Theoretical Computer Science 415 (2012) 60–85

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

The loop formula based semantics of description logic programs[☆]

Yisong Wang[a,c,∗], Jia-Huai You[b], Li Yan Yuan[b], Yi-Dong Shen[c], Mingyi Zhang[d]

[a] *School of Computer Science and Information, Guizhou University, Guiyang, 550025, China*
[b] *Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2R3*
[c] *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, 100190, China*
[d] *Guizhou Academy of Sciences, Guiyang, 550001, China*

A R T I C L E I N F O

Article history:
Received 30 January 2011
Received in revised form 30 August 2011
Accepted 29 October 2011
Communicated by G. Levi

Keywords:
Semantic Web
Answer sets
Description logic programs
Loop formulas

A B S T R A C T

Description logic programs (dl-programs) proposed by Eiter et al. constitute an elegant yet powerful formalism for the integration of answer set programming with description logics, for the Semantic Web. In this paper, we generalize the notions of completion and loop formulas of logic programs to description logic programs and show that the answer sets of a dl-program can be precisely captured by the models of its completion and loop formulas. Furthermore, we propose a new, alternative semantics for dl-programs, called the *canonical answer set semantics*, which is defined by the models of completion that satisfy what are called canonical loop formulas. A desirable property of canonical answer sets is that they are free of circular justifications. Some properties of canonical answer sets are also explored and we compare the canonical answer set semantics with the FLP-semantics and the answer set semantics by translating dl-programs into logic programs with abstract constraints. We present a clear picture on the relationship among these semantics variations for dl-programs.

1. Introduction

Logic programming under the answer set semantics, called Answer Set Programming (ASP), is a nonmonotonic reasoning paradigm for declarative problem solving [20,23]. Recently, there has been extensive interest in combining ASP with other computational and reasoning paradigms. One of the main interests in this direction is the integration of ASP with ontology reasoning, for the Semantic Web.

The Semantic Web is an evolving development of the World Wide Web in which the meaning of information and services on the web are defined, so that the web content can be precisely understood and used by agents [3]. For this purpose, a layered structure including the Rules Layer built on top of the Ontology Layer has been recognized as a fundamental framework. Description Logics (DLs) [2] provide a formal basis for the Web Ontology Language which is the standard of the Ontology Layer [31].

Adding nonmonotonic rules to the Rules Layer would allow default reasoning with ontologies. For example, we know that most *natural kinds* do not have a clear cut definition [1]. For instance, a precise definition of *scientist* seems to be difficult by enumerating what a scientist is, and does. Though we can say that a scientist possesses expert knowledge on the subject of his or her investigation, we still need a quantitative definition of expert knowledge, which seems impossible. Using

☆ This is a revision and substantial extension of Wang et al. (2010) [32].

∗ Corresponding author at: School of Computer Science and Information, Guizhou University, Guiyang, 550025, China. Tel.: +86 851 3621638; fax: +86 851 3621638.

E-mail addresses: yswang168@gmail.com (Y. Wang), you@cs.ualberta.ca (J.-H. You), yuan@cs.ualberta.ca (L.Y. Yuan), ydshen@ios.ac.cn (Y.-D. Shen), zhangmingyi045@yahoo.com.cn (M. Zhang).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2011.10.026

nonmonotonic rules, we can perform default, typicality reasoning over categories, concepts, and roles. The integration of DLs and (nonmonotonic) rules has been extensively investigated as a crucial problem in the study of the Semantic Web, such as the *Semantic Web Rule Language* (SWRL) [14], *MKNF knowledge bases* [22], and *Description logic programs* (*dl-programs*) [8].

There are different approaches to the integration of ASP with description logics. The focus of this paper is on the approach based on dl-programs. Informally, a dl-program is a pair (O, P), where O is a DL knowledge base and P is a logic program whose rule bodies may contain queries, embedded in *dl-atoms*, to the knowledge base O. The answer to such a query depends on inferences by rules over the DL knowledge base O. In this way, rules are built on top of ontologies. On the other hand, ontology reasoning is also enhanced, since it depends not only on O but also on inferences using (nonmonotonic) rules. Two semantics for dl-programs have been proposed, one of which is based on *strong answer sets* and the other based on *weak answer sets*.

In this paper, we generalize the notions of completion and loop formulas of logic programs [17] to dl-programs and show that weak and strong answer sets of a dl-program can be captured precisely by the models of its completion and the corresponding loop formulas. This provides not only a semantic characterization of answer sets for dl-programs but also an alternative mechanism for answer set computation, using a dl-reasoner and a SAT solver.

As commented by [8], the reason to introduce strong answer sets is because some weak answer sets seem counterintuitive due to "self-supporting" loops. Recently however, one of the co-authors of this paper, Yi-Dong Shen, discovered that strong answer sets may also possess self-supporting loops, and a detailed analysis leads to the conclusion that the problem cannot be easily fixed by an alternative definition of *reduct*, since the reduct of dl-atoms may not be able to capture dynamically generated self-supports arising from the integrated context.

One partial solution proposed in this paper is to use loop formulas as a way to define answer sets for dl-programs that are free of circular justifications. Thus, we define what are called *canonical loops* and *canonical loop formulas*. Given a dl-program, the models of its completion satisfying the canonical loop formulas constitute a new class of answer sets, called *canonical answer sets*, that are minimal and noncircular.

