

11-2011

On Elementary Loops of Logic Programs

Martin Gerber
Universitat Potsdam

Joohyung Lee
Arizona State University

Yuliya Lierler
University of Nebraska at Omaha, ylierler@unomaha.edu

Follow this and additional works at: <https://digitalcommons.unomaha.edu/compscifacpub>

 Part of the [Computer Sciences Commons](#)

Please take our feedback survey at: https://unomaha.az1.qualtrics.com/jfe/form/SV_8cchtFmpDyGfBLE

Recommended Citation

Gerber, Martin; Lee, Joohyung; and Lierler, Yuliya, "On Elementary Loops of Logic Programs" (2011).
Computer Science Faculty Publications. 3.
<https://digitalcommons.unomaha.edu/compscifacpub/3>

This Article is brought to you for free and open access by the Department of Computer Science at DigitalCommons@UNO. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of DigitalCommons@UNO. For more information, please contact unodigitalcommons@unomaha.edu.

On Elementary Loops of Logic Programs

Martin Gebser

*Institut für Informatik
Universität Potsdam, Germany
(e-mail: gebser@cs.uni-potsdam.de)*

Joohyung Lee

*School of Computing, Informatics and Decision Systems Engineering
Arizona State University, USA
(e-mail: joolee@asu.edu)*

Yuliya Lierler

*Department of Computer Science
University of Kentucky, USA
(e-mail: yuliya@cs.uky.edu)*

submitted 23 May 2010; revised ; accepted 18 August 2010

Note: To appear in *Theory and Practice of Logic Programming (TPLP)*

Abstract

Using the notion of an elementary loop, Gebser and Schaub refined the theorem on loop formulas due to Lin and Zhao by considering loop formulas of elementary loops only. In this article, we reformulate their definition of an elementary loop, extend it to disjunctive programs, and study several properties of elementary loops, including how maximal elementary loops are related to minimal unfounded sets. The results provide useful insights into the stable model semantics in terms of elementary loops. For a nondisjunctive program, using a graph-theoretic characterization of an elementary loop, we show that the problem of recognizing an elementary loop is tractable. On the other hand, we show that the corresponding problem is **coNP**-complete for a disjunctive program. Based on the notion of an elementary loop, we present the class of Head-Elementary-loop-Free (HEF) programs, which strictly generalizes the class of Head-Cycle-Free (HCF) programs due to Ben-Eliyahu and Dechter. Like an HCF program, an HEF program can be turned into an equivalent nondisjunctive program in polynomial time by shifting head atoms into the body.

KEYWORDS: stable model semantics, loop formulas, unfounded sets

1 Introduction

The theorem on loop formulas due to Lin and Zhao (2004) has contributed to understanding the relationship between the stable model semantics and classical logic. Unlike other translations that modify the vocabulary of a logic program (Ben-Eliyahu and Dechter 1994; Lin and Zhao 2003; Janhunen 2006), the original

theorem on loop formulas characterizes the stable models of a nondisjunctive program in terms of the models of its completion that satisfy the loop formulas of all loops of the program. This allows us to compute stable models using SAT solvers, which led to the design of answer set solvers ASSAT¹ (Lin and Zhao 2004) and CMODELS² (Giunchiglia et al. 2004). Due to its importance in semantic understanding as well as in stable model computation, the theorem on loop formulas has been extended to more general classes of logic programs, such as disjunctive programs (Lee and Lifschitz 2003), infinite programs and programs containing classical negation (Lee 2005), and programs containing aggregates (Liu and Truszczyński 2006; You and Liu 2008; Lee and Meng 2009). Moreover, it has been applied to other nonmonotonic logics, such as circumscription (Lee and Lin 2006) and nonmonotonic causal logic (Lee 2004). The stable model semantics for first-order formulas given in (Ferraris et al. 2007; Ferraris et al. 2011) is also closely related to the idea of loop formulas, as described in (Lee and Meng 2008).

By slightly modifying the definition of a loop, Lee (2005) showed that loop formulas can be viewed as a generalization of completion (Clark 1978). The model-theoretic account of loop formulas given in (Lee 2005) also tells us that the idea of loop formulas is closely related to assumption sets (Saccá and Zaniolo 1990) or unfounded sets (Leone et al. 1997). In a sense, the theorem by Lin and Zhao is an enhancement of the unfounded set based characterization of stable models given in (Saccá and Zaniolo 1990; Leone et al. 1997). The unfounded set based characterization takes into account the loop formulas of *all sets of atoms*, while the theorem by Lin and Zhao considers the loop formulas of *loops* only. Gebser and Schaub (2005) improved this enhancement even further. They defined the notion of an elementary loop of a nondisjunctive program and showed that the theorem by Lin and Zhao remains correct even if we consider loop formulas of *elementary loops* only.

In this article, we reformulate the definition of an elementary loop of a nondisjunctive program by Gebser and Schaub, extend it to disjunctive programs, and study several properties of elementary loops, including how maximal elementary loops are related to minimal unfounded sets. Based on the notion of an elementary loop, we present the class of Head-Elementary-loop-Free (HEF) program, which strictly generalizes the class of Head-Cycle-Free (HCF) programs due to Ben-Eliyahu and Dechter (1994). Like an HCF program, an HEF program can be turned into an equivalent nondisjunctive program in polynomial time by shifting head atoms into the body—a simple transformation defined in (Gelfond et al. 1991). This tells us that an HEF program is an “easy” disjunctive program, which is merely a syntactic variant of a nondisjunctive program. We also observe that several other properties of nondisjunctive and HCF programs can be generalized to HEF programs. The main results from (Lin and Zhao 2003) and (You et al. 2003), characterizing stable models in terms of *inherent tightness* and *weak tightness*, respectively, can be extended to HEF programs, and likewise the operational characterization of stable models of HCF programs due to Leone *et al.* (1997) can be extended to HEF pro-

¹ <http://assat.cs.ust.hk/>

² <http://www.cs.utexas.edu/users/tag/cmodels/>

grams. The properties of elementary loops and HEF programs studied here may be useful in improving the computation of disjunctive answer set solvers, such as CLASP³ (Drescher et al. 2008), Cmodels (Lierler 2005), DLV⁴ (Leone et al. 2006), and GNT⁵ (Janhunen et al. 2006).

The outline of this paper is as follows. In Section 2, we present our reformulated definition of an elementary loop of a nondisjunctive program and provide a corresponding refinement of the theorem on loop formulas, as well as some properties of elementary loops. These results are extended to disjunctive programs in Section 3. In Section 4, we introduce the class of HEF programs and show that their shifted variants preserve stable models. In Section 5, we generalize the notion of inherent tightness to HEF programs. An operational characterization of stable models of HEF programs is presented in Section 6. Finally, Section 7 concludes the paper.

This paper is an extended version of the conference papers (Gebser et al. 2006) and (Gebser et al. 2007).⁶

2 Nondisjunctive Programs

After providing the relevant background on nondisjunctive programs, this section introduces elementary loops of nondisjunctive programs. We further refine elementary loops to elementarily unfounded sets, yielding a syntactic characterization of minimal unfounded sets. Moreover, we show that elementary loops of nondisjunctive programs can be recognized in polynomial time. (The statements of the theorems and the propositions in this section which apply to nondisjunctive programs will be generalized to disjunctive programs or HEF programs in later sections and the proofs will be given there.) Finally, we compare our reformulation of elementary loops with the definition by Gebser and Schaub (2005).

2.1 Background

A *nondisjunctive rule* is an expression of the form

$$a_1 \leftarrow a_2, \dots, a_m, \text{not } a_{m+1}, \dots, \text{not } a_n \quad (1)$$

where $n \geq m \geq 1$ and a_1, \dots, a_n are propositional atoms. A *nondisjunctive program* is a finite set of nondisjunctive rules.

We will identify a nondisjunctive rule (1) with the propositional formula

$$a_2 \wedge \dots \wedge a_m \wedge \neg a_{m+1} \wedge \dots \wedge \neg a_n \rightarrow a_1, \quad (2)$$

and will often write (1) as

$$a_1 \leftarrow B, F \quad (3)$$

³ <http://potassco.sourceforge.net/>

⁴ <http://www.dbai.tuwien.ac.at/proj/dlv/>

⁵ <http://www.tcs.hut.fi/Software/gnt/>

⁶ In (Gebser et al. 2006; Gebser et al. 2007), the term “elementary set” was used in place of “elementary loop.”



Fig. 1. The dependency graph of Program Π_1

where B is a_2, \dots, a_m and F is *not* $a_{m+1}, \dots, \text{not } a_n$. We will sometimes identify B with its corresponding set of atoms.

We will identify an interpretation with the set of atoms that are true in it. We say that a set X of atoms satisfies a rule (1) if X satisfies (2). Moreover, X satisfies a nondisjunctive program Π (symbolically, $X \models \Pi$) if X satisfies every rule (1) of Π . If X satisfies Π , we also call X a *model* of Π .

The reduct Π^X of a nondisjunctive program Π w.r.t. a set X of atoms is obtained from Π by deleting each rule (3) such that $X \not\models F$, and replacing each remaining rule (3) with $a_1 \leftarrow B$. A set X of atoms is a *stable model*, also called an *answer set*, of Π if X is minimal among the sets of atoms that satisfy Π^X .

The (*positive*) *dependency graph* of a nondisjunctive program Π is the directed graph such that its vertices are the atoms occurring in Π , and its edges go from a_1 to a_2, \dots, a_m for all rules (1) of Π . A nonempty set Y of atoms is called a *loop* of Π if, for every pair a, b of atoms in Y , there is a path (possibly of length 0) from a to b in the dependency graph of Π such that all vertices in the path belong to Y . In other words, a nonempty set Y of atoms that occur in Π is a loop of Π if the subgraph of the dependency graph of Π induced by Y is strongly connected. It is clear that every singleton whose atom occurs in Π is a loop of Π .

For illustration, consider the following program Π_1 :

$$\begin{aligned} p &\leftarrow \text{not } s \\ p &\leftarrow r \\ q &\leftarrow r \\ r &\leftarrow p, q. \end{aligned}$$

Figure 1 shows the dependency graph of Π_1 . Program Π_1 has seven loops: $\{p\}$, $\{q\}$, $\{r\}$, $\{s\}$, $\{p, r\}$, $\{q, r\}$, and $\{p, q, r\}$.

For any set Y of atoms, the *external support formula* of Y for a nondisjunctive program Π , denoted by $ES_\Pi(Y)$, is the disjunction of conjunctions $B \wedge F$ for all rules (3) of Π such that $a_1 \in Y$ and $B \cap Y = \emptyset$. The first condition expresses that the atom “supported” by (3) is an element of Y . The second condition ensures that this support is “external”: the atoms in B that it relies on do not belong to Y . Thus, Y is called *externally supported* by Π w.r.t. a set X of atoms if $X \models ES_\Pi(Y)$.

For any set Y of atoms, by $LF_\Pi(Y)$, we denote the following formula:

$$\bigwedge_{a \in Y} a \rightarrow ES_\Pi(Y). \quad (4)$$

(The expression in the antecedent stands for the conjunction of all elements in Y .) Formula (4) is called the (*conjunctive*) *loop formula* of Y for Π . Note that we still call (4) a loop formula even when Y is not a loop of Π .

The following reformulation of the Lin-Zhao theorem, which characterizes the

stability of a model by loop formulas, is a part of the main theorem from (Lee 2005) for the nondisjunctive case.

Theorem 1 ((Lee 2005))

For any nondisjunctive program Π and any set X of atoms that occur in Π , if X is a model of Π , then the following conditions are equivalent:

- (a) X is a stable model of Π ;
- (b) X satisfies $LF_{\Pi}(Y)$ for all nonempty sets Y of atoms that occur in Π ;
- (c) X satisfies $LF_{\Pi}(Y)$ for all loops Y of Π .

According to the equivalence between conditions (a) and (b) in Theorem 1, a model of Π_1 is stable iff it satisfies the loop formulas of all fifteen nonempty sets of atoms formed from the atoms occurring in Π_1 . On the other hand, condition (c) tells us that it is sufficient to restrict attention to the following loop formulas of the seven loops of Π_1 :

$$\begin{aligned}
 p &\rightarrow \neg s \vee r \\
 q &\rightarrow r \\
 r &\rightarrow p \wedge q \\
 s &\rightarrow \perp \\
 p \wedge r &\rightarrow \neg s \\
 q \wedge r &\rightarrow \perp \\
 p \wedge q \wedge r &\rightarrow \neg s .
 \end{aligned} \tag{5}$$

Program Π_1 has six models: $\{p\}$, $\{s\}$, $\{p, s\}$, $\{q, s\}$, $\{p, q, r\}$, and $\{p, q, r, s\}$. Among them, $\{p\}$ is the only stable model of Π_1 , which is also the only model of Π_1 that satisfies all loop formulas in (5).

