \vee b \leftarrow & r_3: & c \leftarrow a & r_5: & c \leftarrow b, d \\ r_2: & b \vee c \leftarrow & r_4: & d \leftarrow \mathit{not}\ b & r_6: & d \leftarrow c, \mathit{not}\ a \end{array} \right\}.$$

Let $U = \{c, d\}$. We have $H(r_1) \cap U = \{a, b\} \cap \{c, d\} = \emptyset$, that is, r_1 satisfies Condition 1. For r_5 and r_6, $B^+(r_5) \cap U = \{b, d\} \cap \{c, d\} \neq \emptyset$ and $B^+(r_6) \cap U = \{c\} \cap \{c, d\} \neq \emptyset$. Hence, both rules satisfy Condition 4. Furthermore, consider the interpretation $Y = \{b, c, d\}$. We have $H(r_2) \cap (Y \setminus U) = \{b, c\} \cap \{b\} \neq \emptyset$, thus r_2 satisfies Condition 2. Finally, for r_3 and r_4, $B^+(r_3) = \{a\} \not\subseteq \{b, c, d\} = Y$ and $B^-(r_4) \cap Y = \{b\} \cap \{b, c, d\} \neq \emptyset$, that is, both rules satisfy Condition 3. From the fact that each rule in P satisfies at least one of the unfoundedness conditions, we conclude that $U = \{c, d\}$ is unfounded for P with respect to $Y = \{b, c, d\}$. $\Diamond$

The basic relation between unfounded sets and answer sets is as follows.

Proposition 1 ([11]). *Let P be a program and Y an interpretation. Then, Y is an answer set of P iff $Y \models P$ and no nonempty subset of Y is unfounded for P with respect to Y.*

Example 3. Program P in Example 2 has two answer sets: $\{a, c, d\}$ and $\{b\}$. For the latter, we just have to check that $\{b\}$ is not unfounded for P with respect to $\{b\}$ itself, which holds in view of either rule r_1 or r_2. To verify via unfounded sets that $Y = \{a, c, d\}$ is an answer set of P, we have to check all nonempty subsets of Y. For instance, take $U = \{c, d\}$. We have already seen that r_1, r_5, and r_6 satisfy Condition 1 or 4, respectively; but the remaining rules r_2, r_3, and r_4 violate all four unfoundedness conditions for U with respect to Y. $\diamondsuit$

We next detail the relation between unfounded sets and models of logic programs as well as of their reducts. First, we have the following relationships between models and unfounded sets.

Lemma 1. *Let P be a program and Y an interpretation. Then, the following statements are equivalent: (a) $Y \models P$; (b) every set $U \subseteq \mathcal{U} \setminus Y$ is unfounded for P with respect to Y; and (c) every singleton $U \subseteq \mathcal{U} \setminus Y$ is unfounded for P with respect to Y.*

Proof. (*a*) $\Rightarrow$ (*b*): Assume that some set $U \subseteq \mathcal{U} \setminus Y$ is not unfounded for P with respect to Y. Then, for some rule $r \in P$, we have

(α) $H(r) \cap U \neq \emptyset$,
(β) $H(r) \cap (Y \setminus U) = \emptyset$,
(γ) $B^+(r) \subseteq Y$ and $B^-(r) \cap Y = \emptyset$, and
(δ) $B^+(r) \cap U = \emptyset$.

Since $U \cap Y = \emptyset$ by the assumption, we conclude from (β) that $H(r) \cap Y = \emptyset$. Since (γ) holds in addition, we have $Y \not\models r$ and thus $Y \not\models P$.

(*b*) $\Rightarrow$ (*c*) is trivial.

(*c*) $\Rightarrow$ (*a*): Assume $Y \not\models P$. Then, there is a rule $r \in P$ such that $Y \not\models r$, that is, $H(r) \cap Y = \emptyset$ and (γ) hold. By the definition of rules, $H(r) \neq \emptyset$. So, consider any $a \in H(r)$ and the singleton $U = \{a\}$. Clearly, (α) holds for r, and (β) holds by $H(r) \cap Y = \emptyset$. Finally, since $B^+(r) \subseteq Y$ and $a \notin Y$, (δ) holds as well. That is, there is a singleton $U \subseteq \mathcal{U} \setminus Y$ that is not unfounded for P with respect to Y. $\square$

We further describe the models of a program's reduct by unfounded sets.

