Simplifying A Logic Program Using Its Consequences

Abstract

1 Introduction

2 Preliminaries

In this paper, we consider only fully grounded finite disjunctive logic programs. A *(disjunctive) logic program* (DLP) is a finite set of (disjunctive) rules of the form

$$a_1 \vee \cdots \vee a_k \leftarrow a_{k+1}, \ldots, a_m, not\, a_{m+1}, \ldots, not\, a_n, \quad (1)$$

where $n \geq m \geq k \geq 0$, $n \geq 1$ and $a_1, \ldots, a_n$ are atoms. If $k \leq 1$, it is a *normal rule*. In particular, a *normal logic program* (NLP) is a finite set of normal rules.

We will also write rule r of form (1) as

$$head(r) \leftarrow body(r), \quad (2)$$

where $head(r)$ is $a_1 \vee \cdots \vee a_k$, $body(r) = body^+(r) \wedge body^-(r)$, $body^+(r)$ is $a_{k+1} \wedge \cdots \wedge a_m$, and $body^-(r)$ is $\neg a_{m+1} \wedge \cdots \wedge \neg a_n$, and we identify $head(r)$, $body^+(r)$, $body^-(r)$ with their corresponding sets of atoms. We denote $Atoms(r) = head(r) \cup body^+(r) \cup body^-(r)$. Let R be a set of rules, we denote $Atoms(R) = \bigcup_{r \in R} Atoms(r)$. Let L be a set of literals, we denote $\overline{L} = \{\neg p \mid p \in L\} \cup \{p \mid \neg p \in L\}$, $L^+ = \{p \mid p \in L\}$, and $L^- = \{p \mid \neg p \in L\}$.

A set S of atoms is said to *satisfy* a rule r, if $body^+(r) \subseteq S$ and $body^-(r) \cap S = \emptyset$ implies $head(r) \cap S \neq \emptyset$. S *satisfies* a program P, if S satisfies every rules in P. Let L be a set of literals and F a propositional formula, we write $L \models F$ if L entails F in the sense of classical logic.

2.1 Answer Set Semantics

The *answer sets* of a logic program are defined as in [Gelfond and Lifschitz, 1991]. Given a DLP P and a set S of atoms, the Gelfond-Lifschitz transformation of P on S, written P^S, is obtained from P by deleting:

1. each rule that has a formula $not\, p$ in its body with $p \in S$, and
2. all formulas of the form $not\, p$ in the bodies of the remaining rules.

A set S of atoms is an *answer set* of P if S is a minimal set satisfying P^S.

A *SE-interpretation* is a pair (X, Y) where X and Y are sets of atoms and $X \subseteq Y$. A SE-interpretation (X, Y) is a *SE-model* of a DLP P if Y satisfies P and X satisfies P^Y. A SE-model (Y, Y) of P is an *equilibrium model* of P, if there does not exist another set X such that $X \subset Y$ and (X, Y) is a SE-model of P.

Theorem 1 (Theorem 1 in [Ferraris, 2005]) *For any DLP, its equilibrium models are identical to its answer sets.*

Two DLPs P_1 and P_2 are *strongly equivalent*, if for any DLP P', programs $P_1 \cup P'$ and $P_2 \cup P'$ have the same set of answer sets.

Theorem 2 (Theorem 1 in [Lifschitz *et al.*, 2001]) *For any DLPs P_1 and P_2, P_1 is strongly equivalent to P_2 if and only if P_1 and P_2 have the same SE-models.*

2.2 External Support Rules and Loop Formulas

Here we briefly review the notions of external support rules and loop formulas in DLPs [Lee and Lifschitz, 2003].

Let E be a nonempty set of atoms. A rule r is an *external support* of E if $head(r) \cap E \neq \emptyset$ and $E \cap body^+(r) = \emptyset$. Given a DLP P, we use $R^-(E, P)$ to denote the set of external support rules of E in P.

The (conjunctive) *loop formula* of E under P, written $LF(E, P)$, is the following implication

$$\bigwedge_{p \in E} p \supset \bigvee_{r \in R^-(E,P)} \left(body(r) \wedge \bigwedge_{q \in head(r) \setminus E} \neg q \right).$$

A set S of atoms *satisfies* $LF(E, P)$, if $E \subseteq S$ implies that there exists a rule $r \in R^-(E, P)$ such that $body^+(r) \subseteq S$, $body^-(r) \cap S = \emptyset$, and $(head(r) \setminus E) \cap S = \emptyset$, *i.e.*, $S \cup \overline{Atoms(P) \setminus S} \models LF(E, P)$.

Theorem 3 (Theorem 1 in [Lee and Lifschitz, 2003]) *Let P be a logic program and S a set of atoms. If S satisfies P, then following conditions are equivalent:*

1. *S is an answer set of P;*
2. *S satisfies $LF(E, P)$ for every nonempty set E of atoms.*

2.3 Well-Founded Semantics

The notion of the well-founded model was proposed by Van Gelder *et al.* [1991] for NLPs. Later, Leone *et al.* [1997] extended the well-founded operator to DLPs. We briefly review the notions here.