As noted in (Lee 2005), the equivalence between conditions (a) and (c) is a reformulation of the theorem by Lin and Zhao; the equivalence between conditions (a) and (b) is a reformulation of Corollary 2 from (Saccá and Zaniolo 1990) and Theorem 4.6 from (Leone et al. 1997) (in the nondisjunctive case), which characterizes the stability of a model in terms of unfounded sets. For any sets X, Y of atoms, we say that Y is *unfounded* by Π w.r.t. X if Y is not externally supported by Π w.r.t. X . Condition (b) can be stated in terms of unfounded sets as follows:

- (b') X contains no nonempty unfounded sets for Π w.r.t. X .

2.2 Elementary Loops of Nondisjunctive Programs

Gebser and Schaub (2005) showed that Y in $LF_{\Pi}(Y)$ in Theorem 1 can be restricted to “elementary” loops only. In this section, we present a reformulation of their definition of an elementary loop and investigate its properties. We compare our reformulation with the original definition by Gebser and Schaub in Section 2.5.

To begin with, the following proposition tells us that a loop can be defined without mentioning a dependency graph.

Proposition 1

For any nondisjunctive program Π and any nonempty set X of atoms that occur in Π , X is a loop of Π iff, for every nonempty proper subset Y of X , there is a rule (3) in Π such that $a_1 \in Y$ and $B \cap (X \setminus Y) \neq \emptyset$.

For any set X of atoms and any subset Y of X , we say that Y is *outbound* in X for a nondisjunctive program Π if there is a rule (3) in Π such that

- $a_1 \in Y$,
- $B \cap (X \setminus Y) \neq \emptyset$, and
- $B \cap Y = \emptyset$.

The following proposition describes the relationship between the external support formula of a set Y of atoms and the external support formula of a subset Z of Y that is not outbound in Y .

Proposition 2

For any nondisjunctive program Π and any sets X, Y, Z of atoms such that $Z \subseteq Y \subseteq X$, if Z is not outbound in Y for Π and $X \models ES_\Pi(Z)$, then $X \models ES_\Pi(Y)$.

Proposition 2 tells us that, in order to verify that a set Y of atoms is externally supported by Π w.r.t. a superset X of Y , it is sufficient to identify some externally supported subset of Y that is not outbound in Y for Π . Conversely, if Y is not externally supported by Π w.r.t. X , then every subset of Y that is externally supported by Π w.r.t. X is outbound in Y for Π .

For any nonempty set X of atoms that occur in Π , we say that X is an *elementary loop* of Π if all nonempty proper subsets of X are outbound in X for Π . As with loops, it is clear from the definition that every singleton whose atom occurs in Π is an elementary loop of Π . It is also clear that every elementary loop of Π is a loop of Π : the condition for being an elementary loop implies the condition for being a loop as stated in Proposition 1. On the other hand, a loop is not necessarily an elementary loop. For instance, one can check that $\{p, q, r\}$ is not an elementary loop of Π_1 since $\{p, r\}$ (or $\{q, r\}$) is not outbound in $\{p, q, r\}$ for Π_1 . All other loops of Π_1 are elementary loops. Note that an elementary loop may be a proper subset of another elementary loop (both $\{p\}$ and $\{p, r\}$ are elementary loops of Π_1).

The following program replaces the last rule of Π_1 with two other rules:

$$\begin{aligned}
 p &\leftarrow \text{not } s \\
 p &\leftarrow r \\
 q &\leftarrow r \\
 r &\leftarrow p \\
 r &\leftarrow q .
 \end{aligned} \tag{6}$$

The program has the same dependency graph as Π_1 , and hence has the same loops. However, its elementary loops are different from those of Π_1 : all its loops are elementary loops as well, including $\{p, q, r\}$.

The definition of an elementary loop X given above is not affected if we check the outboundness condition only for all loops or for all elementary loops that belong to X instead of all nonempty proper subsets of X .

Proposition 3

For any nondisjunctive program Π and any nonempty set X of atoms that occur in Π , X is an elementary loop of Π iff all proper subsets of X that are elementary loops of Π are outbound in X for Π .

The following proposition describes an important relationship between loop formulas of elementary loops and loop formulas of arbitrary sets of atoms.

Proposition 4

For any nondisjunctive program Π and any nonempty set Y of atoms that occur in Π , there is an elementary loop Z of Π such that Z is a subset of Y and $LF_{\Pi}(Z)$ entails $LF_{\Pi}(Y)$.

Proposition 4 allows us to limit attention to loop formulas of elementary loops only. This yields the following theorem, which is a reformulation of Theorem 3 from (Gebser and Schaub 2005).

Theorem 1 (d)

The following condition is equivalent to each of conditions (a)–(c) in Theorem 1:

- (d) X satisfies $LF_{\Pi}(Y)$ for all elementary loops Y of Π .

For instance, according to Theorem 1 (d), a model of Π_1 is stable iff it satisfies the first six loop formulas in (5); the loop formula of the non-elementary loop $\{p, q, r\}$ (the last one in (5)) can be disregarded.

2.3 Elementarily Unfounded Sets for Nondisjunctive Programs

If we modify condition (c) in Theorem 1 by replacing “loops” in its statement with “maximal loops,” the condition becomes weaker, and the modified statement of Theorem 1 is incorrect. For instance, Π_1 has only two maximal loops, $\{p, q, r\}$ and $\{s\}$, and their loop formulas are satisfied by a non-stable model $\{p, q, r\}$. In fact, maximal loop $\{p, q, r\}$ is not even an elementary loop of Π_1 . Similarly, it is not sufficient to consider maximal elementary loops only. If we replace “elementary loops” in the statement of Theorem 1 (d) with “maximal elementary loops,” then the modified statement is incorrect. For instance, the program

$$\begin{aligned} p &\leftarrow q, \text{ not } p \\ q &\leftarrow p, \text{ not } p \\ p &\leftarrow . \end{aligned}$$

has two models, $\{p\}$ and $\{p, q\}$, among which the latter is not stable. On the other hand, the only maximal elementary loop of the program is $\{p, q\}$, and its loop formula $p \wedge q \rightarrow \top$ is satisfied by both models, so that this loop formula alone is not sufficient to refute the stability of $\{p, q\}$. (Model $\{p, q\}$ does not satisfy the loop formula of $\{q\}$, which is $q \rightarrow p \wedge \neg p$.)

However, in the following we show that, if we consider the “relevant” part of a program w.r.t. a given interpretation, it is sufficient to restrict attention to maximal elementary loops. For any nondisjunctive program Π and any set X of atoms, by

Π_X , we denote the set of all rules (3) of Π such that $X \models B, F$. The following proposition states that all nonempty proper subsets of an elementary loop of Π_X are externally supported by Π w.r.t. X .

Proposition 5

For any nondisjunctive program Π , any set X of atoms, and any elementary loop Y of Π_X , X satisfies $ES_\Pi(Z)$ for all nonempty proper subsets Z of Y .

Proposition 5 tells us that any elementary loop Y of Π_X that is unfounded by Π w.r.t. X is maximal among the elementary loops of Π_X . From this, we obtain the following result.

Theorem 1 (e)

The following condition is equivalent to each of conditions (a)–(c) in Theorem 1:

- (e) X satisfies $LF_\Pi(Y)$ for every set Y of atoms such that Y is
- a maximal elementary loop of Π_X , or
 - a singleton whose atom occurs in Π .

Given a nondisjunctive program Π and a set X of atoms, we say that a set Y of atoms that occur in Π is *elementarily unfounded* by Π w.r.t. X if Y is

- an elementary loop of Π_X that is unfounded by Π w.r.t. X or
- a singleton that is unfounded by Π w.r.t. X .⁷

Proposition 5 tells us that any non-singleton elementarily unfounded set for Π w.r.t. X is a maximal elementary loop of Π_X .

It is clear from the definition that every elementarily unfounded set for Π w.r.t. X is an elementary loop of Π and that it is also unfounded by Π w.r.t. X . However, the converse does not hold in general. For instance, $\{p, q\}$ is an elementary loop that is unfounded by the program

$$\begin{aligned} p &\leftarrow q, \text{not } r \\ q &\leftarrow p, \text{not } r \end{aligned}$$

w.r.t. $\{p, q, r\}$, but $\{p, q\}$ is not an elementarily unfounded set w.r.t. $\{p, q, r\}$.

The following corollary, which follows from Proposition 5, states that all nonempty proper subsets of an elementarily unfounded set are externally supported. It is essentially a reformulation of Theorem 5 from (Gebser and Schaub 2005).

Corollary 1

For any nondisjunctive program Π , any set X of atoms, and any elementarily unfounded set Y for Π w.r.t. X , X does not satisfy $ES_\Pi(Y)$, but satisfies $ES_\Pi(Z)$ for all nonempty proper subsets Z of Y .

Corollary 1 tells us that elementarily unfounded sets form an “anti-chain”: one of them cannot be a proper subset of another. (On the other hand, an elementary loop may contain another elementary loop as its proper subset.) Also it tells us that elementarily unfounded sets are minimal among the nonempty unfounded sets occurring in Π . Interestingly, the converse also holds.

⁷ Elementarily unfounded sets are closely related to “active elementary loops” defined in (Gebser and Schaub 2005). We further investigate this relationship in Section 2.5.

Theorem 2

For any nondisjunctive program Π and any sets X, Y of atoms, Y is an elementarily unfounded set for Π w.r.t. X iff Y is minimal among the nonempty sets of atoms occurring in Π that are unfounded by Π w.r.t. X .

Notably, the correspondence between elementarily unfounded sets and minimal nonempty unfounded sets has also led to an alternative characterization of UE-models (Gebser et al. 2008), which characterizes uniform equivalence (Eiter and Fink 2003) of nondisjunctive programs as well as disjunctive programs.

Similar to Theorem 1 (b'), Theorem 1 (e) can be stated in terms of elementarily unfounded sets, thereby restricting attention to minimal nonempty unfounded sets.

Theorem 1 (e')

The following condition is equivalent to each of conditions (a)–(c) in Theorem 1:

(e') X contains no elementarily unfounded sets for Π w.r.t. X .

The notion of an elementarily unfounded set may help improve computation performed by SAT-based answer set solvers. Since there are exponentially many “relevant” loops in the worst case (Lifschitz and Razborov 2006), SAT-based answer set solvers do not add all loop formulas at once. Instead, they check whether a model returned by a SAT solver is stable. If not, a loop formula that is not satisfied by the model is added, and the SAT solver is invoked again. This process is repeated until a stable model is found or the search space is exhausted. In view of Theorem 1 (e'), it is sufficient to restrict attention to elementarily unfounded sets during the computation. This ensures that the considered loop formulas belong to elementary loops. Since every elementary loop is a loop, but not vice versa, the computation may involve fewer loop formulas overall than in the case when arbitrary loops are considered. However, whether this idea will lead to more efficient computation in practice requires further investigation.

2.4 Recognizing Elementary Loops of Nondisjunctive Programs

The definition of an elementary loop given in Section 2.2 involves considering all its nonempty proper subsets (or at least all elementary loops that are subsets). This may seem to imply that deciding whether a given set of atoms is an elementary loop is a computationally hard problem. However, Gebser and Schaub (2005) showed that this is not the case for nondisjunctive programs. They also noted that the notion of a positive dependency graph is not expressive enough to distinguish between elementary and non-elementary loops (Program Π_1 and the program in (6) have the same dependency graph, but their elementary loops are different), and instead used the rather complicated notion of a *body-head dependency graph* to identify elementary loops. In this section, we simplify their result by still referring to a positive dependency graph. We show that removing some *unnecessary* edges from a positive dependency graph is just enough to distinguish between elementary and non-elementary loops.



Fig. 2. The elementary subgraph of $\{p, q, r\}$ for Program Π_1

For any set X of atoms that occur in a nondisjunctive program Π , we define:

$$\begin{aligned}
 EC_{\Pi}^0(X) &= \emptyset, \\
 EC_{\Pi}^{i+1}(X) &= \{(a_1, b) \mid \text{there is a rule (3) in } \Pi \text{ such that } a_1 \in X, \\
 &\quad b \in B \cap X, \text{ and all atoms in } B \cap X \text{ belong to the} \\
 &\quad \text{same strongly connected component in } (X, EC_{\Pi}^i(X))\}, \\
 EC_{\Pi}(X) &= \bigcup_{i \geq 0} EC_{\Pi}^i(X).
 \end{aligned}$$

This is a “bottom-up” construction based on strongly connected components, i.e., maximal strongly connected subgraphs of a given graph. Thus $EC_{\Pi}^i(X)$ is a subset of $EC_{\Pi}^{i+1}(X)$, and the graph $(X, EC_{\Pi}(X))$ is a subgraph of the positive dependency graph of Π . We call the graph $(X, EC_{\Pi}(X))$ the *elementary subgraph* of X for Π . Figure 2 shows the elementary subgraph of $\{p, q, r\}$ for Π_1 , which is not strongly connected.