Lemma 2. *Let P be a program, Y an interpretation such that $Y \models P$, and $U \subseteq \mathcal{U}$. Then, $(Y \setminus U) \models P^Y$ iff U is unfounded for P with respect to Y.*

Proof. ($\Rightarrow$) Assume that U is not unfounded for P with respect to Y. Then, for some rule $r \in P$, (α)–(δ) from the proof of Lemma 1 hold. Clearly, $B^-(r) \cap Y = \emptyset$ implies $(H(r) \leftarrow B^+(r)) \in P^Y$. From $B^+(r) \subseteq Y$ and (δ), we conclude $B^+(r) \subseteq (Y \setminus U)$. Together with ($\beta$), we obtain $(Y \setminus U) \not\models (H(r) \leftarrow B^+(r))$ and thus $(Y \setminus U) \not\models P^Y$.

($\Leftarrow$) Assume $(Y \setminus U) \not\models P^Y$. Then, there is a rule $r \in P$ such that $(Y \setminus U) \not\models \{r\}^Y$. We conclude that r satisfies (β), $B^+(r) \subseteq (Y \setminus U)$, and $B^-(r) \cap Y = \emptyset$. Since $B^+(r) \subseteq$

$(Y \setminus U)$ immediately implies $B^+(r) \subseteq Y$, (γ) holds. Moreover, $B^+(r) \subseteq (Y \setminus U)$ also implies (δ). It remains to show (α). From (γ) and $Y \models r$ (which holds by the assumption $Y \models P$), we conclude $H(r) \cap Y \neq \emptyset$. Together with (β), this implies (α). Since (α), (β), (γ), and (δ) jointly hold for some rule $r \in P$, we have that U is not unfounded for P with respect to Y. □

Example 4. For illustration, reconsider P from Example 2 and $Y = \{b, c, d\}$. For singleton $\{a\}$ and r_1, we have $H(r_1) \cap (Y \setminus \{a\}) = \{a, b\} \cap \{b, c, d\} \neq \emptyset$. Furthermore, $a \notin H(r)$ for all $r \in \{r_2, \ldots, r_6\}$. That is, $\{a\}$ is unfounded for P with respect to Y. From this, we can conclude by Lemma 1 that Y is a model of P, i.e., $Y \models P$.

As we have already seen in Example 2, $U = \{c, d\}$ is unfounded for P with respect to Y. Lemma 2 now tells us that $(Y \setminus U) = \{b\}$ is a model of $P^Y = \{r_1, r_2, r_3, r_5, (H(r_6) \leftarrow B^+(r_6))\}$. Moreover, one can check that $\{a, c, d\}$ is as well unfounded for P with respect to Y. ♢

The last observation in Example 4 stems from a more general side effect of Lemma 2: For any program P, any interpretation Y such that $Y \models P$, and $U \subseteq \mathcal{U}$, U is unfounded for P with respect to Y iff $(U \cap Y)$ is unfounded for P with respect to Y. For models Y, this allows us to restrict our attention to unfounded sets $U \subseteq Y$.

We are now in a position to state the following alternative characterizations of SE- and UE-models.

Theorem 1. *Let P be a program, Y an interpretation such that $Y \models P$, and $U \subseteq \mathcal{U}$. Then, $(Y \setminus U, Y)$ is an SE-model of P iff $(U \cap Y)$ is unfounded for P with respect to Y.*

Theorem 2. *Let P be a program, Y an interpretation such that $Y \models P$, and $U \subseteq \mathcal{U}$. Then, $(Y \setminus U, Y)$ is a UE-model of P iff $(U \cap Y)$ is unfounded for P with respect to Y and no nonempty proper subset of $(U \cap Y)$ is unfounded for P with respect to Y.*

Note that the inherent maximality criterion of UE-models is now reflected by a *minimality condition* on (nonempty) unfounded sets. Theorem 1 and 2 allow us to characterize strong and uniform equivalence in terms of unfounded sets, avoiding an explicit use of programs' reducts. Details will be discussed in Section 4.