Let P be a DLP and I a set of literals, a set X of atoms is an *unfounded set* of P *w.r.t.* I if for each atom $p \in X$ and each rule $r \in P$ such that $p \in head(r)$, at least one of the following conditions holds:

- $body(r) \cap \overline{I} \neq \emptyset$,
- $X \cap body^+(r) \neq \emptyset$, or
- $(head(r) \setminus X) \cap I \neq \emptyset$.

If P is an NLP, the union of two unfounded sets is also an unfounded set of P. However, the property does not hold for DLPs in general. Let P be a DLP and I a set of literals, I is *unfounded-free* if $I \cap X = \emptyset$ for each unfounded set X of P *w.r.t.* I. From [Leone *et al.*, 1997], if I is unfounded-free, then the union of two unfounded sets of P *w.r.t.* I is also an unfounded set, thus there exists the greatest unfounded set of P *w.r.t.* I. We use $U_P(I)$ to denote the greatest unfounded set of P *w.r.t.* I, if exists.

Now we define two operators for a DLP P *w.r.t.* a set I of literals:

$$T_P(I) = \{p \mid \text{there exists a rule } r \in P \textit{ s.t. } p \in head(r) \text{ and } I \models body(r) \wedge \bigwedge_{q \in head(r)\setminus\{p\}} \neg q\};$$

$$W_P(I) = T_P(I) \cup \overline{U_P(I)}.$$

T_P, U_P, and W_P are monotonic operators. We use $WFM(P)$ (resp. $T(P)$) to denote the least fixed point of the operator W_P (resp. T_P). If P is an NLP, then $WFM(P)$ is the *well-founded model* of P as defined in [Van Gelder *et al.*, 1991].

3 Simplifying A Logic Program

A *consequence* of a program is a consistent set of literals that are satisfied by every answer set of the program. The problem of simplifying a logic program considers how to simplify the program by a consequence, so that the answer sets of the program can be computed from the answer sets of the resulting program with the consequence. $WFM(P)$, the well-founded model if P is an NLP, is a consequence that is commonly used to simplify the program in the grounding engines, like lparse [Syrjänen, 2000] and gringo [Gebser *et al.*, 2007b], of modern ASP solvers, like smodels [Syrjänen and Niemelä, 2001], ASSAT [Lin and Zhao, 2004], cmodels [Lierler and Maratea, 2004], clasp [Gebser *et al.*, 2007a], claspD [Drescher *et al.*, 2008], and DLV [Leone *et al.*, 2002]. In this section, we consider how to extend the idea to other consequences of the program.

Let set L of literals be a consequence of a DLP P, we define $tr_n(P, L)$ to be the program obtained from P by

1. deleting each rule r that has an atom $p \in body^+(r)$ with $\neg p \in L$, and
2. replacing each rule r that has an atom $p \in head(r)$ or $p \in body^-(r)$ with $\neg p \in L$ by a rule r' such that $head(r') = head(r) \setminus L^-$, $body^+(r') = body^+(r)$, and $body^-(r') = body^-(r) \setminus L^-$.

We define $tr_p(P, L)$ to be the program obtained from P by

1. deleting each rule r that has an atom $p \in head(r)$ or $p \in body^-(r)$ with $p \in L$, and
2. replacing each rule r that has an atom $p \in body^+(r)$ with $p \in L$ by a rule r' such that $head(r') = head(r)$, $body^+(r') = body^+(r) \setminus L^+$, and $body^-(r') = body^-(r)$.

Note that $tr_n(P, L)$ (resp. $tr_p(P, L)$) does not contain any atoms in L^- (resp. L^+) and $tr_p(tr_n(P, L), L)$ does not contain any atoms occurring in L.

The following property explains why $WFM(P)$ can be used to simplify the program P.

Proposition 1 *Let P be a DLP and $L = WFM(P)$.*

- *Programs P and $tr_p(tr_n(P, L), L) \cup \{p \leftarrow \mid p \in L\}$ have the same set of answer sets.*
- *The program $P \cup \{\leftarrow p \mid \neg p \in L\}$ is strongly equivalent to the program*

  $$tr_p(tr_n(P, L), L) \cup \{p \leftarrow \mid p \in L\} \cup \{\leftarrow p \mid \neg p \in L\}.$$

When L is a general consequence of a DLP P, we show that $tr_n(P, L)$ could simplify P in general while $tr_p(P, L)$ could only simplify P for some special cases.

Proposition 2 *Let L be a consequence of a DLP P.*

- *A set S is an answer set of P iff S is an answer set of $tr_n(P, L)$.*
- *The program $P \cup \{\leftarrow p \mid \neg p \in L\}$ is strongly equivalent to the program $tr_n(P, L) \cup \{\leftarrow p \mid \neg p \in L\}$.*
- *If for every atom $p \in L$ there is a rule $p \leftarrow$ in P, then a set S is an answer set of P iff $S \setminus L$ is an answer set of $tr_p(P, L)$.*
- *A set S is an answer set of P implies S is an answer set of $tr_p(P, L) \cup \{p \leftarrow \mid p \in L\}$, but not vice versa in general.*

Intuitively, for any nonempty set E of atoms, the loop formula $LF(E, P)$ is equivalent to $LF(E, tr_n(P, L))$ under L, while $LF(E, P)$ only implies $LF(E, tr_p(P, L) \cup \{p \leftarrow \mid p \in L\})$ under L.