The following theorem is similar to Theorem 10 from (Gebser and Schaub 2005), but instead of referring to the notion of a body-head dependency graph, it refers to the notion of an elementary subgraph.

Theorem 3

For any nondisjunctive program Π and any nonempty set X of atoms that occur in Π , X is an elementary loop of Π iff the elementary subgraph of X for Π is strongly connected.

Since an elementary subgraph can be constructed in polynomial time, the problem of deciding whether a given set of atoms is an elementary loop of a nondisjunctive program is tractable.

2.5 Comparison with Gebser-Schaub Definition

In this section, we compare our reformulation of elementary loops with the original definition by Gebser and Schaub (2005) for nondisjunctive programs.

Let Π be a nondisjunctive program. A loop of Π is called *trivial* if it consists of a single atom such that the dependency graph of Π does not contain an edge from the atom to itself, and *nontrivial* otherwise.⁸ For a nontrivial loop L of Π , let

- $R_{\Pi}^{-}(L) = \{(3) \in \Pi \mid a_1 \in L, B \cap L = \emptyset\}$, and
- $R_{\Pi}^{+}(L) = \{(3) \in \Pi \mid a_1 \in L, B \cap L \neq \emptyset\}$.

Definition 1 ((Gebser and Schaub 2005))

A nontrivial loop L of a nondisjunctive program Π is called a *GS-elementary loop* of Π if $R_{\Pi}^{-}(L') \cap R_{\Pi}^{+}(L) \neq \emptyset$ for all proper subsets L' of L that are nontrivial loops of Π .

⁸ In (Lin and Zhao 2004) and (Gebser and Schaub 2005), loops were defined to be nontrivial.

Proposition 6

For any nondisjunctive program Π and any set L of atoms, L is a GS-elementary loop of Π iff L is a nontrivial elementary loop of Π .

Proof. *From left to right:* Assume that L is a GS-elementary loop of Π . If L is a singleton, it is a (nontrivial) elementary loop according to our definition. If L is not a singleton, we have that $R_{\Pi}^{-}(L') \cap R_{\Pi}^{+}(L) \neq \emptyset$ for any proper subset L' of L that is a nontrivial loop of Π . In other words, there is a rule (3) in Π such that

$$a_1 \in L', \quad (7)$$

$$B \cap L' = \emptyset, \quad (8)$$

and

$$B \cap (L \setminus L') \neq \emptyset. \quad (9)$$

We thus have that L' is outbound in L for Π . Furthermore, for any trivial loop $\{a_1\}$ of Π contained in L , there must be a rule (3) in Π such that $B \cap (L \setminus \{a_1\}) \neq \emptyset$, as L cannot be a loop of Π otherwise. Since $\{a_1\}$ is trivial, $B \cap \{a_1\} = \emptyset$, so that $\{a_1\}$ is outbound in L for Π . By Proposition 3, it follows that L is a (nontrivial) elementary loop of Π .

From right to left: Assume that L is a nontrivial elementary loop of Π . From the definition of an elementary loop, it follows that any proper subset L' of L that is a nontrivial loop of Π is outbound in L for Π . That is, there is a rule (3) in Π such that (7), (8), and (9) hold, so that L is a GS-elementary loop of Π . ■

For a nondisjunctive program Π and a set X of atoms, a loop L of Π_X is a GS-elementary loop of Π_X iff L is a nontrivial elementary loop of Π_X . Thus an *active elementary loop* of Π according to (Gebser and Schaub 2005) is a nontrivial elementary loop of Π_X that is unfounded by Π w.r.t. X . Hence, any active elementary loop L of Π is an elementarily unfounded set for Π w.r.t. X , while the converse does not hold in general if L is a singleton.

In fact, there are a few differences between Definition 1 and our definition of an elementary loop. First, our definition of an elementary loop does not a priori assume that its atoms form a loop. Rather, the fact that an elementary loop is a loop follows from its definition in view of Proposition 1. Second, the two definitions do not agree on trivial loops: a trivial loop is an elementary loop, but not a GS-elementary loop. This originates from the difference between the definition of a loop in (Lin and Zhao 2004) and its reformulation given in (Lee 2005). As shown in the main theorem from (Lee 2005), identifying a trivial loop as a loop admits a simpler reformulation of the Lin-Zhao theorem by allowing us to view completion formulas (Clark 1978) as a special case of loop formulas. Furthermore, the reformulated definition of an elementary loop enables us to identify a close relationship between maximal elementary loops (elementarily unfounded sets) and minimal nonempty unfounded sets.

Importantly, trivial loops allow us to extend the notion of an elementary loop to disjunctive programs without producing counterintuitive results. For instance,

consider the following disjunctive program:

$$\begin{aligned} p ; q &\leftarrow r \\ p ; r &\leftarrow q \\ q ; r &\leftarrow p . \end{aligned} \tag{10}$$

The nontrivial loops of this program are $\{p, q\}$, $\{p, r\}$, $\{q, r\}$, and $\{p, q, r\}$, but not the singletons $\{p\}$, $\{q\}$, and $\{r\}$. If we were to extend GS-elementary loops to disjunctive programs, a natural extension would say that $\{p, q, r\}$ is a GS-elementary loop since $\{p, q\}$, $\{p, r\}$, and $\{q, r\}$ are “outbound” in $\{p, q, r\}$. But note that $\{p, q, r\}$ is unfounded w.r.t. $\{p, q, r\}$; moreover, every singleton is also unfounded w.r.t. $\{p, q, r\}$. This is in contrast with Proposition 5, according to which all nonempty proper subsets of an elementary loop should be externally supported. The next section shows that such an anomaly does not arise with our definition of an elementary loop that is extended to disjunctive programs.

3 Disjunctive Programs

After providing the relevant background on disjunctive programs, this section generalizes the notions of an elementary loop and an elementarily unfounded set to disjunctive programs. We also provide the proofs of the generalizations of the statements given in the previous section; such generalized results also apply to the class of nondisjunctive programs as a fragment of disjunctive programs. Furthermore, we show that, in contrast to nondisjunctive programs, recognizing an elementary loop is intractable in the case of arbitrary disjunctive programs, but stays tractable under a certain syntactic condition.

3.1 Background

A *disjunctive rule* is an expression of the form

$$a_1; \dots; a_k \leftarrow a_{k+1}, \dots, a_l, \text{not } a_{l+1}, \dots, \text{not } a_m, \text{not not } a_{m+1}, \dots, \text{not not } a_n \tag{11}$$

where $n \geq m \geq l \geq k \geq 0$ and a_1, \dots, a_n are propositional atoms. A *disjunctive program* is a finite set of disjunctive rules. Note that any program with nested expressions can be turned into an equivalent program whose rules are of the form (11) (Lifschitz et al. 1999).

We will identify a disjunctive rule (11) with the propositional formula

$$a_{k+1} \wedge \dots \wedge a_l \wedge \neg a_{l+1} \wedge \dots \wedge \neg a_m \wedge \neg \neg a_{m+1} \wedge \dots \wedge \neg \neg a_n \rightarrow a_1 \vee \dots \vee a_k$$

and will often write (11) as

$$A \leftarrow B, F \tag{12}$$

where A is a_1, \dots, a_k , B is a_{k+1}, \dots, a_l , and F is

$$\text{not } a_{l+1}, \dots, \text{not } a_m, \text{not not } a_{m+1}, \dots, \text{not not } a_n .$$

We will sometimes identify A and B with their corresponding sets of atoms.

The reduct Π^X of a disjunctive program Π w.r.t. a set X of atoms is obtained from Π by deleting each rule (12) such that $X \not\models F$, and replacing each remaining rule (12) with $A \leftarrow B$. A set X of atoms is a *stable model*, also called an *answer set*, of Π if X is minimal among the sets of atoms that satisfy Π^X .

The definition of a (positive) *dependency graph* is extended to a disjunctive program Π in the straightforward way: the vertices of the graph are the atoms occurring in Π , and its edges go from the elements of A to the elements of B for all rules (12) of Π . With this extended definition of a dependency graph, the definition of a loop for a nondisjunctive program is straightforwardly extended to a disjunctive program.

For any set Y of atoms, the *external support formula* of Y for a disjunctive program Π , denoted by $ES_\Pi(Y)$, is the disjunction of conjunctions

$$B \wedge F \wedge \bigwedge_{a \in A \setminus Y} \neg a$$

for all rules (12) of Π such that $A \cap Y \neq \emptyset$ and $B \cap Y = \emptyset$. When Π is a nondisjunctive program, this definition reduces to the definition of $ES_\Pi(Y)$ for nondisjunctive programs given earlier. As before, we say that Y is *externally supported* by Π w.r.t. a set X of atoms if $X \models ES_\Pi(Y)$; Y is *unfounded* Π w.r.t. X if $X \not\models ES_\Pi(Y)$.

The notion of $LF_\Pi(Y)$ and the term (conjunctive) *loop formula* similarly apply to formulas (4) when Π is a disjunctive program.

As shown in (Lee 2005), Theorem 1 remains correct after replacing “nondisjunctive program” in its statement with “disjunctive program.”

Theorem^d 1 ((Lee 2005))

For any disjunctive program Π and any set X of atoms that occur in Π , if X is a model of Π , then the following conditions are equivalent:⁹

- (a) X is a stable model of Π ;
- (b) X satisfies $LF_\Pi(Y)$ for all nonempty sets Y of atoms that occur in Π ;
- (b') X contains no nonempty unfounded sets for Π w.r.t. X ;
- (c) X satisfies $LF_\Pi(Y)$ for all loops Y of Π .

For instance, the loop formulas of the seven loops of the program in (10) are:

$$\begin{aligned} p &\rightarrow (r \wedge \neg q) \vee (q \wedge \neg r) \\ q &\rightarrow (r \wedge \neg p) \vee (p \wedge \neg r) \\ r &\rightarrow (q \wedge \neg p) \vee (p \wedge \neg q) \\ p \wedge q &\rightarrow r \\ p \wedge r &\rightarrow q \\ q \wedge r &\rightarrow p \\ p \wedge q \wedge r &\rightarrow \perp. \end{aligned} \tag{13}$$

\emptyset is the only model of (13) and it is the only stable model of (10) in accordance with the equivalence between (a) and (c) in Theorem^d 1.

⁹ Superscript ^d indicates that the statement is a generalization to disjunctive programs.

3.2 Elementary Loops of Disjunctive Programs

In this section, we generalize the definition of an elementary loop to disjunctive programs.

A loop of a disjunctive program can be defined without referring to a dependency graph, as in Proposition 1.

Proposition^d 1

For any disjunctive program Π and any nonempty set X of atoms that occur in Π , X is a loop of Π iff, for every nonempty proper subset Y of X , there is a rule (12) in Π such that $A \cap Y \neq \emptyset$ and $B \cap (X \setminus Y) \neq \emptyset$.

Proof. *From left to right:* Assume that X is a loop of Π . If X is a singleton, it is clear. If X is not a singleton, take any nonempty proper subset Y of X . Since both Y and $X \setminus Y$ are nonempty, there is a path from some atom in Y to some atom in $X \setminus Y$ in the dependency graph of Π such that all vertices in the path belong to X . This implies that there is an edge from an atom in Y to an atom in $X \setminus Y$, i.e., $A \cap Y \neq \emptyset$ and $B \cap (X \setminus Y) \neq \emptyset$ for some rule (12) in Π .

From right to left: Assume that X is not a loop of Π . Then the subgraph of the dependency graph of Π induced by X is not strongly connected. Consequently, there is a nonempty proper subset Y of X such that no edge connects an atom in Y to an atom in $X \setminus Y$. This implies that there is no rule (12) in Π such that $A \cap Y \neq \emptyset$ and $B \cap (X \setminus Y) \neq \emptyset$. ■

For any set X of atoms and any subset Y of X , we say that Y is *outbound* in X for a disjunctive program Π if there is a rule (12) in Π such that

- $A \cap Y \neq \emptyset$,
- $B \cap (X \setminus Y) \neq \emptyset$,
- $A \cap (X \setminus Y) = \emptyset$, and
- $B \cap Y = \emptyset$.

As with nondisjunctive programs, for any nonempty set X of atoms that occur in Π , we say that X is an *elementary loop* of Π if all nonempty proper subsets of X are outbound in X for Π . Clearly, every singleton whose atom occurs in Π is an elementary loop of Π , and every elementary loop of Π is a loop of Π . The definition of an elementary loop of a disjunctive program is stronger than the alternative characterization of a loop provided in Proposition^d 1: it requires that the rule (12) satisfy two additional conditions, $A \cap (X \setminus Y) = \emptyset$ and $B \cap Y = \emptyset$.