Example 5. Recall programs $P = \{a \vee b \leftarrow\}$ and $Q = \{a \leftarrow \mathit{not}\, b;\ b \leftarrow \mathit{not}\, a\}$ from Example 1. We have seen that the only difference in their SE-models is the pair $(\emptyset, ab)$, which is an SE-model of Q, but not of P. Clearly, $Y = \{a, b\}$ is a classical model of P and of Q, and, in view of Theorem 1, we expect that Y is unfounded for Q with respect to Y, but not for P with respect to Y. The latter is easily checked since the rule $r = (a \vee b \leftarrow)$ yields (1) $H(r) \cap Y \neq \emptyset$; (2) $H(r) \cap (Y \setminus Y) = \emptyset$; (3) $B^+(r) \subseteq Y$ and $B^-(r) \cap Y = \emptyset$; and (4) $B^+(r) \cap Y = \emptyset$. Thus, none of the four unfoundedness conditions is met. However, for $r_1 = a \leftarrow \mathit{not}\, b$ and $r_2 = b \leftarrow \mathit{not}\, a$, we have $B^-(r_i) \cap Y \neq \emptyset$, for $i \in \{1, 2\}$, and thus Y is unfounded for Q with respect to Y.

Recall that $(\emptyset, ab)$ is not a UE-model of Q. In view of Theorem 2, we thus expect that $Y = \{a, b\}$ is not a minimal nonempty unfounded set. As one can check, both nonempty proper subsets $\{a\}$ and $\{b\}$ are in fact unfounded for Q with respect to Y. ♢

In the remainder of this section, we provide a further characterization of UE-models that makes use of elementary sets [5]. This not only gives us a more intrinsic characterization of the difference $U = (Y \setminus X)$ for a UE-model (X, Y) than that stated in Theorem 2, but also yields a further direct relation to loops. We make use of this fact and provide a new result for the UE-models of tight programs.

We define a nonempty set $U \subseteq \mathcal{U}$ as *elementary* for a program P if, for each V such that $\emptyset \subset V \subset U$, there is some $r \in P$ jointly satisfying

1. $H(r) \cap V \neq \emptyset$,
2. $H(r) \cap (U \setminus V) = \emptyset$,
3. $B^+(r) \cap V = \emptyset$, and
4. $B^+(r) \cap (U \setminus V) \neq \emptyset$.

Due to Conditions 1 and 4, every elementary set is also a loop of P, but the converse does not hold in general [5].

To link elementary sets and unfounded sets together, for a program P, an interpretation Y, and $U \subseteq \mathcal{U}$, we define:

$$P_{Y,U} = \{r \in P \mid H(r) \cap (Y \setminus U) = \emptyset,\ B^+(r) \subseteq Y,\ B^-(r) \cap Y = \emptyset\}.$$

Provided that $H(r) \cap U \neq \emptyset$, a rule $r \in P_{Y,U}$ supports U with respect to Y, while no rule in $(P \setminus P_{Y,U})$ supports U. Analogously to Gebser, Lee, and Lierler [5], we say that U is *elementarily unfounded* for P with respect to Y iff (i) U is unfounded for P with respect to Y and (ii) U is elementary for $P_{Y,U}$. Any elementarily unfounded set of P with respect to Y is also elementary for P, but an elementary set U that is unfounded for P with respect to Y is not necessarily elementarily unfounded because U might not be elementary for $P_{Y,U}$ [5].

Elementarily unfounded sets coincide with minimal nonempty unfounded sets.

Proposition 2 ([5]). *Let P be a program, Y an interpretation, and $U \subseteq \mathcal{U}$. Then, U is a minimal nonempty unfounded set of P with respect to Y iff U is elementarily unfounded for P with respect to Y.*

The fact that every nonempty unfounded set contains some elementarily unfounded set, which by definition is an elementary set, allows us to derive some properties of the difference $U = (Y \setminus X)$ for SE-interpretations (X, Y). For instance, we can exploit the fact that every elementary set is also a loop in the characterization of minimal nonempty unfounded sets, where the latter are only defined with respect to interpretations.

Formally, we derive the following properties for UE-models (resp., SE-models):

Corollary 1. *Let P be a program and (X, Y) a UE-model (resp., SE-model) of P. If $X \neq Y$, then $(Y \setminus X)$ is (resp., contains) (a) an elementarily unfounded set of P with respect to Y; (b) an elementary set of P; and (c) a loop of P.*

For tight programs, i.e., programs such that every loop is a singleton, we obtain the following property:

Corollary 2. *Let P be a tight program and (X, Y) an SE-model of P. Then, (X, Y) is a UE-model of P iff $X = Y$ or $(Y \setminus X)$ is a singleton that is unfounded for P with respect to Y.*

Example 6. Recall the SE-model $(\emptyset, ab)$ of $Q = \{a \leftarrow \mathit{not}\, b;\ b \leftarrow \mathit{not}\, a\}$. The loops of Q are $\{a\}$ and $\{b\}$; thus, Q is tight. This allows us to immediately conclude that $(\emptyset, ab)$ is not a UE-model of Q, without looking for any further SE-model to rebut it. $\Diamond$

The above result shows that, for tight programs, the structure of UE-models is particularly simple, viz. they are always of the form (Y, Y) or $(Y \setminus \{a\}, Y)$, for some $a \in Y$. As we will see in the next section, this also allows for simplified encodings.