Example 1 *Consider the logic program P_1:*

$$a \leftarrow b, \quad c \leftarrow a, \quad b \leftarrow c, \quad c \leftarrow d, \quad a \leftarrow f,$$
$$d \leftarrow not\, e, \quad e \leftarrow not\, d, \quad \leftarrow not\, a, \quad f \leftarrow a.$$

$L = \{a, f\}$ is a consequence of P_1, then $tr_p(P_1, L)$ is:

$$c \leftarrow, \quad b \leftarrow c, \quad c \leftarrow d, \quad d \leftarrow not\, e, \quad e \leftarrow not\, d.$$

The only answer set of P_1 is $\{a, b, c, d, f\}$. However, $tr_p(P_1, L)$ has two answer sets: $\{b, c, d\}$ and $\{b, c, e\}$.

Given a consequence L of a program P, we can first simplify the program by $tr_n(P, L)$ which would no longer contain atoms in L^-. However, we cannot use $tr_p(P, L)$ to simplify P. In next sections, we identify some classes of sets U of atoms, such that P can be simplified by $tr_p(P, U)$, *i.e.*, the relations in Proposition 1 would be maintained respectively.

4 Reliable Sets

In this section, we propose the notion of the reliable set of a logic program. We show that, the reliable set can be used to simplify the program, and the notion is closely related to the well-founded operator. At last, we provide the computational complexity for identifying a reliable set.

Given a DLP P, a set U of atoms is said to be a *reliable set* of P, if for every nonempty subset E of U, there exists a rule $r \in R^-(E, P)$ such that $head(r) \subseteq E$ and $U \models body(r)$.

Notice that, $r \in R^-(E, P)$ and $U \models body(r)$ implies $body^+(r) \subseteq U \setminus E$ and $body^-(r) = \emptyset$. Then U is also a consequence of P.

Proposition 3 *If U is a reliable set of a DLP P, then U is a consequence of P.*

Theorem 4 *Let U be a reliable set of a DLP P. P is strongly equivalent to the program $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$.*

Proof Sketch: It is easy to verify that, there does not exist a SE-model (X, Y) of P such that $U \not\subseteq Y$ or $U \not\subseteq X$. Then both programs have the same set of SE-models. ■

From Theorem 4, we can simplify a logic program by its reliable sets. Moreover, the notion of the reliable set specifies a necessary and sufficient condition for the strongly equivalent relation between the following two programs.

Theorem 5 *Let U be a set of atoms of a DLP P. The program*

$$\{r \mid r \in P,\ head(r) \subseteq U\}$$

is strongly equivalent to the program

$$\{p \leftarrow \mid p \in U\}$$

if and only if U is a reliable set of P.

Proof Sketch: Let $P_1 = \{r \mid r \in P,\ head(r) \subseteq U\}$.

$\Leftarrow$: U is a reliable set of P implies U is a reliable set of P_1. The direction can be proved from Theorem 4.

$\Rightarrow$: Assume that U is not a reliable set of P, then there exists a nonempty subset E of U which prevents U to be a reliable set. It is easy to verify that $U \setminus E$ satisfies P_1^U, so $(U \setminus E, U)$ is a SE-model of P_1, which conflicts to the fact that $(U \setminus E, U)$ is not a SE-model of $\{p \leftarrow \mid p \in U\}$. ■

From the definition, the union of two reliable sets is still a reliable set of the program.

Proposition 4 *If U_1 and U_2 are reliable sets of a DLP P, then $U_1 \cup U_2$ is also a reliable set of P.*

Given a DLP P, there exists a greatest reliable set of P, denoted by $GRS(P)$, *i.e.*, the union of all possible reliable sets of P. We show that $GRS(P)$ is closely related to $T(P)$, *i.e.*, the least fixed point of the operator T_P.

Proposition 5 *Let P be an NLP. $GRS(P) = T(P)$.*

Proof Sketch: As there is a finite number of atoms in P, the result can be induced from the facts that U is a reliable set implies $T_P(U)$ is a reliable set, and if there is an atom $p \in U$, U is a reliable set, and $p \notin T(P)$, then $U \setminus \{p\} \not\subseteq T(P)$. ■

Corollary 6 *Let L be the well-founded model of an NLP P. $L^+ = GRS(tr_n(P, L))$.*

Proposition 6 *Let P be a DLP and $L = WFM(P)$.*

- *$T(P)$ is a reliable set of P.*
- *$L^+ \subseteq GRS(tr_n(P, L))$.*

When P is a DLP, it is possible that $GRS(P) \not\subseteq T(P)$.

Example 2 *Consider the DLP P_2:*

$$a \vee b \leftarrow, \qquad a \leftarrow b, \qquad b \leftarrow a.$$

$GRS(P_2) = \{a, b\}$ and $T(P_2) = \emptyset$.

From Proposition 5, $GRS(P)$ can be computed from $T(P)$ in polynomial time, when P is an NLP. However, it is coNP-hard, when P is a DLP.