To illustrate the definition of an elementary loop of a disjunctive program, consider the loop $\{p, q, r\}$ of the program in (10). The loop is not an elementary loop because, for instance, $\{p\}$ is not outbound in $\{p, q, r\}$: although the first two rules (12) in (10) are such that $A \cap \{p\} \neq \emptyset$, $B \cap \{q, r\} \neq \emptyset$, and $B \cap \{p\} = \emptyset$, we also have that $A \cap \{q, r\} \neq \emptyset$ for each of them. Similarly, $\{q\}$ and $\{r\}$ are not outbound in $\{p, q, r\}$. On the other hand, the remaining loops of the program, $\{p\}$, $\{q\}$, $\{r\}$, $\{p, q\}$, $\{p, r\}$, and $\{q, r\}$, are elementary loops.

With the extended definitions given above, Propositions 2, 3, 4 and Theorem 1 (d) remain correct after replacing “nondisjunctive program” in their statements with “disjunctive program.” In the following, we present proofs for these generalizations.

Proposition^d 2

For any disjunctive program Π and any sets X, Y, Z of atoms such that $Z \subseteq Y \subseteq X$, if Z is not outbound in Y for Π and $X \models ES_{\Pi}(Z)$, then $X \models ES_{\Pi}(Y)$.

Proof. Assume that Z is not outbound in Y for Π and that $X \models ES_{\Pi}(Z)$. From the latter, it follows that there is a rule (12) in Π such that

$$A \cap Z \neq \emptyset, \quad (14)$$

$$B \cap Z = \emptyset, \quad (15)$$

$$X \models B, F, \quad (16)$$

and

$$X \cap (A \setminus Z) = \emptyset. \quad (17)$$

From (14), since $Z \subseteq Y$,

$$A \cap Y \neq \emptyset. \quad (18)$$

From (17), since $Z \subseteq Y \subseteq X$,

$$X \cap (A \setminus Y) = \emptyset \quad (19)$$

and

$$Y \cap (A \setminus Z) = \emptyset,$$

where the latter is equivalent to

$$A \cap (Y \setminus Z) = \emptyset. \quad (20)$$

Since Z is not outbound in Y for Π , from (14), (15), and (20), it follows that

$$B \cap (Y \setminus Z) = \emptyset,$$

which, in combination with (15), gives us that

$$B \cap Y = \emptyset. \quad (21)$$

Finally, from (16), (18), (19), and (21), we conclude that $X \models ES_{\Pi}(Y)$. ■

Proposition^d 3

For any disjunctive program Π and any nonempty set X of atoms that occur in Π , X is an elementary loop of Π iff all proper subsets of X that are elementary loops of Π are outbound in X for Π .

Proof. From left to right is clear.

From right to left: Assume that X is not an elementary loop of Π . Then there is a nonempty proper subset Y of X that is not outbound in X for Π . If Y is an elementary loop of Π , it is clear. Otherwise, there is a nonempty proper subset Z of Y that is not outbound in Y for Π . For the sake of contradiction, assume that Z is outbound in X for Π , i.e., that there is a rule (12) in Π such that

$$A \cap Z \neq \emptyset, \quad (22)$$

$$B \cap (X \setminus Z) \neq \emptyset, \quad (23)$$

$$A \cap (X \setminus Z) = \emptyset, \quad (24)$$

and

$$B \cap Z = \emptyset. \quad (25)$$

From (22) and (24), since $Z \subseteq Y \subseteq X$,

$$A \cap Y \neq \emptyset, \quad (26)$$

$$A \cap (X \setminus Y) = \emptyset, \quad (27)$$

and

$$A \cap (Y \setminus Z) = \emptyset. \quad (28)$$

Since Z is not outbound in Y for Π , from (22), (25), and (28), it follows that

$$B \cap (Y \setminus Z) = \emptyset,$$

which, in combination with (23) and (25), gives us that

$$B \cap Y = \emptyset \quad (29)$$

and

$$B \cap (X \setminus Y) \neq \emptyset. \quad (30)$$

However, (26), (27), (29), and (30) together contradict that Y is not outbound in X for Π , from which we conclude that Z is not outbound in X for Π . We have thus shown that every nonempty proper subset of X that is not outbound in X for Π and not an elementary loop of Π contains in turn a nonempty proper subset that is not outbound in X for Π . Since X is finite, there is some (not necessarily unique) minimal nonempty proper subset of X that is not outbound in X for Π , and such a subset must be an elementary loop of Π . ■

Proposition^d 4

For any disjunctive program Π and any nonempty set Y of atoms that occur in Π , there is an elementary loop Z of Π such that Z is a subset of Y and $LF_\Pi(Z)$ entails $LF_\Pi(Y)$.

Proof. If Y is an elementary loop of Π , it is clear. Otherwise, by Proposition^d 3, some proper subset Z of Y is an elementary loop of Π that is not outbound in Y for Π . Take any set X of atoms such that $X \models LF_\Pi(Z)$. If $Y \not\subseteq X$, then $X \not\models \bigwedge_{a \in Y} a$ and $X \models LF_\Pi(Y)$. If $Y \subseteq X$, $X \models \bigwedge_{a \in Z} a$ and $X \models ES_\Pi(Z)$, and, by Proposition^d 2, we conclude that $X \models ES_\Pi(Y)$ and $X \models LF_\Pi(Y)$. ■

Theorem^d 1 (d)

The following condition is equivalent to each of conditions (a)–(c) in Theorem^d 1:

- (d) X satisfies $LF_\Pi(Y)$ for all elementary loops Y of Π .

Proof. We show the equivalence between (b) and (d). From (b) to (d) is clear, and from (d) to (b) follows immediately from Proposition^d 4. ■

For instance, for the program in (10), the loop formula of non-elementary loop $\{p, q, r\}$ (the last one in (13)) can be disregarded in view of Theorem^d 1 (d).

3.3 Elementarily Unfounded Sets for Disjunctive Programs

Let Π be a disjunctive program. For any sets X, Y of atoms, by $\Pi_{X,Y}$ we denote the set of all rules (12) of Π such that $X \models B, F$ and $X \cap (A \setminus Y) = \emptyset$. That is, the program $\Pi_{X,Y}$ contains all rules of Π that can provide supports for Y w.r.t. X . If $Y = X$, we also denote $\Pi_{X,Y}$ by Π_X . When Π is a nondisjunctive program, this definition reduces to the definition of Π_X for nondisjunctive programs given earlier. Furthermore, when Π is nondisjunctive and Y is not a singleton, then Y is an elementary loop of $\Pi_{X,Y}$ iff Y is an elementary loop of Π_X .

We extend the definition of an elementarily unfounded set to disjunctive programs by replacing “ Π_X ” with “ $\Pi_{X,Y}$ ”: for a disjunctive program Π and a set X of atoms, we say that a set Y of atoms that occur in Π is *elementarily unfounded* by Π w.r.t. X if Y is

- an elementary loop of $\Pi_{X,Y}$ that is unfounded by Π w.r.t. X or
- a singleton that is unfounded by Π w.r.t. X .

It is clear from the definition that every elementarily unfounded set for Π w.r.t. X is an elementary loop of Π and that it is also unfounded by Π w.r.t. X .

For instance, let Π be the program (10). The program $\Pi_{\{p,q,r\},\{p,q\}}$ consists of the first rule in (10), so that $\{p, q\}$ is not an elementary loop of $\Pi_{\{p,q,r\},\{p,q\}}$. On the other hand, $\Pi_{\{p,q\},\{p,q\}}$ consists of the last two rules in (10), and $\{p, q\}$ is an elementary loop of $\Pi_{\{p,q\},\{p,q\}}$. Since $\{p, q\}$ is also unfounded by Π w.r.t. $\{p, q\}$, it is an elementarily unfounded set for Π w.r.t. $\{p, q\}$.

Proposition 5, Corollary 1, and Theorem 2 remain correct after replacing “nondisjunctive program” in their statements with “disjunctive program,” and “ Π_X ” with “ $\Pi_{X,Y}$.”

Proposition^d 5

For any disjunctive program Π , any set X of atoms, and any elementary loop Y of $\Pi_{X,Y}$, X satisfies $ES_\Pi(Z)$ for all nonempty proper subsets Z of Y .

Proof. From the fact that Y is an elementary loop of $\Pi_{X,Y}$, it follows that any nonempty proper subset Z of Y is outbound in Y for $\Pi_{X,Y}$. If Y is not a singleton, this implies that Y is a subset of X and that, for each nonempty proper subset Z of Y , there is a rule (12) in Π such that

$$A \cap Z \neq \emptyset, \quad (31)$$

$$A \cap (Y \setminus Z) = \emptyset, \quad (32)$$

$$B \cap Z = \emptyset, \quad (33)$$

$$X \models B, F, \quad (34)$$

and

$$X \cap (A \setminus Y) = \emptyset . \quad (35)$$

From (32) and (35), it follows that

$$X \cap (A \setminus Z) = \emptyset . \quad (36)$$

Finally, from (31), (33), (34), and (36), we conclude that $X \models ES_{\Pi}(Z)$. ■

Corollary^d 1

For any disjunctive program Π , any set X of atoms, and any elementarily unfounded set Y for Π w.r.t. X , X does not satisfy $ES_{\Pi}(Y)$, but satisfies $ES_{\Pi}(Z)$ for all nonempty proper subsets Z of Y .

Proof. From the definition of an elementarily unfounded set, $X \not\models ES_{\Pi}(Y)$, and, by Proposition^d 5, $X \models ES_{\Pi}(Z)$ for all nonempty proper subsets Z of Y . ■

Theorem^d 2

For any disjunctive program Π and any sets X, Y of atoms, Y is an elementarily unfounded set for Π w.r.t. X iff Y is minimal among the nonempty sets of atoms occurring in Π that are unfounded by Π w.r.t. X .

Proof. From left to right follows immediately from Corollary^d 1.

From right to left: Assume that Y is minimal among the nonempty unfounded sets for Π w.r.t. X whose atoms occur in Π . If Y is a singleton, it is elementarily unfounded by Π w.r.t. X . Otherwise, if $Y \not\subseteq X$, there is an atom $a \in (Y \setminus X)$, and one can check that $(Y \setminus \{a\})$ is also unfounded by Π w.r.t. X , which contradicts that Y is a minimal nonempty unfounded set for Π w.r.t. X . Hence, from the minimality assumption on Y , it follows that Y is a subset of X . It also holds that $X \models ES_{\Pi}(Z)$ for every nonempty proper subset Z of Y , so that there is a rule (12) in Π such that

$$A \cap Z \neq \emptyset , \quad (37)$$

$$B \cap Z = \emptyset , \quad (38)$$

$$X \models B, F , \quad (39)$$

and

$$X \cap (A \setminus Z) = \emptyset . \quad (40)$$

From (40), since $Z \subseteq Y \subseteq X$,

$$A \cap (Y \setminus Z) = \emptyset \quad (41)$$

and

$$X \cap (A \setminus Y) = \emptyset . \quad (42)$$

Since Y is unfounded by Π w.r.t. X , from (37), (39), and (42), it follows that

$$B \cap Y \neq \emptyset ,$$

which, in combination with (38), gives us that

$$B \cap (Y \setminus Z) \neq \emptyset. \quad (43)$$

In view of (39) and (42), we have that the rule (12) is contained in $\Pi_{X,Y}$. From (37), (38), (41), and (43), we further conclude that Z is outbound in Y for $\Pi_{X,Y}$. Consequently, Y is an elementary loop of $\Pi_{X,Y}$ and elementarily unfounded by Π w.r.t. X . ■

Theorem 1 (e) and (e') can now be extended to disjunctive programs as follows.

Theorem^d 1 (e')

The following conditions are equivalent to each of conditions (a)–(c) in Theorem^d 1:

- (e) X satisfies $LF_{\Pi}(Y)$ for every set Y of atoms such that Y is
 - maximal among all sets Z of atoms that are elementary loops of $\Pi_{X,Z}$
 - or
 - a singleton whose atom occurs in Π ;
- (e') X contains no elementarily unfounded sets for Π w.r.t. X .

Proof. We first show the equivalence between (b') and (e'): from (b') to (e') is clear, and from (e') to (b') follows immediately from Theorem^d 2. Moreover, the equivalence between (e') and (e) holds in view of Proposition^d 5, which tells us that an elementarily unfounded set Y for Π w.r.t. X cannot be a proper subset of any set Z of atoms that is an elementary loop of $\Pi_{X,Z}$. ■

3.4 Recognizing Elementary Loops of Disjunctive Programs

Although deciding whether a given set of atoms is an elementary loop of a nondisjunctive program can be done efficiently, it turns out that the corresponding problem in the case of arbitrary disjunctive programs is intractable.

Theorem 4

For any disjunctive program Π and any set Y of atoms, deciding whether Y is an elementary loop of Π is **coNP**-complete.

Proof. Containment in **coNP** is clear, since it is easy to check that a given non-empty proper subset of Y is not outbound in Y for Π .