4 Characterizations for Program Equivalence

In this section, we further exploit unfounded sets to characterize different notions of program equivalence. We start by comparing two programs, P and Q, regarding their unfounded sets for deriving conditions under which P and Q are ordinarily, strongly, and uniformly equivalent, respectively. Based on these conditions, we then provide novel encodings in propositional logic.

4.1 Characterizations based on Unfounded Sets

Two programs are ordinarily equivalent if they possess the same answer sets. As Proposition 1 shows, answer sets are precisely the models of a program that do not contain any nonempty unfounded set. Hence, ordinary equivalence can be described as follows:

Theorem 3. *Let P and Q be programs. Then, P and Q are ordinarily equivalent iff, for every interpretation Y, the following two conditions are equivalent*:

1. *$Y \models P$ and no nonempty subset of Y is unfounded for P with respect to Y;*
2. *$Y \models Q$ and no nonempty subset of Y is unfounded for Q with respect to Y.*

Note that ordinarily equivalent programs are not necessarily classically equivalent, as is for instance witnessed by programs $P = \{a \vee b \leftarrow\}$ and $Q = \{a \vee b \leftarrow;\ a \leftarrow c\}$ possessing the same answer sets: $\{a\}$ and $\{b\}$. However, $\{b, c\}$ is a model of P but not of Q. In turn, for strong and uniform equivalence, classical equivalence is a necessary (but, in general, not a sufficient) condition. This follows from the fact that every model of a program participates in at least one SE-model (resp., UE-model) and is thus relevant for testing strong (resp., uniform) equivalence. Indeed, the following characterization of strong equivalence considers all classical models.

Theorem 4. *Let P and Q be programs. Then, P and Q are strongly equivalent iff, for every interpretation Y such that $Y \models P$ or $Y \models Q$, P and Q possess the same unfounded sets with respect to Y.*

Proof. ($\Rightarrow$) Assume that P and Q are strongly equivalent. Fix any interpretation Y such that $Y \models P$ (or $Y \models Q$). Then, (Y, Y) is an SE-model of P (or Q), and since P and Q are strongly equivalent, (Y, Y) is also an SE-model of Q (or P). That is, both $Y \models P$ and $Y \models Q$ hold. Fix any set $U \subseteq \mathcal{U}$. By Lemma 2 and the fact that P and Q are strongly equivalent, U is unfounded for P with respect to Y iff $(Y \setminus U, Y)$ is an SE-model of P. But the latter holds iff $(Y \setminus U, Y)$ is an SE-model of Q, which in turn holds iff U is unfounded for Q with respect to Y.

($\Leftarrow$) Assume that P and Q are not strongly equivalent. Then, without loss of generality, there is an SE-model (X, Y) of P that is not an SE-model of Q (the other case is symmetric). By the definition of SE-models, we have $Y \models P$, and by Lemma 2, $(Y \setminus X)$ is unfounded for P with respect to Y, but either $Y \not\models Q$ or $(Y \setminus X)$ is not unfounded for Q with respect to Y. If $(Y \setminus X)$ is not unfounded for Q with respect to Y, then P and Q do not possess the same unfounded sets with respect to Y. Otherwise, if $Y \not\models Q$, by Lemma 1, there is a set $U \subseteq \mathcal{U} \setminus Y$ that is not unfounded for Q with respect to Y, but U is unfounded for P with respect to Y. □

Theorem 4 shows that strong equivalence focuses primarily on the unfounded sets admitted by the compared programs. In the setting of uniform equivalence, the consideration of unfounded sets is further restricted to minimal ones (cf. Theorem 2), and by Proposition 2, these are exactly the elementarily unfounded sets.