Proposition 7 *Let P be a DLP and U a set of atoms.*

- *Deciding whether U is a reliable set of P is coNP-complete.*
- *Deciding whether U is equivalent to $GRS(P)$ is coNP-hard.*
- *Deciding whether an atom p is in $GRS(P)$ is coNP-hard.*

Proof Sketch: Deciding whether U is a reliable set of P is a coNP problem, as we can guess a nonempty subset E of U which prevents U to be a reliable set.

The hardness is proved by converting the SAT problem to the problem of deciding whether U is not a reliable set of P.

Let t and e be new atoms that are not appeared in a set $\mathcal{C}$ of clauses. We can construct a DLP P from $\mathcal{C}$ by:

- adding rules $t \vee e \vee \bigvee_{p \in Atoms(\mathcal{C})} p \leftarrow$ and $t \leftarrow e$,
- for each clause $C \in \mathcal{C}$, adding the rule $e \vee \bigvee_{\neg p \in C} p \leftarrow \bigwedge_{q \in C} q$, and
- for each atom $p \in Atoms(\mathcal{C})$, adding rules $t \leftarrow p$ and $p \leftarrow e$.

It can be verified that $U = Atoms(\mathcal{C}) \cup \{t, e\}$ is a reliable set of P if and only if $\mathcal{C}$ is not satisfiable.

Moreover, U is a reliable set of P iff $U = GRS(P)$ iff $e \in GRS(P)$. ■

In Section 6, we will provide an algorithm to compute $GRS(P)$ for a DLP P.

5 Strong and Weak Reliable Sets

The notion of the reliable set specifies a necessary and sufficient condition for the strongly equivalent relation between two specific programs. In this section, we propose notions of the strong reliable set and the weak reliable set, so that they respectively specify necessary and sufficient conditions for the strongly equivalent relation and the answer set equivalent relation between a program and its simplified result. We also provide computational complexity results for identifying a strong or a weak reliable set.

Given a DLP P, a set U of atoms is said to be a *strong reliable set* of P, if for every nonempty subset E of U and every SE-model (X, Y) of P, there exists a rule $r \in R^-(E, P)$ such that $head(r) \cap X \subseteq E$, $body^+(r) \subseteq X \cup U$, and $body^-(r) \cap (Y \cup U) = \emptyset$.

Proposition 8 *If U is a strong reliable set of a DLP P, then for every SE-model (X, Y) of P, $U \subseteq X$.*

Proof Sketch: From the definition, for every SE-model (X, Y) of P and every nonempty subset E of U, $U \setminus E \subseteq X$ implies $E \cap X \neq \emptyset$. $Atoms(P)$ is finite, then $U \subseteq X$. ■

Corollary 7 *If U is a strong reliable set of a DLP P, then U is a consequence of P.*

Theorem 8 *Let U be a set of atoms of a DLP P. P is strongly equivalent to $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$ if and only if U is a strong reliable set of P.*

Proof Sketch: $\Leftarrow$: From Proposition 8, for every SE-model (X, Y) of P, $U \subseteq X$. Then both programs have the same set of SE-models.

$\Rightarrow$: Assume that U is not a strong reliable set of P, then there exists a nonempty subset E of U and a SE-model (X, Y) of P that prevent U to be a strong reliable set. From the definition, $X \setminus E$ satisfies P^Y, so $(X \setminus E, Y)$ is also a SE-model of P, which conflicts to the fact that $(X \setminus E, Y)$ is not a SE-model of $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$. ■

So the notion of the strong reliable set specifies a necessary and sufficient condition for the strongly equivalent relation between a program and its simplified result.

Before defining the notion of the weak reliable set, we introduce another notion. Given a DLP P and a set U of atoms, a SE-model (X, Y) of P is called a *U-equilibrium model* of P, if $U \subseteq Y$, $Y \setminus U = X \setminus U$ and there does not exist another set X' such that $X' \setminus U \subset X \setminus U$ and $(X' \cup U, Y)$ is a SE-model of P.

Lemma 1 *Let P be a DLP and U a set of atoms. A SE-interpretation (X, Y) is a SE-model of $tr_p(P, U)$ if and only if there exists a SE-model (X^*, Y^*) of P such that $X^* = X \cup U$ and $Y^* = Y \cup U$.*

Proposition 9 *Let P be a DLP and U a set of atoms. A set S is an answer set of $tr_p(P, U)$ if and only if there exists a U-equilibrium model (X, Y) of P such that $X \setminus U = Y \setminus U = S$.*

Proof Sketch: $\Leftarrow$: (X, Y) is a U-equilibrium model of P, then $X \cup U = Y \cup U$ and $(X \cup U, Y \cup U)$ is a SE-model of P. So $(X \setminus U, Y \setminus U)$ is a SE-model of $tr_p(P, U)$. Meanwhile, there does not exist a set X' such that $X' \setminus U \subset X \setminus U$ and $(X' \cup U, Y \cup U)$ is a SE-model of P. From Lemma 1, $(X' \setminus U, Y \setminus U)$ is a SE-model of $tr_p(P, U)$. So $X \setminus U$ is an answer set of $tr_p(P, U)$.