For **coNP**-hardness, we reduce the **coNP**-hard problem of deciding whether a finite set X of atoms is “unfounded-free” for a disjunctive program Π (Leone et al. 1997), i.e., X contains no nonempty unfounded sets for Π w.r.t. X . Using a new atom e that does not occur in Π or X , we construct a program Π' as follows: for every rule (12) of Π_X , include a rule $A \leftarrow e, B, F$ in Π' , and, for every $a \in X \cup \{e\}$, include a rule $e \leftarrow a$ in Π' . Given the rules of the latter type, it is clear that any proper subset Z of $Y = X \cup \{e\}$ that is not outbound in Y for Π' cannot contain e . For every rule (12) of Π' such that $A \neq \{e\}$, since $e \in B$, we then have that

$$B \cap (Y \setminus Z) \neq \emptyset.$$

Hence, if a nonempty proper subset Z of Y is not outbound in Y for Π' , for every rule (12) of Π' such that $A \neq \{e\}$, at least one of the following conditions holds:

$$A \cap Z = \emptyset, \quad (44)$$

$$A \cap (Y \setminus Z) \neq \emptyset, \quad (45)$$

or

$$B \cap Z \neq \emptyset. \quad (46)$$

Since $e \notin A$, (45) implies that

$$A \cap (X \setminus Z) \neq \emptyset. \quad (47)$$

We have thus shown that (44), (46), or (47) holds for every rule (12) of Π' such that $A \neq \{e\}$, and, similarly, for every rule (12) of Π_X . Furthermore, we have that

$$X \not\models B, F \quad (48)$$

for every rule (12) of $\Pi \setminus \Pi_X$. Consequently, (44), (46), (47), or (48) holds for every rule (12) of Π , which shows that Z is unfounded by Π w.r.t. X . Conversely, if a nonempty subset Z of X is unfounded by Π w.r.t. X , the fact that (44), (46), or (47) holds for every rule (12) of Π_X implies that every rule (12) of Π' satisfies (44), (45), or (46), so that Z is not outbound in Y for Π' . Consequently, we conclude that X is unfounded-free for Π iff $Y = X \cup \{e\}$ is an elementary loop of Π' . ■

However, for the class of disjunctive programs called “Head-Cycle-Free” (Ben-Eliyahu and Dechter 1994), deciding whether a set of atoms is an elementary loop is tractable. We say that a disjunctive program Π is *Head-Cycle-Free (HCF)* if $|A \cap Y| \leq 1$ for every rule (12) of Π and every loop Y of Π .

The definition of an *elementary subgraph* for a nondisjunctive program can be extended to disjunctive programs by modifying the equation for EC_{Π}^{i+1} as follows:

$$\begin{aligned} EC_{\Pi}^{i+1}(X) = \{ (a, b) \mid & \text{there is a rule (12) in } \Pi \text{ such that } A \cap X = \{a\}, \\ & b \in B \cap X, \text{ and all atoms in } B \cap X \text{ belong to the} \\ & \text{same strongly connected component in } (X, EC_{\Pi}^i(X)) \} . \end{aligned}$$

With this extended definition of an elementary subgraph, Theorem 3 remains correct after replacing “nondisjunctive program” in its statement with “HCF program.”

In the next section, we introduce “Head-Elementary-loop-Free” programs, and show that Theorem 3 can be further generalized to such programs.

4 Head-Elementary-Loop-Free Programs

In general, computing stable models of a disjunctive program is harder than computing stable models of a nondisjunctive program (Eiter and Gottlob 1995). On the other hand, HCF programs are “easy” disjunctive programs that can be turned into equivalent nondisjunctive programs in polynomial time (Ben-Eliyahu and Dechter 1994). This property plays an important role in the computation of stable models of disjunctive programs, and is used by answer set solvers CLASP, Cmodels, and DLV.

By referring to elementary loops in place of loops in the definition of an HCF program, we define a class of programs that is more general than HCF programs: we say that a disjunctive program Π is *Head-Elementary-loop-Free (HEF)* if $|A \cap Y| \leq 1$ for every rule (12) of Π and every elementary loop Y of Π . Since every elementary loop is also a loop, it is clear that every HCF program is an HEF program as well. However, not all HEF programs are HCF. For example, consider the following program Π_2 :

$$\begin{aligned} p &\leftarrow r \\ q &\leftarrow r \\ r &\leftarrow p, q \\ p ; q &\leftarrow . \end{aligned}$$

This program has six loops: $\{p\}$, $\{q\}$, $\{r\}$, $\{p, r\}$, $\{q, r\}$, and $\{p, q, r\}$. Since the head of the last rule contains two atoms from the loop $\{p, q, r\}$, Π_2 is not HCF. On the other hand, Π_2 is HEF since $\{p, q, r\}$ is not an elementary loop of Π_2 : its subsets $\{p, r\}$ and $\{q, r\}$ are not outbound in $\{p, q, r\}$ for Π_2 .

Let us write a rule (12) in the following form:

$$a_1; \dots; a_k \leftarrow B, F . \quad (49)$$

Gelfond *et al.* (1991) defined a mapping from a disjunctive program Π to a nondisjunctive program Π_{sh} , the *shifted variant* of Π , by replacing each rule (49) with $k > 1$ in Π by k new rules:

$$a_i \leftarrow B, F, \text{not } a_1, \dots, \text{not } a_{i-1}, \text{not } a_{i+1}, \dots, \text{not } a_k . \quad (50)$$

They showed that every stable model of Π_{sh} is also a stable model of Π . Although the converse does not hold in general, Ben-Eliyahu and Dechter (1994) showed that the converse holds if Π is HCF. We below extend this result to HEF programs.

The following proposition compares the elementary loops of Π with the elementary loops of Π_{sh} .

Proposition 7

For any disjunctive program Π , if X is an elementary loop of Π , then X is an elementary loop of Π_{sh} .

Proof. Assume that X is an elementary loop of Π . Then every nonempty proper subset Y of X is outbound in X for Π , so that there is a rule (49) in Π such that

$$\{a_1, \dots, a_k\} \cap Y \neq \emptyset , \quad (51)$$

$$B \cap (X \setminus Y) \neq \emptyset , \quad (52)$$

$$\{a_1, \dots, a_k\} \cap (X \setminus Y) = \emptyset ,$$

and

$$B \cap Y = \emptyset . \quad (53)$$

For some $a_i \in \{a_1, \dots, a_k\} \cap Y$, (51) implies that some rule (50) in Π_{sh} satisfies

$$\{a_i\} \cap Y \neq \emptyset$$

and

$$\{a_i\} \cap (X \setminus Y) = \emptyset .$$

Together with (52) and (53), this means that Y is outbound in X for Π_{sh} . Consequently, X is an elementary loop of Π_{sh} . ■

The converse of Proposition 7 does not hold even if Π is HEF. For example, consider the following HEF program Π_3 :

$$\begin{aligned} p &; q \leftarrow r \\ r &\leftarrow p \\ r &\leftarrow q . \end{aligned}$$

Set $\{p, q, r\}$ is not an elementary loop of Π_3 since, for instance, $\{p\}$ is not outbound in $\{p, q, r\}$ for Π_3 . On the other hand, $\{p, q, r\}$ is an elementary loop of $(\Pi_3)_{sh}$:

$$\begin{aligned} p &\leftarrow r, \text{ not } q \\ q &\leftarrow r, \text{ not } p \\ r &\leftarrow p \\ r &\leftarrow q . \end{aligned} \tag{54}$$

However, the following proposition shows that there is a certain subset of Π_{sh} whose elementary loops are also elementary loops of Π .

Proposition 8

For any disjunctive program Π , any set X of atoms, and any subset Y of X , if Y is an elementary loop of $(\Pi_{sh})_X$, then Y is an elementary loop of Π .

Proof. Assume that Y is an elementary loop of $(\Pi_{sh})_X$, and not an elementary loop of Π for the sake of contradiction. Consider any rule (49) in Π , and any proper subset Z of Y . Since Y is not an elementary loop of Π , at least one of the following conditions holds:

$$\begin{aligned} \{a_1, \dots, a_k\} \cap Z &= \emptyset , \\ \{a_1, \dots, a_k\} \cap (Y \setminus Z) &\neq \emptyset , \\ B \cap Z &\neq \emptyset , \text{ or} \\ B \cap (Y \setminus Z) &= \emptyset . \end{aligned} \tag{55}$$

We will show that any rule (50) in $(\Pi_{sh})_X$ obtained from (49) by shifting satisfies at least one of the following conditions:

$$\begin{aligned} \{a_i\} \cap Z &= \emptyset , \\ \{a_i\} \cap (Y \setminus Z) &\neq \emptyset , \\ B \cap Z &\neq \emptyset , \text{ or} \\ B \cap (Y \setminus Z) &= \emptyset . \end{aligned}$$

This contradicts the assumption that Y is an elementary loop of $(\Pi_{sh})_X$.

Case 1: The first, the third, or the fourth condition of (55) holds. The claim trivially follows.

Case 2: $\{a_1, \dots, a_k\} \cap (Y \setminus Z) \neq \emptyset$. Recall that

$$X \models B, F, \text{ not } a_1, \dots, \text{ not } a_{i-1}, \text{ not } a_{i+1}, \text{ not } a_k,$$

by $(\Pi_{sh})_X$ construction. It follows that $|\{a_1, \dots, a_k\} \cap X| \leq 1$. From the fact that $Y \subseteq X$ and $Z \subset Y$ we conclude that $|\{a_1, \dots, a_k\} \cap Y| \leq 1$ and $\{a_1, \dots, a_k\} \cap Z = \emptyset$, so that $\{a_i\} \cap Z = \emptyset$. ■

For instance, for $X = \{p, q, r\}$ and $(\Pi_3)_{sh}$, we have that $[(\Pi_3)_{sh}]_X$ consists of the last two rules in (54). Only the singletons $\{p\}$, $\{q\}$, and $\{r\}$ are elementary loops of $[(\Pi_3)_{sh}]_X$, and clearly they are elementary loops of Π_3 as well.

We are now ready to show the equivalence between an HEF program and its shifted variant.

Theorem 5

For any HEF program Π and any set X of atoms, X is a stable model of Π iff X is a stable model of Π_{sh} .

Proof. *From left to right:* Assume that X is a stable model of Π . Then X is a model of Π_{sh} such that all its atoms occur in Π_{sh} and also in $[\Pi_{sh}]_X$. Furthermore, by Theorem^d 1 (d), we have that X satisfies $LF_{\Pi}(Y)$ for all elementary loops Y of Π . By Proposition 8, the elementary loops of Π include all elementary loops Y of $[\Pi_{sh}]_X$ that are contained in X , and, since Π is HEF, it holds that $ES_{\Pi_{sh}}(Y)$ and $ES_{\Pi}(Y)$ as well as $LF_{\Pi_{sh}}(Y)$ and $LF_{\Pi}(Y)$ are equivalent to each other. This implies that X satisfies $ES_{\Pi_{sh}}(Y)$ for all elementary loops Y of $[\Pi_{sh}]_X$ that are contained in X , so that X contains no elementarily unfounded sets for Π_{sh} w.r.t. X . By Theorem^d 1 (e'), we conclude that X is a stable model of Π_{sh} .

From right to left: Assume that X is a stable model of Π_{sh} . Then X is a model of Π such that all its atoms occur in Π . Furthermore, by Theorem^d 1 (d), we have that X satisfies $LF_{\Pi_{sh}}(Y)$ for all elementary loops Y of Π_{sh} . By Proposition 7, the elementary loops of Π_{sh} include all elementary loops Y of Π , and, since Π is HEF, it holds that $ES_{\Pi}(Y)$ and $ES_{\Pi_{sh}}(Y)$ as well as $LF_{\Pi}(Y)$ and $LF_{\Pi_{sh}}(Y)$ are equivalent to each other. This implies that X satisfies $LF_{\Pi}(Y)$ for all elementary loops Y of Π . By Theorem^d 1 (d), we conclude that X is a stable model of Π . ■

For instance, one can check that both Π_2 and $(\Pi_2)_{sh}$ have $\{p\}$ and $\{q\}$ as their stable models. It follows that HEF programs are not more expressive than nondisjunctive programs, so that one can regard the use of disjunctive rules in such programs as a syntactic variant. Furthermore, the problem of deciding whether a model is stable for an HEF program is tractable, just as the same problem for a nondisjunctive program. (In the case of arbitrary disjunctive programs, it is **coNP**-complete (Eiter and Gottlob 1995).) These properties were known for HCF programs, and here we extended them to HEF programs.

In Section 3.4, we defined the notion of an elementary subgraph of a set X of atoms for a disjunctive program Π . Theorem 3 still applies to HEF programs.

Theorem^{hef} 3

For any HEF program Π and any nonempty set X of atoms that occur in Π , X is an elementary loop of Π iff the elementary subgraph of X for Π is strongly connected.