Theorem 5. *Let P and Q be programs. Then, P and Q are uniformly equivalent iff, for every interpretation Y such that $Y \models P$ or $Y \models Q$, P and Q possess the same elementarily unfounded sets with respect to Y.*

Proof. ($\Rightarrow$) Assume that P and Q are uniformly equivalent. Fix any interpretation Y such that $Y \models P$ (or $Y \models Q$). Then, (Y, Y) a UE-model of P (or Q), and since P and Q are uniformly equivalent, (Y, Y) is also a UE-model of Q (or P). That is, both $Y \models P$ and $Y \models Q$ hold. Fix any elementarily unfounded set U for P (or Q) with respect to Y. If $U \subseteq \mathcal{U} \setminus Y$, by Lemma 1 and Proposition 2, U is a singleton that is unfounded for both P and Q with respect to Y, which implies that U is elementarily unfounded for Q (or P) with respect to Y. Otherwise, if $U \cap Y \neq \emptyset$, then Lemma 1 and Proposition 2 imply $U \subseteq Y$. By Proposition 2 and Theorem 2, $(Y \setminus U, Y)$ is a UE-model of P (or Q), and since P and Q are uniformly equivalent, $(Y \setminus U, Y)$ is also a UE-model of Q (or P). Since $\emptyset \neq U \subseteq Y$, by Theorem 2 and Proposition 2, we conclude that U is elementarily unfounded for Q (or P) with respect to Y.

($\Leftarrow$) Assume that P and Q are not uniformly equivalent. Then, without loss of generality, there is a UE-model (X, Y) of P that is not a UE-model of Q (the other case is symmetric). Since (X, Y) is also an SE-model of P, we have $Y \models P$. If $Y \not\models Q$, by Lemma 1, there is a singleton $U \subseteq \mathcal{U} \setminus Y$ that is not unfounded for Q with respect to Y, but U is unfounded for P with respect to Y. That is, U is elementarily unfounded for P with respect to Y, but not for Q with respect to Y. Otherwise, if $Y \models Q$, (Y, Y) is a UE-model both of P and of Q. We conclude that $X \subset Y$, and by Theorem 2 and Proposition 2, $(Y \setminus X)$ is elementarily unfounded for P with respect to Y. Furthermore, the fact that (X, Y) is not a UE-model of Q, by Theorem 2 and Proposition 2, implies that $(Y \setminus X)$ is not elementarily unfounded for Q with respect to Y. □

In contrast to arbitrary unfounded sets, elementarily unfounded sets exhibit a certain structure as they are in fact loops or, even more accurately, elementary sets (cf. Corollary 1). By Theorem 5, such structures alone are material to uniform equivalence.

4.2 Characterizations in Propositional Logic

We now exploit the above results about unfounded sets to encode program equivalence in propositional logic. For ordinary equivalence, we use the well-known concept of loop formulas, while for strong and uniform equivalence we directly refer to unfounded sets.

In what follows, we write for a set of default literals, like $B(r)$, and a set of atoms, like $H(r)$, $B(r) \rightarrow H(r)$ as a shorthand for

$$\big(\bigwedge_{a \in B^+(r)} a \wedge \bigwedge_{a \in B^-(r)} \neg a\big) \rightarrow \bigvee_{a \in H(r)} a,$$

where, as usual, empty conjunctions (resp., disjunctions) are understood as $\top$ (resp., $\bot$). For instance, for a rule r of the form (1), $B(r) \rightarrow H(r)$ yields the implication

$$a_{k+1} \wedge \cdots \wedge a_m \wedge \neg a_{m+1} \wedge \cdots \wedge \neg a_n \rightarrow a_1 \vee \cdots \vee a_k .$$

Furthermore, within the subsequent encodings, an occurrence of a program P is understood as $\bigwedge_{r \in P}(B(r) \rightarrow H(r))$.

As a basis for the encodings, we use the following concept. Following Lee [7], for a program P and $U \subseteq \mathcal{U}$, the *external support formula* of U for P is

$$ES_P(U) = \bigvee_{r \in P, H(r) \cap U \neq \emptyset, B^+(r) \cap U = \emptyset} \neg\big(B(r) \rightarrow (H(r) \setminus U)\big). \qquad (2)$$

The relation between unfounded sets and external support formulas is as follows:

Lemma 3. *Let P be a program, Y an interpretation, and $U \subseteq \mathcal{U}$. Then, U is unfounded for P with respect to Y iff $Y \not\models ES_P(U)$.*

Proof. ($\Rightarrow$) Assume that $Y \models ES_P(U)$. Then, there is a rule $r \in P$ such that

(α) $H(r) \cap U \neq \emptyset$,
(β) $B^+(r) \cap U = \emptyset$,
(γ) $B^+(r) \subseteq Y$ and $B^-(r) \cap Y = \emptyset$, and
(δ) $(H(r) \setminus U) \cap Y = H(r) \cap (Y \setminus U) = \emptyset$.