$\Rightarrow$: S is answer set of $tr_p(P, U)$, then $(S \cup U, S \cup U)$ is a SE-model of P and there does not exists a set S' such that $S' \subset S$ and (S', S) is a SE-model of $tr_p(P, U)$. From Lemma 1, $(S' \cup U, S \cup U)$ is a SE-model of P. Then $(S \cup U, S \cup U)$ is a U-equilibrium model of P. ■

Given a DLP P, a set U of atoms is said to be a *weak reliable set* of P, if for every nonempty subset E of U and every U-equilibrium model (X, Y) of P, there exists a rule $r \in R^-(E, P)$ such that $head(r) \cap X \subseteq E$, $body^+(r) \subseteq X \cup U$, and $body^-(r) \cap (Y \cup U) = \emptyset$.

Similar to the proof of Proposition 8, we have the following proposition.

Proposition 10 *If U is a weak reliable set of a DLP P, then for every U-equilibrium model (X, Y) of P, $U \subseteq X$.*

Corollary 9 *If U is a weak reliable set of a DLP P, then U is a consequence of P.*

Theorem 10 *Let U be a set of atoms of a DLP P. Programs P and $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$ have the same set of answer sets if and only if U is a weak reliable set of P.*

Proof Sketch: $\Leftarrow$: From Proposition 10, for every U-equilibrium model (X, Y) of P, $U \subseteq X$, then $X = Y$. From Proposition 9, for every answer set S of $tr_p(P, U)$, there exists a U-equilibrium model $(S \cup U, S \cup U)$ of P, then $S \cup U$ is an answer set P. So programs P and $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$ have the same set of answer sets.

$\Rightarrow$: If U is not a weak reliable set of P, then there exists a U-equilibrium model (X, Y) of P and a nonempty subset E of U that prevent U to be a weak reliable set of P. Similar to the proof for Theorem 8, $(X \setminus E, Y)$ is also a U-equilibrium model of P. Then Y is not an answer set of P but Y is an answer set of $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$. ■

So the notion of the weak reliable set specifies a necessary and sufficient condition for the answer set equivalent relation between a program and its simplified result.

Proposition 11 *Let P be a DLP and U a set of atoms.*

- *U is a reliable set of P implies U is a strong reliable set of P.*
- *U is a strong reliable set of P implies U is a weak reliable set of P.*

Proposition 12 *If U_1 and U_2 are strong (resp. weak) reliable sets of a DLP P, then $U_1 \cup U_2$ is also a strong (resp. weak) reliable set of P.*

Given a DLP P, there exists a greatest strong (resp. weak) reliable set of P, denoted by $GSRS(P)$ (resp. $GWRS(P)$), *i.e.*, the union of all possible strong (resp. weak) reliable sets of P.

Proposition 13 *Let P be a DLP and U a set of atoms.*

- *Deciding whether U is a strong reliable set of P is coNP-complete.*
- *Deciding whether U is a weak reliable set of P is coNP-hard.*
- *Deciding whether U is equivalent to $GSRS(P)$ (resp. $GWRS(P)$) is coNP-hard.*
- *Deciding whether an atom p is in $GSRS(P)$ (resp. $GWRS(P)$) is coNP-hard.*

Proof Sketch: The first item is a coNP problem, as we can guess a nonempty subset E of U and a SE-model (X, Y) of P which prevents U to be a strong reliable set.

In the proof of Proposition 7, the set $U = Atoms(\mathcal{C}) \cup \{t, e\}$ is a reliable set iff it is a strong or a weak reliable set of P iff $U = GSRS(P)$ iff $U = GWRS(P)$. ■

Even when P is an NLP, it is still hard to identify a strong or a weak reliable set.

Proposition 14 *Let P be an NLP and U a set of atoms.*

- *Deciding whether U is a strong (resp. weak) reliable set of P is coNP-complete.*
- *Deciding whether U is equivalent to $GSRS(P)$ (resp. $GWRS(P)$) is coNP-hard.*
- *Deciding whether an atom p is in $GSRS(P)$ (resp. $GWRS(P)$) is coNP-hard.*

Proof Sketch: As P is an NLP, whether a SE-model (X, Y) is a U-equilibrium model can be checked in polynomial time. So deciding whether U is a weak reliable set of an NLP P is a coNP-problem.

The hardness is proved by converting the SAT problem to the problem of deciding whether U is not a strong reliable set of an NLP P.

Let e, t, and t' be new atoms that are not appeared in a set $\mathcal{C}$ of clauses. We can construct an DLP P from $\mathcal{C}$ by:

- adding rules $e \leftarrow t$, $t \leftarrow not\, t'$, $t' \leftarrow not\, t$, and $\leftarrow t', e$,
- for each atom $p \in Atoms(\mathcal{C})$, adding the rule $p \leftarrow e$, and
- for each clause $C \in \mathcal{C}$, adding the rule $e \leftarrow \bigwedge_{\neg p \in C} p \wedge \bigwedge_{q \in C} \neg q$.

It can be verified that $U = Atoms(\mathcal{C}) \cup \{e, t\}$ is a strong reliable set of P if and only if $\mathcal{C}$ is not satisfiable.