Proof. *From left to right:* Assume that X is an elementary loop of Π , and, for the sake of contradiction, that the elementary subgraph of X for Π is not strongly

connected. Then there is a strongly connected component in $(X, EC_\Pi(X))$ whose atoms Y are not reached from any atom in $X \setminus Y$. Clearly Y is a nonempty proper subset of X , and so is $X \setminus Y$. Furthermore, for every rule (12) in Π , at least one of the following conditions holds:

$$|A \cap X| > 1 , \quad (56)$$

$$A \cap (X \setminus Y) = \emptyset , \quad (57)$$

$$B \cap Y = \emptyset , \quad (58)$$

or

$$B \cap (X \setminus Y) \neq \emptyset . \quad (59)$$

However, (56) contradicts the assumption that Π is HEF. Also the fact that at least one of the conditions (57), (58), and (59) holds contradicts the assumption that $X \setminus Y$ is outbound in X for Π .

From right to left: Assume that the elementary subgraph of X for Π is strongly connected. For every nonempty proper subset Y of X , there is a minimum integer $i \geq 0$ such that $EC_\Pi^i(X)$ does not contain any edge from an atom in Y to an atom in $X \setminus Y$, but $EC_\Pi^{i+1}(X)$ contains such an edge. Thus some rule (12) in Π satisfies

$$|A \cap X| = 1 , \quad (60)$$

$$A \cap Y \neq \emptyset , \quad (61)$$

$$B \cap (X \setminus Y) \neq \emptyset ,$$

and

$$B \cap Y = \emptyset .$$

From (60) and (61), since $Y \subseteq X$,

$$A \cap (X \setminus Y) = \emptyset .$$

This shows that Y is outbound in X for Π . We conclude that X is an elementary loop of Π . ■

Although many properties of HCF programs still apply to HEF programs (e.g., equivalence between an HEF program and its shifted variant), the computational complexities of recognizing them are different. While an HCF program can be recognized in polynomial time (by computing the strongly connected components of its dependency graph), Fasseti and Palopoli (2010) showed that deciding whether a disjunctive program is HEF is **coNP**-complete.¹⁰ Theorem 4 established a similar complexity gap by showing that elementary loops are hard to verify in the case of arbitrary disjunctive programs, while for loops it remains a question of reachability.

¹⁰ The problem was left open in (Gebser et al. 2007), one of our conference papers that this paper extends.

Such elevated complexities may appear daunting, but the semantic similarities between HEF and HCF programs still exhibit that the syntactic concept of reachability merely gives a rough approximation of properties rendering disjunctive programs more difficult than nondisjunctive ones. As noted in (Fassetti and Palopoli 2010), identifying subclasses of (not necessarily HCF) disjunctive programs for which verifying the HEF property is tractable may be an interesting line of future research.

5 HEF Programs and Inherent Tightness

When we add more rules to a program, a stable model of the original program remains to be a stable model of the extended program if it satisfies the new rules.

Proposition 9

For any disjunctive program Π and any model X of Π , X is a stable model of Π iff there is a subset Π' of Π such that X is a stable model of Π' .

Proof. From left to right is clear.

From right to left: Assume that X is not a stable model of Π . Then some proper subset Y of X is a model of Π^X . For each subset Π' of Π , we have that $(\Pi')^X \subseteq \Pi^X$, so that Y is a model of $(\Pi')^X$ and X is not a stable model of Π' . ■

In view of Theorem^d 1, Proposition 9 tells us that, provided that X is a model of Π , in order to verify that X is a stable model of Π , it is sufficient to identify a subset Π' of Π such that X is a stable model of Π' . Of course, one can trivially take Π itself as the subset Π' , but there are nontrivial subsets that deserve attention. In fact, if Π is nondisjunctive, it is known that the subset Π' can be further restricted to a “tight” program (Fages 1994; Erdem and Lifschitz 2003)—the result known as “inherently tight” (Lin and Zhao 2003) or “weakly tight” (You et al. 2003) program. In the following, we simplify these notions and show that they can be extended to HEF programs.

Recall that a loop of Π is called *trivial* if it consists of a single atom such that the dependency graph of Π does not contain an edge from the atom to itself. In other words, a loop $\{a\}$ of Π is trivial if there is no rule (12) in Π such that $a \in A \cap B$.

Definition 2 ((Lee 2005))

A disjunctive program Π is called *tight* if every loop of Π is trivial.

As defined previously (Apt et al. 1988; Baral and Gelfond 1994; Inoue and Sakama 1998; Lee 2005), we call a set X of atoms *supported* by a disjunctive program Π if, for every $a \in X$, there is a rule (12) in Π_X such that $A \cap X = \{a\}$. Note that Definition 2 and the notion of support also apply to nondisjunctive programs as a special case.

The property of inherent tightness, introduced by Lin and Zhao (2003) for the case of nondisjunctive programs, can now be reformulated and generalized as follows.

Definition 3

A disjunctive program Π is called *inherently tight* on a set X of atoms if there is a subset Π' of Π such that Π' is tight and X is supported by Π' .

In the case of nondisjunctive programs, this reformulation of inherent tightness is similar to “well-supportedness” (Fages 1994). Furthermore, weak tightness, introduced in (You et al. 2003), is closely related to the notion of inherent tightness.

For nondisjunctive programs, it is known that their stable models can be characterized in terms of inherent tightness.

Proposition 10 ((Fages 1994; Lin and Zhao 2003; You et al. 2003))

For any nondisjunctive program Π and any model X of Π , X is a stable model of Π iff Π is inherently tight on X .

One may wonder whether Proposition 10 can be extended to disjunctive programs as well, given that Definition 3 readily applies to them. However, only one direction of Proposition 10 holds in the case of arbitrary disjunctive programs.

Proposition 11

For any disjunctive program Π and any model X of Π , if Π is inherently tight on X , then X is a stable model of Π .

Proof. Assume that Π is inherently tight on X . Then there is a subset Π' of Π such that Π' is tight and X is supported by Π' . By Proposition 2 from (Lee and Lifschitz 2003), X is a stable model of Π' , and, by Proposition 9, X is a stable model of Π . ■

To see that the converse of Proposition 11 does not hold, consider Π as follows:

$$\begin{aligned} p &\leftarrow q \\ q &\leftarrow p \\ p ; q &\leftarrow . \end{aligned}$$

Set $\{p, q\}$ is a stable model of Π . On the other hand, since any tight subset Π' of Π must exclude the first or the second rule, it follows that $\{p, q\}$ is not supported by Π' . But this means that Π is not inherently tight on $\{p, q\}$. It is also worthwhile to note that $\{p, q\}$ is an elementary loop of Π , so that Π is not HEF (and not HCF). Indeed, the following theorem tells us that Proposition 10 can be extended to HEF programs.

Theorem 6

For any HEF program Π and any model X of Π , X is a stable model of Π iff Π is inherently tight on X .

Proof. *From left to right:* Assume that X is a stable model of Π . By Proposition^d 3 (and the fact that every atom of X occurs in Π_X), any nonempty subset Y of X contains some elementary loop Z of Π_X that is not outbound in Y for Π_X .¹¹ That is, every rule (12) of Π_X satisfies at least one of the following conditions:

$$A \cap Z = \emptyset ,$$

¹¹ If Y is an elementary loop of Π_X , take $Z = Y$.

$$B \cap (Y \setminus Z) = \emptyset, \quad (62)$$

$$A \cap (Y \setminus Z) \neq \emptyset, \quad (63)$$

or

$$B \cap Z \neq \emptyset.$$

From (63), since $Y \subseteq X$,

$$A \cap (X \setminus Z) \neq \emptyset.$$

On the other hand, since X is a stable model of Π and Z is a nonempty subset of X , by Theorem^d 1 (b), there is a rule (12) in Π_X such that

$$A \cap Z \neq \emptyset, \quad (64)$$

$$A \cap (X \setminus Z) = \emptyset, \quad (65)$$

and

$$B \cap Z = \emptyset, \quad (66)$$

so that (62) must hold, which, in combination with (66), gives us that

$$B \cap Y = \emptyset. \quad (67)$$

Furthermore, since $\Pi_X \subseteq \Pi$, we have that Z is an elementary loop of Π . Given that Π is HEF, from (64) and (65), we conclude that

$$A \cap X = \{a\} \quad (68)$$

for some $a \in Z$, where $a \in Y$ also holds because $Z \subseteq Y$. We have thus shown that, for any nonempty subset Y of X , there is a rule (12) in Π_X such that (67) and (68) for $a \in Y$ hold. Starting from $X^0 = \emptyset$ and $\Pi^0 = \emptyset$, when, for $1 \leq i \leq |X|$, we let

- $Y = (X \setminus X^{i-1})$,
- Π^i is obtained from Π^{i-1} by adding some rule (12) in Π_X such that (67) and (68) for $a \in Y$ hold, and
- $X^i = X^{i-1} \cup \{a\}$,

then X is supported by $\Pi' = \Pi^{|X|}$. Furthermore, since a rule (12) in $(\Pi^i \setminus \Pi^{i-1})$ satisfies $A \cap X^{i-1} = \emptyset$ and $B \subseteq X^{i-1}$ for every $1 \leq i \leq |X|$, Π' is tight by construction, which shows that Π is inherently tight on X .

From right to left follows immediately from Proposition 11. ■

Since every HCF program is HEF, Theorem 6 applies also to HCF programs.

We demonstrated that, by turning to the notion of an elementary loop in place of a loop, we obtain generalizations of results known for loops, such as Theorem 5. This brings our attention to the following question. As a tight program can be characterized in terms of loops, can the notion of a tight program be generalized by referring to elementary loops instead? To answer it, let us first modify Definition 2 in the following way.

Definition 4

A disjunctive program Π is called *e-tight* if every elementary loop of Π is trivial.

Since every elementary loop is a loop, it is clear that a tight program is e-tight as well. But is the class of e-tight programs more general than the class of tight programs? One reason why this is an interesting question to consider is because, if so, it would lead to a generalization of Fages' theorem (Fages 1994), which would yield a more general class of programs for which the stable model semantics coincides with the completion semantics. However, it turns out that e-tight programs are not more general than tight programs.

Proposition 12

For any disjunctive program Π , Π is e-tight iff Π is tight.

Proof. *From left to right:* Assume that Π is not tight. Then there is a minimal nontrivial loop X of Π , and the subgraph of the dependency graph of Π induced by X yields a simple directed cycle. That is, for any nonempty proper subset Y of X , there is a rule (12) in Π such that $A \cap X = \{a\}$, $B \cap X = \{b\}$ for atoms $a \in Y$, $b \in X \setminus Y$. This shows that Y is outbound in X for Π , so that X is a nontrivial elementary loop of Π .

From right to left is clear. ■

This result also tells us that the notion of an inherently tight program does not become more general by referring to elementary loops, i.e., by replacing the part “ Π' is tight” in the statement of Definition 3 with “ Π' is e-tight.”

6 HEF Programs and Stability Checking

For a disjunctive program, the problem of deciding whether a given model is stable is **coNP**-complete (Eiter and Gottlob 1995). On the other hand, in view of Theorem 5, the same problem is tractable for HEF programs. In order to check the stability of a model in polynomial time, Leone *et al.* (1997) presented an operational framework, which, for HCF programs, allows for deciding whether a model is stable. Given a disjunctive program Π and sets X, Y of atoms, they defined a sequence $R_{\Pi,X}^0(Y), R_{\Pi,X}^1(Y), \dots$, which converges to a limit $R_{\Pi,X}^\omega(Y)$, in the following way:

- $R_{\Pi,X}^0(Y) = Y$ and
- $R_{\Pi,X}^{i+1}(Y)$ is obtained by removing every atom a from $R_{\Pi,X}^i(Y)$ such that some rule (12) in Π_X satisfies $A \cap (X \cup \{a\}) = \{a\}$ and $B \cap R_{\Pi,X}^i(Y) = \emptyset$.¹²

The disjunctive rules considered in (Leone *et al.* 1997) do not admit double negations in rule bodies, but its (dis)use merely affects conditions like $X \models B, F$ (or $X \not\models B, F$) determining Π_X , while it is inconsequential otherwise. Hence, the following results remain valid for disjunctive programs whose rules are of the form (12).

¹² If Y is a subset of X , the condition “ $A \cap (X \cup \{a\}) = \{a\}$ ” can be replaced with “ $A \cap X = \{a\}$ ” without altering $R_{\Pi,X}^\omega(Y)$.

Proposition 13 ((Leone et al. 1997, Lemma 6.4))

For any disjunctive program Π and any sets X, Y of atoms that occur in Π , all subsets of Y that are unfounded by Π w.r.t. X are contained in $R_{\Pi, X}^\omega(Y)$.

Proposition 14 ((Leone et al. 1997, Proposition 6.5))

For any disjunctive program Π and any set X of atoms that occur in Π , if $R_{\Pi, X}^\omega(X) = \emptyset$, then X contains no nonempty unfounded sets for Π w.r.t. X .

Proposition 15 ((Leone et al. 1997, Theorem 6.9))

For any HCF program Π and any set X of atoms that occur in Π , X contains no nonempty unfounded sets for Π w.r.t. X iff $R_{\Pi, X}^\omega(X) = \emptyset$.