That is, U is not unfounded for P with respect to Y.

($\Leftarrow$) Assume that U is not unfounded for P with respect to Y. Then, there is a rule $r \in P$ for which (α), (β), (γ), and (δ) hold. From (γ) and (δ), we conclude

$$Y \models \neg\big(B(r) \rightarrow (H(r) \setminus U)\big),$$

which together with (α) and (β) implies $Y \models ES_P(U)$. □

For a program P and $U \subseteq \mathcal{U}$, the (conjunctive) *loop formula* [7] of U for P is

$$LF_P(U) = \big(\bigwedge_{p \in U} p\big) \rightarrow ES_P(U).$$

With respect to an interpretation Y, the loop formula of U is violated if Y contains U as an unfounded set, otherwise, the loop formula of U is satisfied.

Proposition 3 ([7, 5]). *Let P be a program and Y an interpretation such that $Y \models P$. Then, the following statements are equivalent:*

(*a*) *Y is an answer set of P;*
(*b*) *$Y \models LF_P(U)$ for every nonempty subset U of $\mathcal{U}$;*
(*c*) *$Y \models LF_P(U)$ for every loop U of P;*
(*d*) *$Y \models LF_P(U)$ for every elementary set U of P.*

For ordinary equivalence, the following encodings (as well as different combinations thereof) can thus be obtained.

Theorem 6. *Let P and Q be programs. Let $\mathcal{L}$ and $\mathcal{E}$ denote the set of all loops and elementary sets, respectively, of P and Q. Then, the following statements are equivalent:*

(a) *P and Q are ordinarily equivalent;*
(b) $\big(P \wedge \bigwedge_{\emptyset \neq U \subseteq \mathcal{U}} LF_P(U)\big) \leftrightarrow \big(Q \wedge \bigwedge_{\emptyset \neq U \subseteq \mathcal{U}} LF_Q(U)\big)$ *is a tautology;*
(c) $\big(P \wedge \bigwedge_{U \in \mathcal{L}} LF_P(U)\big) \leftrightarrow \big(Q \wedge \bigwedge_{U \in \mathcal{L}} LF_Q(U)\big)$ *is a tautology;*
(d) $\big(P \wedge \bigwedge_{U \in \mathcal{E}} LF_P(U)\big) \leftrightarrow \big(Q \wedge \bigwedge_{U \in \mathcal{E}} LF_Q(U)\big)$ *is a tautology.*

Recall that, for tight programs, each loop (and thus, each elementary set) is a singleton. In this case, the encodings in (c) and (d) are thus polynomial in the size of the compared programs. Moreover, one can verify that they amount to checking whether the completions [12] of the compared programs are equivalent in classical logic.

For strong and uniform equivalence between P and Q, the models of P and Q along with the corresponding unfounded sets are compared, as Theorem 4 and 5 show. We thus directly consider external support formulas, rather than loop formulas.

Theorem 4 and Lemma 3 yield the following encoding for strong equivalence:

Theorem 7. *Let P and Q be programs. Then, P and Q are strongly equivalent iff* $\big(P \vee Q\big) \rightarrow \big(\bigwedge_{U \subseteq \mathcal{U}} \big(ES_P(U) \leftrightarrow ES_Q(U)\big)\big)$ *is a tautology.*

In order to also encode uniform equivalence, we have to single out elementarily unfounded sets. To this end, we modify the definition of the external support formula, $ES_P(U)$, and further encode the case that U is (not) a minimal nonempty unfounded set. For a program P and $U \subseteq \mathcal{U}$, we define the *minimality external support* formula as

$$ES^{\star}_P(U) = ES_P(U) \vee \neg\big(\bigwedge_{\emptyset \subset V \subset U} ES_P(V)\big).$$

Similar to external support formulas and unfounded sets, minimality external support formulas correspond to elementarily unfounded sets as follows.