Moreover, U is a strong reliable set of P iff U is a weak reliable set of P iff $U = GSRS(P) = GWRS(P)$ iff $e \in GSRS(P)$. ■

6 Strong and Weak Reliable Sets Under A Consequence

As discussed in the previous section, notions of the strong and the weak reliable set can be used to simplify a logic program. They specify necessary and sufficient conditions for simplifying a logic program maintaining the strongly equivalent relation and the answer set equivalent relation respectively. However, they cannot be identified efficiently. In this section, we propose notions of the strong and the weak reliable set under a consequence of the program, which approximate the strong and the weak reliable set respectively. We show that, both notions can be used to simplify the program and can be computed efficiently.

Given a DLP P and a consistent set L of literals, a set U of atoms is said to be a *strong reliable set* of P under L, if for every nonempty subset E of $U \cup L^+$ with $E \cap U \neq \emptyset$, there exists a rule $r \in R^-(E, P)$ such that $head(r) \setminus L^- \subseteq E$ and $U \cup L \models body(r)$. U is said to be a *weak reliable set* of P under L, if for every nonempty subset E of U, there exists a rule $r \in R^-(E, P)$ such that $head(r) \setminus L^- \subseteq E$ and $U \cup L \models body(r)$. Note that, U is a reliable set of P iff U is a strong reliable set of P under $\emptyset$ iff U is a weak reliable set of P under $\emptyset$.

When L is a consequence of P, a strong and a weak reliable set of P under L is also a strong and a weak reliable set of the corresponding program, respectively.

Proposition 15 *Let P be a DLP, U a set of atoms, and L a consequence of P.*

- *U is a strong reliable set of P under L implies U is a strong reliable set of the program P':*

$$P \cup \{\leftarrow not\, p \mid p \in L\} \cup \{\leftarrow p \mid \neg p \in L\}.$$

- *U is a weak reliable set of P under L implies U is a weak reliable set of the program P'.*

Proof Sketch: (1) U is a strong reliable set of P under L. From the definition, for any SE-model (X, Y) of P with $L^+ \subseteq Y$ and $L^- \cap Y = \emptyset$, any nonempty subset E of U, and any subset E' of L^+, if $(U \cup L^+) \setminus (E \cup E') \subseteq X$, then $(E \cup E') \cap X \neq \emptyset$. We can induce from the case that $E \cup E' = U \cup L^+$, then there exists an atom $p \in E \cup E'$ such that $p \in X$. There is a finite number of atoms and E is a nonempty set, then $U \subseteq X$.

Moreover, for any nonempty subset E of U, there exists a rule $r \in R^-(E, P)$ such that $head(r) \cap X \subseteq E$, $body^+(r) \subseteq X \cup U$ and $body^-(r) \cap (Y \cup U) = \emptyset$. So U is a strong reliable set of P'.

(2) U is weak reliable set of P under L. For any U-equilibrium model (X, Y) of P', $L^+ \subseteq Y$, $L^- \cap Y = \emptyset$, $L^+ \setminus U \subseteq X$. From the definition, for any U-equilibrium model (X, Y) of P' and any nonempty subset E of U, if $U \setminus E \subseteq X$, then $E \cap X \neq \emptyset$. There is a finite number of atoms, then $U \subseteq X$. So there exists a rule $r \in R^-(E, P)$ such that $head(r) \cap X \subseteq E$, $body^+(r) \subseteq X \cup U$ and $body^-(r) \cap (Y \cup U) = \emptyset$. U is a weak reliable set of P'. ■

From Proposition 15, Theorem 8 and 10, the notions can be used to simplify a logic program.

Corollary 11 *Let P be a DLP, U a set of atoms, and L a consequence of P.*

- *If U is a strong reliable set of P under L, then the program*

$$P \cup \{\leftarrow not\, p \mid p \in L\} \cup \{\leftarrow p \mid \neg p \in L\}$$

is strongly equivalent to the program

$$tr_p(P, U) \cup \{p \leftarrow \mid p \in U\} \cup \{\leftarrow not\, p \mid p \in L\} \cup \{\leftarrow p \mid \neg p \in L\}.$$

- *If U is a weak reliable set of P under L, then programs P and $tr_p(P, U) \cup \{p \leftarrow \mid p \in U\}$ have the same set of answer sets.*

The strongly equivalent relation maintained by the notion of the strong reliable set of a program under a consistent set of literals, can be further specified by the following theorem.

Theorem 12 *Let P be a DLP, U a set of atoms, and L a consistent set of literals. The program*

$$\{r \mid r \in P,\ head(r) \setminus L^- \subseteq U \cup L^+\} \cup \{\leftarrow p \mid \neg p \in L\} \cup \{\leftarrow not\, p \mid p \in L\}$$

is strongly equivalent to the program

$$\{p \leftarrow \mid p \in U\} \cup \{\leftarrow p \mid \neg p \in L\} \cup \{\leftarrow not\, p \mid p \in L\}$$

if and only if U is a strong reliable set of P under L.

Proof Sketch: Let P_1, P_2 stand for these programs respectively.

$\Leftarrow$: U is a strong reliable set of P under L implies U is a strong reliable set of P_1 under L. The direction can be proved from Corollary 11.