For a model X of Π , in view of Theorem^d 1 (b'), Proposition 14 tells us that X is a stable model of Π if $R_{\Pi, X}^\omega(X) = \emptyset$. As stated in Proposition 15, the converse also holds if Π is HCF. We below extend this result to HEF programs.

Proposition 16

For any HEF program Π , any set X of atoms, and any subset Y of X whose atoms occur in Π , if $R_{\Pi, X}^\omega(Y) \neq \emptyset$, then $R_{\Pi, X}^\omega(Y)$ contains an elementary loop Z of Π that is unfounded by Π w.r.t. X .

Proof. Assume that $R_{\Pi, X}^\omega(Y) \neq \emptyset$. Then, for every rule (12) of Π_X , at least one of the following conditions holds:

$$|A \cap X| > 1, \quad (69)$$

$$A \cap R_{\Pi, X}^\omega(Y) = \emptyset, \quad (70)$$

or

$$B \cap R_{\Pi, X}^\omega(Y) \neq \emptyset. \quad (71)$$

By Proposition^d 3, $R_{\Pi, X}^\omega(Y)$ contains some elementary loop Z of Π that is not outbound in $R_{\Pi, X}^\omega(Y)$ for Π .¹³ For the sake of contradiction, assume that Z is not unfounded by Π w.r.t. X . Then there is a rule (12) in Π_X such that

$$A \cap Z \neq \emptyset, \quad (72)$$

$$A \cap (X \setminus Z) = \emptyset, \quad (73)$$

and

$$B \cap Z = \emptyset. \quad (74)$$

From (73), since $R_{\Pi, X}^\omega(Y) \subseteq X$,

$$A \cap (R_{\Pi, X}^\omega(Y) \setminus Z) = \emptyset,$$

which, in combination with (72), (74), and the fact that Z is not outbound in $R_{\Pi, X}^\omega(Y)$ for Π , gives us that

$$B \cap (R_{\Pi, X}^\omega(Y) \setminus Z) = \emptyset. \quad (75)$$

¹³ If $R_{\Pi, X}^\omega(Y)$ is an elementary loop of Π , take $Z = R_{\Pi, X}^\omega(Y)$.

From (74) and (75), we conclude that (71) does not hold. Furthermore, since $Z \subseteq R_{\Pi, X}^\omega(Y)$, (72) implies that (70) does not hold. Hence, (69) must hold, which, in combination with (73), gives us that

$$|A \cap Z| > 1.$$

But since Z is an elementary loop of Π , this contradicts that Π is HEF. ■

We are now ready to generalize Proposition 15 to HEF programs.

Theorem 7

For any HEF program Π and any set X of atoms that occur in Π , X contains no nonempty unfounded sets for Π w.r.t. X iff $R_{\Pi, X}^\omega(X) = \emptyset$.

Proof. From left to right follows immediately from Proposition 16.

From right to left follows immediately from Proposition 14. ■

Regarding the models of HEF programs, we derive the following corollary.

Corollary 2

For any HEF program Π and any set X of atoms that occur in Π , X is a stable model of Π iff X is a model of Π such that $R_{\Pi, X}^\omega(X) = \emptyset$.

Proof. Both directions follow immediately from Theorem^d 1 (b') and Theorem 7. ■

For instance, reconsider Π_2 from Section 4, which is HEF, but not HCF. Hence, Corollary 2 applies, but Proposition 15 does not apply. Indeed, since $R_{\Pi_2, X}^\omega(X) = X$ for (non-stable) model $X = \{p, q, r\}$ of Π_2 , Corollary 2 allows us to conclude that X is not a stable model of Π_2 . On the other hand, for model $\{p\}$ of Π_2 , we have that $R_{\Pi_2, \{p\}}^\omega(\{p\}) = \emptyset$, which implies that $\{p\}$ is a stable model of Π_2 .

In Section 3.3, we defined the notion of an elementarily unfounded set and showed that it coincides with a minimal nonempty unfounded set. Thus stability checking can be cast into the problem of ensuring the absence of elementarily unfounded sets. Since every elementarily unfounded set is a loop, it is clearly contained in a maximal loop, which allows us to modularize the consideration of (elementarily) unfounded sets. The idea of using maximal loops for partitioning a program and confining stability checking to subprograms was already exploited by Leone *et al.* (1997) and Koch *et al.* (2003). In fact, for a disjunctive program Π and a set X of atoms, Leone *et al.* (1997) showed how stability can be checked separately for maximal loops of Π , and Koch *et al.* (2003) developed this idea further by considering maximal loops of the smaller program $\Pi_{X, R_{\Pi, X}^\omega(X)}$. We below describe a notion called “bounding loops,” which go beyond such maximal loops.

For a disjunctive program Π and a set X of atoms, we say that a subset Y of X is a *bounding loop* of Π w.r.t. X if Y is maximal among all subsets Z of X such that Z is a loop of $\Pi_{X, Z}$ and $R_{\Pi, X}^\omega(Z) = Z$. Note that there are two crucial differences between bounding loops and elementarily unfounded sets. First, a bounding loop Y of Π w.r.t. X is not necessarily an elementary loop of $\Pi_{X, Y}$. Next, it does not need to be unfounded by Π w.r.t. X . To see this, observe that $X = \{p, q, r\}$ is a

bounding loop of Π_2 w.r.t. X that is not (elementarily) unfounded by Π_2 w.r.t. X . Furthermore, one can check that $\{p, r\}$ and $\{q, r\}$ are (elementarily) unfounded by Π_2 w.r.t. X , and thus Proposition^d 5 tells us that X is not an elementary loop of $(\Pi_2)_{X,X}$.

The following two propositions describe properties of bounding loops that are similar to those of maximal loops, as used in (Leone et al. 1997; Koch et al. 2003).

Proposition 17

For any disjunctive program Π and any set X of atoms, all bounding loops of Π w.r.t. X are mutually disjoint.

Proof. Let Y_1, Y_2 be subsets of X such that Y_1 is a loop of Π_{X,Y_1} , Y_2 is a loop of Π_{X,Y_2} , $R_{\Pi,X}^\omega(Y_1) = Y_1$, and $R_{\Pi,X}^\omega(Y_2) = Y_2$. If $Y_1 \cap Y_2 \neq \emptyset$, the fact that $\Pi_{X,Y_1} \cup \Pi_{X,Y_2} \subseteq \Pi_{X,Y_1 \cup Y_2}$ implies that $Y_1 \cup Y_2$ is a loop of $\Pi_{X,Y_1 \cup Y_2}$. Furthermore, since $R_{\Pi,X}^\omega(Y_1) = Y_1$ and $R_{\Pi,X}^\omega(Y_2) = Y_2$, for any rule (12) of Π_X such that $A \cap (Y_1 \cup Y_2) \neq \emptyset$ and $B \cap (Y_1 \cup Y_2) = \emptyset$, we have that $|A \cap X| > 1$, from which we conclude that $R_{\Pi,X}^\omega(Y_1 \cup Y_2) = Y_1 \cup Y_2$. Since bounding loops of Π w.r.t. X are maximal among all subsets Z of X such that Z is a loop of $\Pi_{X,Z}$ and $R_{\Pi,X}^\omega(Z) = Z$, this shows that they must be mutually disjoint. ■

Proposition 18

For any disjunctive program Π and any set X of atoms that occur in Π , every elementarily unfounded set for Π w.r.t. X is a singleton or contained in some bounding loop of Π w.r.t. X .

Proof. Assume that Y is an elementarily unfounded set for Π w.r.t. X that is not a singleton. From the definition of an elementarily unfounded set, it follows that Y is a loop of $\Pi_{X,Y}$ and a subset of X . Since Y is also unfounded by Π w.r.t. X , by Proposition 13, we have that $R_{\Pi,X}^\omega(Y) = Y$. This shows that Y is contained in some maximal subset Z of X such that Z is a loop of $\Pi_{X,Z}$ and $R_{\Pi,X}^\omega(Z) = Z$. ■

Proposition 17 and Proposition 18 tell us that checking the absence of elementarily unfounded sets can be accomplished separately for bounding loops.

Proposition 19

For any disjunctive program Π and any model X of Π , X is a stable model of Π iff

- X is supported by Π and
- no bounding loop of Π w.r.t. X contains a nonempty unfounded set for Π w.r.t. X .

Proof. From left to right follows immediately from Theorem^d 1 (b') (and the fact that every atom of X occurs in Π).

From right to left: Assume that X is not a stable model of Π but supported by Π . Then, by Theorem^d 1 (e'), X contains some elementarily unfounded set Y for Π w.r.t. X . If Y is not a singleton, by Proposition 18, Y is contained in some bounding loop of Π w.r.t. X . Otherwise, if Y is a singleton, the assumption that X is supported by Π implies that there is a rule (12) in $\Pi_{X,Y}$ such that $A \cap X = Y$, so

that Y is a loop of $\Pi_{X,Y}$. Since Y is unfounded by Π w.r.t. X , by Proposition 13, we also have that $R_{\Pi,X}^\omega(Y) = Y$. This shows that Y is contained in some maximal subset Z of X such that Z is a loop of $\Pi_{X,Z}$ and $R_{\Pi,X}^\omega(Z) = Z$. ■

As it is easy to check that X is supported by Π , Proposition 19 tells us that the investigation of bounding loops constitutes the hard part of stability checking. But this is not due to the hardness of identifying them. In fact, the following method can be used to compute all bounding loops Z of Π w.r.t. X in polynomial time:

1. Let $Y = X$.
2. Let $Z = R_{\Pi,X}^\omega(Y)$. (Note that $Z = R_{\Pi,X}^\omega(Z)$.)
3. If Z is a loop of $\Pi_{X,Z}$, then mark Z as a bounding loop of Π w.r.t. X . Otherwise, proceed with Step 2 for every maximal loop Y of $\Pi_{X,Z}$ that is contained in Z .

The soundness of this approach is straightforward, given that operator R is monotone, i.e., $R_{\Pi,X}^\omega(Z) \subseteq R_{\Pi,X}^\omega(Y)$ if $Z \subseteq Y$ (used in Step 2), and likewise that $\Pi_{X,Y} \subseteq \Pi_{X,Z}$ if $Y \subseteq Z$ (used in Step 3).

For illustration, consider the following program Π_4 :

$$\begin{array}{lll}
 p \leftarrow r & s \leftarrow p & p ; q \leftarrow u \\
 q \leftarrow r & s \leftarrow t & s ; t \leftarrow q \\
 r \leftarrow p, q & t \leftarrow s, u & r ; u \leftarrow t \quad u ; v \leftarrow .
 \end{array}$$

For $X = \{p, q, r, s, t, u\}$, it holds that $(\Pi_4)_{X,X} = \Pi_4$ and also that X is a loop of Π_4 . However, $Y = R_{\Pi_4,X}^\omega(X) = \{p, q, r, s, t\} \neq X$, so that X is not a bounding loop of Π_4 w.r.t. X . On the other hand, Y is not a loop of $(\Pi_4)_{X,Y}$, which does not include the last two rules where u occurs in the head. Rather, $Z = \{p, q, r\}$ and $Z' = \{s, t\}$ are the maximal loops of $(\Pi_4)_{X,Y}$ that are contained in Y . In view of the rules in the second column, $R_{\Pi_4,X}^\omega(Z') = \emptyset$, which shows that no subset of Z' is a bounding loop of Π_4 w.r.t. X . For $Z = \{p, q, r\}$, we obtain $R_{\Pi_4,X}^\omega(Z) = Z$, and Z is also a loop of $(\Pi_4)_{X,Z}$ (due to the rules in the first column). That is, Z is the single bounding loop of Π_4 w.r.t. X .

Let us compare this outcome with the ones of previous methods for modularizing stability checking. Since the approach described in (Leone et al. 1997) considers maximal loops of the original program and $X = \{p, q, r, s, t, u\}$ is a (maximal) loop of Π_4 , it cannot be used to decompose X , and the only applicable simplification is to remove u by means of R , that is, $Y = R_{\Pi_4,X}^\omega(X) = \{p, q, r, s, t\}$. The approach in (Koch et al. 2003) considers the maximal loops of $\Pi_{X,R_{\Pi,X}^\omega(X)}$, which, for $(\Pi_4)_{X,Y}$, gives $Z = \{p, q, r\}$ and $Z' = \{s, t\}$. As described in (Koch et al. 2003), since the subprograms $(\Pi_4)_{X,Z}$ and $(\Pi_4)_{X,Z'}$ are not HCF, they are not simplified any further and used to separately check for a nonempty unfounded subset of Z or Z' , respectively. Unlike this, the notion of a bounding loop allowed us to eliminate all subsets of Z' as potential nonempty unfounded sets.

Turning back to HEF programs, from Proposition 16, we derive the following proposition for a subprogram $\Pi_{X,Y}$ associated with a bounding loop Y of Π w.r.t. X .