Lemma 4. *Let P be a program, Y an interpretation, and $\emptyset \subset U \subseteq \mathcal{U}$. Then, U is elementarily unfounded for P with respect to Y iff $Y \not\models ES^{\star}_P(U)$.*

Proof. ($\Rightarrow$) Assume that $Y \models ES^{\star}_P(U)$. Then, one of the following two cases holds:

1. $Y \models ES_P(U)$: By Lemma 3, U is not unfounded for P with respect to Y, which implies that U is not elementarily unfounded for P with respect to Y.
2. $Y \not\models \big(\bigwedge_{\emptyset \subset V \subset U} ES_P(V)\big)$: For some V such that $\emptyset \subset V \subset U$, we have $Y \not\models ES_P(V)$. By Lemma 3, V is unfounded for P with respect to Y. We conclude that U is not a minimal nonempty unfounded set of P with respect to Y, and by Proposition 2, U is not elementarily unfounded for P with respect to Y.

($\Leftarrow$) Assume that $Y \not\models ES^{\star}_P(U)$. Then, $Y \not\models ES_P(U)$, and by Lemma 3, U is unfounded for P with respect to Y. Furthermore, $Y \models \big(\bigwedge_{\emptyset \subset V \subset U} ES_P(V)\big)$, and thus no set V such that $\emptyset \subset V \subset U$ is unfounded for P with respect to Y (again by Lemma 3). We conclude that U is a minimal nonempty unfounded set of P with respect to Y, and by Proposition 2, U is elementarily unfounded for P with respect to Y. $\square$

Theorem 5 and Lemma 4 allow us to encode uniform equivalence as follows.

Theorem 8. *Let P and Q be programs. Let $\mathcal{L}$ and $\mathcal{E}$ denote the set of all loops and elementary sets, respectively, of P and Q. Then, the following statements are equivalent:*

(*a*) *P and Q are uniformly equivalent;*
(*b*) $\big(P \vee Q\big) \rightarrow \big(\bigwedge_{U\subseteq\mathcal{U}} \big(ES^{\star}_{P}(U) \leftrightarrow ES^{\star}_{Q}(U)\big)\big)$ *is a tautology;*
(*c*) $\big(P \vee Q\big) \rightarrow \big(\bigwedge_{U\in\mathcal{L}} \big(ES^{\star}_{P}(U) \leftrightarrow ES^{\star}_{Q}(U)\big)\big)$ *is a tautology;*
(*d*) $\big(P \vee Q\big) \rightarrow \big(\bigwedge_{U\in\mathcal{E}} \big(ES^{\star}_{P}(U) \leftrightarrow ES^{\star}_{Q}(U)\big)\big)$ *is a tautology.*

Proof. By Theorem 5, P and Q are uniformly equivalent iff, for every interpretation Y such that $Y \models P$ or $Y \models Q$, P and Q possess the same elementarily unfounded sets with respect to Y. Clearly, any elementarily unfounded set of P or Q belongs to the set $\mathcal{E}$ of all elementary sets of P and Q, which is a subset of the set $\mathcal{L}$ of all loops of P and Q, and every element of $\mathcal{L}$ is a subset of $\mathcal{U}$. Furthermore, by Lemma 4, a set $\emptyset \subset U \subseteq \mathcal{U}$ is elementarily unfounded for P (resp., Q) with respect to Y iff $Y \not\models ES^{\star}_{P}(U)$ (resp., $Y \not\models ES^{\star}_{Q}(U)$). Finally, we have $ES^{\star}_{P}(\emptyset) \equiv ES^{\star}_{Q}(\emptyset) \equiv \bot$, so that $Y \models \big(ES^{\star}_{P}(\emptyset) \leftrightarrow ES^{\star}_{Q}(\emptyset)\big)$ for any interpretation Y. From this, the statement of Theorem 8 follows. $\square$

Again, we exploit the fact that, for tight programs, all loops and elementary sets are singletons. It is thus sufficient to consider only the external support formulas of singletons. To the best of our knowledge, this provides a novel technique to decide uniform equivalence between tight programs. Indeed, the following result is an immediate consequence of (c), or likewise (d), in Theorem 8.

Corollary 3. *Let P and Q be tight programs. Then, P and Q are uniformly equivalent iff $\big(P \vee Q\big) \rightarrow \big(\bigwedge_{a\in\mathcal{U}} \big(ES_{P}(\{a\}) \leftrightarrow ES_{Q}(\{a\})\big)\big)$ is a tautology.*

Indeed, for singletons $\{a\}$, $\neg\big(\bigwedge_{\emptyset\subset V\subset\{a\}} ES_{P}(V)\big)$ (resp., $\neg\big(\bigwedge_{\emptyset\subset V\subset\{a\}} ES_{Q}(V)\big)$) can be dropped from $ES^{\star}_{P}(\{a\})$ (resp., $ES^{\star}_{Q}(\{a\})$) because it is equivalent to $\bot$.