$\Rightarrow$: Assume that U is not a strong reliable set of P under L, then there exists a nonempty set $E \subseteq U \cup L^+$ and $E \cap U \neq \emptyset$ which prevents U to be a strong reliable set of P under L. It can be verified that $(U \cup L^+) \setminus E$ satisfies $P_1^{U \cup L^+}$, so $((U \cup L^+) \setminus E, U \cup L^+)$ is a SE-model of P_1, which conflicts to the fact it is not a SE-model of P_2. ■

Proposition 16 *If U_1 and U_2 are strong (resp. weak) reliable sets of a DLP P under a consistent set L of literals, then $U_1 \cup U_2$ is also a strong (resp. weak) reliable set of P under L.*

Given a DLP P and a consistent set L of literals, there exists a greatest strong (resp. weak) reliable set of P under L. With a slight abuse of the notion, we denote it by $\mathit{GSRS}(P, L)$ (resp. $\mathit{GWRS}(P, L)$), *i.e.*, the union of all possible strong (resp. weak) reliable sets of P under L.

Proposition 17 *Let P be a DLP, U a set of atoms, and L a consequence of P.*

- *Deciding whether U is a strong (resp. weak) reliable set of P under L is coNP-complete.*
- *Deciding whether U is equivalent to $\mathit{GSRS}(P, L)$ (resp. $\mathit{GWRS}(P, L)$) is coNP-hard.*
- *Deciding whether an atom p is in $\mathit{GSRS}(P, L)$ (resp. $\mathit{GWRS}(P, L)$) is coNP-hard.*

Proof Sketch: The first item is a coNP problem, as we can guess a corresponding set E which prevents U to be a strong (resp. weak) reliable set of P under L.

In the proof of Proposition 7, the set $U = \mathit{Atoms}(\mathcal{C}) \cup \{t, e\}$ is a reliable set of P iff U is a strong or a weak reliable set of P under $\emptyset$ iff $U = \mathit{GSRS}(P, \emptyset)$ or $U = \mathit{GWRS}(P, \emptyset)$. ■

In the following, we provide approaches to compute $\mathit{GSRS}(P, L)$ and $\mathit{GWRS}(P, L)$. We show that, although $\mathit{GSRS}(P, L)$ and $\mathit{GWRS}(P, L)$ are hard to be computed in the worst case, approximates of them can be computed efficiently. Firstly, we provide polynomial time algorithms to compute $\mathit{GSRS}(P, L)$ and $\mathit{GWRS}(P, L)$ when P is an NLP.

Let P be a DLP, X a set of atoms and L a set of literals.

$$T^w_{P,L}(X) = \{p \mid \text{there exists a rule } r \in P \text{ such that } p \in \mathit{head}(r),\ p \notin \mathit{body}^+(r), \text{ and } L \cup X \models \mathit{body}(r) \wedge \bigwedge_{q \in \mathit{head}(r) \setminus \{p\}} \neg q\};$$

$$T^s_{P,L}(X) = \{p \mid \text{there exists a rule } r \in P \text{ such that } p \in \mathit{head}(r),\ p \notin \mathit{body}^+(r), \text{ and } (L \setminus L^+) \cup X \models \mathit{body}(r) \wedge \bigwedge_{q \in \mathit{head}(r) \setminus \{p\}} \neg q\}.$$

$T^w_{P,L}$ and $T^s_{P,L}$ are monotonic operators. We use $T^w(P, L)$ and $T^s(P, L)$ to denote the least fixed point of these operators respectively.

Proposition 18 *Let P be an NLP and L a consistent set of literals. $\mathit{GSRS}(P, L) = T^s(P, L)$ and $\mathit{GWRS}(P, L) = T^w(P, L)$.*

Proof Sketch: (1) If U is a strong (resp. weak) reliable set of P under L, then $T^s_{P,L}(U)$ (resp. $T^w_{P,L}(U)$) is also a strong (resp. weak) reliable set of P under L. So $T^s(P, L) \subseteq \mathit{GSRS}(P, L)$ and $T^w(P, L) \subseteq \mathit{GWRS}(P, L)$.

(2) If there is an atom $p \in U$, U is a weak reliable set of P under L, and $p \notin T^w(P, L)$, then $U \setminus \{p\} \not\subseteq T^w(P, L)$. So $\mathit{GWRS}(P, L) \subseteq T^w(P, L)$.

Let U be a strong reliable set of P under L and $E = (U \cup L^+) \setminus T^s(P, L)$. If $U \not\subseteq T^s(P, L)$ then $E \cap U \neq \emptyset$. From the definition, $(U \cup L^+) \setminus E \subseteq T^s(P, L)$ and there exists an atom p and a rule $r \in R^-(E, P)$ such that $\mathit{head}(r) = \{p\}$, $p \in E$ and $p \in T^s(P, L)$, which conflicts to the condition that $E \cap T^s(P, L) = \emptyset$. So $U \subseteq T^s(P, L)$ and $\mathit{GSRS}(P, L) \subseteq T^s(P, L)$. ■

Proposition 19 *Let P be a DLP and L a consistent set of literals. $T^s(P, L) \subseteq \mathit{GSRS}(P, L)$ and $T^w(P, L) \subseteq \mathit{GWRS}(P, L)$.*

We also provide approaches to compute $\mathit{GSRS}(P, L)$ and $\mathit{GWRS}(P, L)$, when P is a DLP.