Proposition 20

For any disjunctive program Π , any set X of atoms, and any bounding loop Y of Π w.r.t. X , if $\Pi_{X,Y}$ is HEF, then Y contains a nonempty unfounded set for Π w.r.t. X .

Proof. Assume that $\Pi_{X,Y}$ is HEF. From the definition of a bounding loop, it follows that all atoms of Y occur in $\Pi_{X,Y}$ and that $Y = R_{\Pi,X}^\omega(Y) = R_{(\Pi_{X,Y}),X}^\omega(Y) \subseteq X$. By Proposition 16, we conclude that $R_{(\Pi_{X,Y}),X}^\omega(Y)$ contains an elementary loop Z of $\Pi_{X,Y}$ that is unfounded by $\Pi_{X,Y}$ w.r.t. X . From the definition of $\Pi_{X,Y}$ and since Z is contained in Y , we conclude that Z is also unfounded by Π w.r.t. X . ■

Proposition 20 tells us that the existence of a bounding loop Y of Π w.r.t. X whose associated subprogram $\Pi_{X,Y}$ is HEF is already sufficient to conclude that X is not a stable model of Π . Reconsidering the bounding loop $Z = \{p, q, r\}$ of Π_4 w.r.t. $X = \{p, q, r, s, t, u\}$, we have that $(\Pi_4)_{X,Z}$, consisting of the rules in the first column along with the disjunctive rule containing p and q in the head, is HEF (neither $\{p, q\}$ nor $\{p, q, r\}$ is an elementary loop of $(\Pi_4)_{X,Z}$). Thus X is not a stable model of Π_4 . Indeed, Z contains two (elementarily) unfounded sets for Π_4 w.r.t. X : $\{p, r\}$ and $\{q, r\}$.

7 Conclusion

The notion of an elementary loop and its properties provide useful insights into the concept of a loop and the relationship between nondisjunctive programs and disjunctive programs. By turning to the notion of an elementary loop in place of a loop, we could strengthen the theorem by Lin and Zhao (2004), its generalization to disjunctive programs (Lee and Lifschitz 2003), and the main theorem (for programs in canonical form) from (Lee 2005). The semantic (e.g., Theorem^d 2) and complexity-theoretic (e.g., Theorem 4) properties of elementary loops indicate their close relationship to unfounded sets. Compared with loops, elementary loops provide a deeper understanding of the internal structure of unfounded sets. In fact, we have shown that loop formulas of non-elementary loops are unnecessary for checking the stability of a model. It is an interesting open question whether this result can be used to improve computation performed by answer set solvers that calculate loop formulas, such as ASSAT, CLASPD, and CMODELS.

A method to identify an elementarily unfounded set for an HEF program was presented in (Gebser et al. 2007).¹⁴ An orthogonal approach is implemented in CLASPD: it greedily adds atoms a to an unfounded set Y if $Y \cup \{a\}$ stays unfounded and has a shorter loop formula than Y . However, the potential of unfounded set “optimization” has not been studied in-depth so far, and the theoretical foundations laid here may be useful for future investigations in this direction. Regarding nondisjunctive programs, elementary loops can be distinct from loops of programs called “binary” (Janhunen 2006). Moreover, modularity aspects of disjunctive stable models (Janhunen et al. 2009), which are closely related to loop formulas (Lemma 5.4 from (Janhunen et al. 2009)), can be refined by referring to elementary loops in

¹⁴ It is omitted in this paper for brevity.

place of loops. Lifting elementary loops to first-order programs, as already done for loops (Chen et al. 2006; Lee and Meng 2008), may also be a direction to explore.

The notion of an HEF program is a strict generalization of the notion of an HCF program. On the one hand, nice properties of HCF programs still apply to HEF programs. In particular, their inherent tightness on stable models grants the soundness of shifting head atoms into the body as well as the possibility of performing stability checks in polynomial time. This however implies that a disjunctive program encoding an instance of a Σ_2^P -hard problem is unlikely to be HEF. On the other hand, recognizing elementary loops and verifying the HEF property are both intractable in the case of arbitrary disjunctive programs, but tractable for nondisjunctive and HCF programs. This parallels the complexity of stability checking (Eiter and Gottlob 1995), and it also tells us that the inherent complexities of computational tasks dealing with elementary loops tightly correlate to the fragment of disjunctive programs under consideration. As the latter does not apply to recognizing loops or verifying the HCF property, the notion of an HEF program more precisely renders what makes arbitrary disjunctive programs more difficult than nondisjunctive programs. Whether this admits (syntactic) characterizations of yet unknown subclasses of disjunctive programs for which verifying the HEF property is tractable is an interesting open question.

Acknowledgments

We are grateful to Selim Erdoğan, Tomi Janhunen, Dan Lessin, Vladimir Lifschitz, Torsten Schaub, Jicheng Zhao, and the anonymous referees of (Gebser et al. 2006; Gebser et al. 2007) and this paper for their useful comments. Martin Gebser was partially supported by the German Research Foundation under Grant SCHA 550/8-1. Joohyung Lee was partially supported by the National Science Foundation under Grant IIS-0916116 and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through US army. Yuliya Lierler was partially supported by the National Science Foundation under Grant IIS-0712113 and by a 2010 Computing Innovation Fellowship.

References

- APT, K., BLAIR, H., AND WALKER, A. 1988. Towards a theory of declarative knowledge. In *Foundations of Deductive Databases and Logic Programming*, J. Minker, Ed. Morgan Kaufmann, 89–148.
- BARAL, C. AND GELFOND, M. 1994. Logic programming and knowledge representation. *Journal of Logic Programming* 19,20, 73–148.
- BEN-ELIYAHU, R. AND DECHTER, R. 1994. Propositional semantics for disjunctive logic programs. *Annals of Mathematics and Artificial Intelligence* 12, 53–87.
- CHEN, Y., LIN, F., WANG, Y., AND ZHANG, M. 2006. First-order loop formulas for normal logic programs. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR’06)*, P. Doherty, J. Mylopoulos, and C. Welty, Eds. AAAI Press, 298–307.
- CLARK, K. 1978. Negation as failure. In *Logic and Data Bases*, H. Gallaire and J. Minker, Eds. Plenum Press, New York, 293–322.

- DRESCHER, C., GEBSER, M., GROTE, T., KAUFMANN, B., KÖNIG, A., OSTROWSKI, M., AND SCHAUB, T. 2008. Conflict-driven disjunctive answer set solving. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, G. Brewka and J. Lang, Eds. AAAI Press, 422–432.
- EITER, T. AND FINK, M. 2003. Uniform equivalence of logic programs under the stable model semantics. In *Proceedings of the Nineteenth International Conference on Logic Programming (ICLP'03)*, C. Palamidessi, Ed. Springer-Verlag, 224–238.
- EITER, T. AND GOTTLÖB, G. 1995. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence* 15, 3-4, 289–323.
- ERDEM, E. AND LIFSCHITZ, V. 2003. Tight logic programs. *Theory and Practice of Logic Programming* 3, 499–518.
- FAGES, F. 1994. Consistency of Clark's completion and existence of stable models. *Journal of Methods of Logic in Computer Science* 1, 51–60.
- FASSETTI, F. AND PALOPOLI, L. 2010. On the complexity of identifying head elementary set free programs. *Theory and Practice of Logic Programming* 10(1), 113–123.
- FERRARIS, P., LEE, J., AND LIFSCHITZ, V. 2007. A new perspective on stable models. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI'07)*. 372–379.
- FERRARIS, P., LEE, J., AND LIFSCHITZ, V. 2011. Stable models and circumscription. *Artificial Intelligence* 175, 236–263.
- GEBSER, M., LEE, J., AND LIERLER, Y. 2006. Elementary sets for logic programs. In *Proceedings of the Twenty-first AAAI Conference on Artificial Intelligence (AAAI'06)*.
- GEBSER, M., LEE, J., AND LIERLER, Y. 2007. Head-elementary-set-free logic programs. In *Proceedings of the Ninth International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR'07)*. 149–161.
- GEBSER, M. AND SCHAUB, T. 2005. Loops: Relevant or redundant? In *Proceedings of the Eighth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'05)*. 53–65.
- GEBSER, M., SCHAUB, T., TOMPITS, H., AND WOLTRAN, S. 2008. Alternative characterizations for program equivalence under answer-set semantics based on unfounded sets. In *Proceedings of the Fifth International Symposium on Foundations of Information and Knowledge Systems (FoIKS'08)*, S. Hartmann and G. Kern-Isberner, Eds. Springer-Verlag, 24–41.
- GELFOND, M., LIFSCHITZ, V., PRZYMUSIŃSKA, H., AND TRUSZCZYŃSKI, M. 1991. Disjunctive defaults. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, J. Allen, R. Fikes, and E. Sandewall, Eds. 230–237.
- GIUNCHIGLIA, E., LIERLER, Y., AND MARATEA, M. 2004. SAT-based answer set programming. In *Proceedings of the Nineteenth AAAI Conference on Artificial Intelligence (AAAI)*. 61–66.
- INOUE, K. AND SAKAMA, C. 1998. Negation as failure in the head. *Journal of Logic Programming* 35, 39–78.
- JANHUNEN, T. 2006. Some (in)translatability results for normal logic programs and propositional theories. *Journal of Applied Non-Classical Logics* 16, 1-2, 35–86.
- JANHUNEN, T., NIEMELÄ, I., SEIPEL, D., SIMONS, P., AND YOU, J.-H. 2006. Unfolding partiality and disjunctions in stable model semantics. *ACM Transactions on Computational Logic* 7, 1, 1–37.
- JANHUNEN, T., OIKARINEN, E., TOMPITS, H., AND WOLTRAN, S. 2009. Modularity aspects of disjunctive stable models. *Journal of Artificial Intelligence Research* 35, 813–857.

- KOCH, C., LEONE, N., AND PFEIFER, G. 2003. Enhancing disjunctive logic programming systems by SAT checkers. *Artificial Intelligence* 151, 177–212.
- LEE, J. 2004. Nondefinite vs. definite causal theories. In *Proceedings of the Seventh International Conference on Logic Programming and Nonmonotonic Reasoning (LP-NMR'04)*. 141–153.
- LEE, J. 2005. A model-theoretic counterpart of loop formulas. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*. Professional Book Center, 503–508.
- LEE, J. AND LIFSCHITZ, V. 2003. Loop formulas for disjunctive logic programs. In *Proceedings of Nineteenth International Conference on Logic Programming (ICLP'03)*. 451–465.
- LEE, J. AND LIN, F. 2006. Loop formulas for circumscription. *Artificial Intelligence* 170, 2, 160–185.
- LEE, J. AND MENG, Y. 2008. On loop formulas with variables. In *Proceedings of the Eleventh International Conference on Knowledge Representation and Reasoning (KR'08)*. 444–453.
- LEE, J. AND MENG, Y. 2009. On reductive semantics of aggregates in answer set programming. In *Proceedings of the Tenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'09)*. 182–195.
- LEONE, N., PFEIFER, G., FABER, W., EITER, T., GOTTLOB, G., PERRI, S., AND SCARCELLO, F. 2006. The DLV system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7, 3.
- LEONE, N., RULLO, P., AND SCARCELLO, F. 1997. Disjunctive stable models: Unfounded sets, fixpoint semantics, and computation. *Information and Computation* 135(2), 69–112.
- LIERLER, Y. 2005. CMODELS: SAT-based disjunctive answer set solver. In *Proceedings of the Eighth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'05)*. 447–452.
- LIFSCHITZ, V., TANG, L. R., AND TURNER, H. 1999. Nested expressions in logic programs. *Annals of Mathematics and Artificial Intelligence* 25, 369–389.
- LIFSCHITZ, V. AND RAZBOROV, A. 2006. Why are there so many loop formulas? *ACM Transactions on Computational Logic* 7, 2, 261–268.
- LIN, F. AND ZHAO, J. 2003. On tight logic programs and yet another translation from normal logic programs to propositional logic. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*. 853–858.
- LIN, F. AND ZHAO, Y. 2004. ASSAT: Computing answer sets of a logic program by SAT solvers. *Artificial Intelligence* 157, 115–137.
- LIU, L. AND TRUSZCZYNSKI, M. 2006. Properties and applications of programs with monotone and convex constraints. *Journal of Artificial Intelligence Research* 27, 299–334.
- SACCÁ, D. AND ZANIOLO, C. 1990. Stable models and non-determinism in logic programs with negation. In *Proceedings of the Ninth ACM Symposium on Principles of Database Systems (PODS)*. 205–217.
- YOU, J.-H. AND LIU, G. 2008. Loop formulas for logic programs with arbitrary constraint atoms. In *Proceedings of the Twenty-third AAAI Conference on Artificial Intelligence (AAAI'08)*. 584–589.
- YOU, J.-H., YUAN, L.-Y., AND ZHANG, M. 2003. On the equivalence between answer sets and models of completion for nested logic programs. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*. 859–866.