Except for ordinary and uniform equivalence between tight programs, all of the above encodings are of exponential size. As with the known encodings for answer sets, reproduced in Proposition 3, we do not suggest to *a priori* reduce the problem of deciding program equivalence to propositional logic. Rather, our encodings provide an alternative view on the conditions underlying program equivalence; similar characterizations have already been successfully exploited in answer-set solving [6, 13].

For strong equivalence, however, we can resolve the exponential number of conjuncts in Theorem 7 as follows. We use a copy $\mathcal{U}' = \{p' \mid p \in \mathcal{U}\}$ of the universe $\mathcal{U}$, where all p' are mutually distinct new atoms, and introduce a module representing $ES_{P}(U)$, as given in (2), but without explicitly referring to certain sets U; rather, a particular U is determined by the true atoms from the copy $\mathcal{U}'$ of $\mathcal{U}$. We define:

$$ES_{P} = \bigvee_{r\in P}\big(\bigvee_{p\in H(r)}p' \wedge \bigwedge_{p\in H(r)}(p' \vee \neg p) \wedge \bigwedge_{p\in B^{+}(r)}(p \wedge \neg p') \wedge \bigwedge_{p\in B^{-}(r)}\neg p\big).$$

Given a program P, for an interpretation Y (over $\mathcal{U}$) and $U \subseteq \mathcal{U}$, U is unfounded for P with respect to Y iff $(Y \cup \{p' \mid p \in U\}) \not\models ES_{P}$. This yields the following result:

Theorem 9. *Let P and Q be programs. Then, P and Q are strongly equivalent iff $(P \vee Q) \rightarrow (ES_{P} \leftrightarrow ES_{Q})$ is a tautology.*

5 Discussion

We provided novel characterizations for program equivalence in terms of unfounded sets, along with the related notions of loops and elementary sets. This allowed us to identify close relationships between these important concepts. While answer sets, and thus ordinary equivalence, rely on the absence of (nonempty) unfounded sets, we have shown that potential extensions of programs, captured by SE- and UE-models, can also be characterized directly by appeal to unfounded sets, thereby avoiding any reference to reducts of programs. We have seen that uniform equivalence is located in between ordinary and strong equivalence, in the sense that it considers all models, similar to strong equivalence, but only minimal (nonempty) unfounded sets, which are sufficient to decide whether a model is an answer set. This allowed us to develop particularly simple characterizations for uniform equivalence between tight programs.

References

1. Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2003)
2. Turner, H.: Strong equivalence made easy: Nested expressions and weight constraints. Theory and Practice of Logic Programming **3**(4-5) (2003) 602–622
3. Eiter, T., Fink, M.: Uniform equivalence of logic programs under the stable model semantics. In Palamidessi, C., ed.: Proceedings of the 19th International Conference on Logic Programming (ICLP'03), Springer-Verlag (2003) 224–238
4. Van Gelder, A., Ross, K., Schlipf, J.: The well-founded semantics for general logic programs. Journal of the ACM **38**(3) (1991) 620–650
5. Gebser, M., Lee, J., Lierler, Y.: Elementary sets for logic programs. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06), AAAI Press (2006)
6. Lin, F., Zhao, Y.: ASSAT: Computing answer sets of a logic program by SAT solvers. Artificial Intelligence **157**(1-2) (2004) 115–137
7. Lee, J.: A model-theoretic counterpart of loop formulas. In Kaelbling, L., Saffiotti, A., eds.: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05), Professional Book Center (2005) 503–508
8. Fages, F.: Consistency of Clark's completion and the existence of stable models. Journal of Methods of Logic in Computer Science **1** (1994) 51–60
9. Erdem, E., Lifschitz, V.: Tight logic programs. Theory and Practice of Logic Programming **3**(4-5) (2003) 499–518
10. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. ACM Transactions on Computational Logic **2**(4) (2001) 526–541
11. Leone, N., Rullo, P., Scarcello, F.: Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. Information and Computation **135**(2) (1997) 69–112
12. Clark, K.: Negation as failure. In Gallaire, H., Minker, J., eds.: Logic and Data Bases. Plenum Press (1978) 293–322
13. Giunchiglia, E., Lierler, Y., Maratea, M.: Answer set programming based on propositional satisfiability. Journal of Automated Reasoning **36**(4) (2006) 345–377