Given a DLP P, a set X of atoms, and a set L of literals, the operator $\mathit{RS}^w_{P,L}(X)$ is defined in Algorithm 1. Similarly, the operator $\mathit{RS}^s_{P,L}(X)$ can be defined by replacing $X \cup L \models \mathit{body}(r)$ with $X \cup (L \setminus L^+) \models \mathit{body}(r)$ in Algorithm 1. Note that, both operators are monotonic. We use $\mathit{RS}^w(P, L)$ and $\mathit{RS}^s(P, L)$ to denote the least fixed point of the corresponding operators.

Algorithm 1: $\mathit{RS}^w_{P,L}(X)$

```
A := X;
H := {head(r) \ L⁻ | r ∈ P, head(r) ∩ body⁺(r) = ∅,
        X ∩ head(r) = ∅, and X ∪ L ⊨ body(r)};
for each C ∈ H do
    if |C| = 1 or
      for each p ∈ C, C ⊆ RS^w_{P,L}(X ∪ {p}) then
        A := A ∪ C;
return A
```

Proposition 20 *Let P be a DLP and L a consistent set of literals. $\mathit{GSRS}(P, L) = \mathit{RS}^s(P, L)$ and $\mathit{GWRS}(P, L) = \mathit{RS}^w(P, L)$.*

Proof Sketch: The proof is similar to the proof for Proposition 18. Additionally with the fact that, if $C \not\subseteq \mathit{RS}^w_{P,L}(X \cup \{p\})$ (resp. $\mathit{RS}^s_{P,L}(X \cup \{p\})$) for some $p \in C$, then for the set $E = C \setminus \{p\}$, there does not exist a corresponding rule $r \in R^-(E, P)$, which prevents

$C \cup X$ to be a weak (resp. strong) reliable set of P under L. ■

Note that, $RS^s(P, \emptyset) = RS^w(P, \emptyset) = GRS(P)$.

7 Experiments

8 Conclusion

References

[Drescher *et al.*, 2008] Christian Drescher, Martin Gebser, Torsten Grote, Benjamin Kaufmann, Arne König, Max Ostrowski, and Torsten Schaub. Conflict-driven disjunctive answer set solving. In *Proceedings of the 11th International Conference on Principles of Knowledge Respresentation and Reasoning (KR-08)*, pages 422–432, 2008.

[Ferraris, 2005] Paolo Ferraris. Answer sets for propositional theories. In *Proceedings of the 8th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-05), Diamante, Italy, September 5-8, 2005*, pages 119–131, 2005.

[Gebser *et al.*, 2007a] Martin Gebser, Benjamin Kaufmann, André Neumann, and Torsten Schaub. clasp: A conflict-driven answer set solver. In *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-07)*, pages 260–265. Springer, 2007.

[Gebser *et al.*, 2007b] Martin Gebser, Torsten Schaub, and Sven Thiele. Gringo: A new grounder for answer set programming. In *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-07)*, pages 266–271. Springer, 2007.

[Gelfond and Lifschitz, 1991] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New generation computing*, 9(3-4):365–385, 1991.

[Lee and Lifschitz, 2003] J. Lee and V. Lifschitz. Loop formulas for disjunctive logic programs. In *Proceedings of the 19th International Conference on Logic Programming (ICLP-03)*, pages 451–465, 2003.

[Leone *et al.*, 1997] Nicola Leone, Pasquale Rullo, and Francesco Scarcello. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Information and computation*, 135(2):69–112, 1997.

[Leone *et al.*, 2002] Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Francesco Calimeri, Tina Dell'Armi, Thomas Eiter, Georg Gottlob, Giovambattista Ianni, Giuseppe Ielpa, Christoph Koch, et al. The dlv system. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA-02)*, pages 537–540. Springer-Verlag, 2002.

[Lierler and Maratea, 2004] Yuliya Lierler and Marco Maratea. Cmodels-2: Sat-based answer set solver enhanced to non-tight programs. In *Proceedings of the 7th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-04)*, pages 346–350. Springer, 2004.

[Lifschitz *et al.*, 2001] Vladimir Lifschitz, David Pearce, and Agustín Valverde. Strongly equivalent logic programs. *ACM Transactions on Computational Logic (TOCL)*, 2(4):526–541, 2001.

[Lin and Zhao, 2004] Fangzhen Lin and Yuting Zhao. Assat: Computing answer sets of a logic program by sat solvers. *Artificial Intelligence*, 157(1):115–137, 2004.

[Syrjänen and Niemelä, 2001] Tommi Syrjänen and Ilkka Niemelä. The smodels system. In *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-01)*, pages 434–438. Springer, 2001.

[Syrjänen, 2000] Tommi Syrjänen. Lparse 1.0 user's manual. 2000.

[Van Gelder *et al.*, 1991] Allen Van Gelder, Kenneth A Ross, and John S Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM (JACM)*, 38(3):619–649, 1991.