We also compare the canonical answer set semantics with the FLP-answer set semantics [9] for dl-programs and reveal that the former excludes certain circular justification of the latter, i.e., canonical answer sets are FLP-answer sets, but not vice versa in general. To study the relationships further, we map a dl-program to a logic program with abstract constraints [21]. We show that the answer sets (in the sense of [28]) of the logic program with abstract constraints mapped from a dl-program are canonical answer sets of the dl-program, but not vice versa. For dl-programs containing no nonmonotonic dl-atoms, all the above considered semantics coincide with each other, except for the weak answer set semantics.

This is a substantial revision and extension of [32] in two respects. First, the definitions of loop formulas for various semantics are simplified. Second, we present a spectrum of possible semantics for dl-programs, which include the well-known semantics based on weak and strong answer sets, the one based on canonical answer sets proposed in [32], and the semantics to be introduced and discussed in this paper, namely the one by mapping dl-programs to logic programs with abstract constraints and FLP-based semantics. In particular, we present a clear picture of how these semantics are related.

The paper is organized as follows. In the next section, we recall the basic definitions of description logics and dl-programs. In Section 3, we define completion, weak and strong loop formulas for dl-programs. The new semantics of dl-programs based on canonical loop formulas is given in Section 4. Section 5 discusses related work, and finally Section 6 gives concluding remarks and future work.

2. Preliminaries

In this section, we briefly review the basic notations for description logics and description logic programs [8].

2.1. Description logics

Description logics are a family of class-based (concept-based) knowledge representation formalisms. We assume a set $\mathbf{E}$ of *elementary datatypes* and a set $\mathbf{V}$ of *data values*. A *datatype theory* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a *datatype* (or *concrete*) *domain* $\Delta^{\mathbf{D}}$ and a mapping $\cdot^{\mathbf{D}}$ that assigns to every elementary datatype a subset of $\Delta^{\mathbf{D}}$ and to every data value an element of $\Delta^{\mathbf{D}}$. Let $\Psi = (\mathbf{A} \cup \mathbf{R}_A \cup \mathbf{R}_D, \mathbf{I} \cup \mathbf{V})$ be a vocabulary, where $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_D$, and $\mathbf{I}$ are pairwise disjoint (denumerable) sets of *atomic concepts, abstract roles, datatype* (or *concrete*) roles, and *individuals*, respectively.

A *role* is an element of $\mathbf{R_A} \cup \mathbf{R_A^-} \cup \mathbf{R_D}$, where $\mathbf{R}_A^-$ means the set of inverses of all $R \in \mathbf{R}_A$. *Concepts* are inductively defined as follows: (1) every atomic concept $C \in \mathbf{A}$ is a concept, (2) if $o_1, o_2, \ldots$ are individuals from $\mathbf{I}$, then $\{o_1, o_2, \ldots\}$ is a concept (called *oneOf*), (3) if C and D are concepts, then also $(C \sqcap D)$, $(C \sqcup D)$, and $\neg C$ are concepts (called *conjunction, disjunction*, and *negation* respectively). (4) if C is a concept, R is an abstract role from $\mathbf{R}_A \cup \mathbf{R}_A^-$, and n is a nonnegative integer, then $\exists R.C$, $\forall R.C$, $\geq nR$, and $\leq nR$ are concepts (called *exists, value, atleast*, and *atmost restriction*, respectively), (5) if D is a datatype, U is a datatype role from $\mathbf{R}_D$, and n is a nonnegative integer, then $\exists U.D$, $\forall U.D$, $\geq nU$, and $\leq nU$ are concepts (called *datatype exists, value, atleast*, and *atmost restriction*, respectively).

An *axiom* is an expression of one of the forms: (1) $C \sqsubseteq D$, called *concept inclusion axiom*, where C and D are concepts; (2) $R \sqsubseteq S$, called *role inclusion axiom*, where either $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_D$; (3) $\mathrm{Trans}(R)$, called *transitivity axiom*, where $R \in \mathbf{R}_A$; (4) $C(a)$, called *concept membership axiom*, where C is a concept and $a \in \mathbf{I}$; (5) $R(a, b)$ (resp., $U(a, v)$), called *role* *membership axiom* where $R \in \mathbf{R}_A$ (resp., $U \in \mathbf{R}_D$) $a, b \in \mathbf{I}$ (resp., $a \in \mathbf{I}$ and v is a data value), (6) $a \approx b$ (resp., $a \not\approx b$), called *equality* (resp., *inequality*) *axiom* where $a, b \in \mathbf{I}$.

A *description logic (DL) knowledge base O* is a finite set of axioms. The $\mathcal{SHOIN}(\mathbf{D})$ *knowledge base* consists of a finite set of above axioms, while the $\mathcal{SHIF}(\mathbf{D})$ *knowledge base* is the one of $\mathcal{SHOIN}(\mathbf{D})$, but without the *oneOf* constructor and with the *atleast* and *atmost* constructors limited to 0 and 1.

The semantics of the two description logics are defined in terms of general first-order interpretations. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ with respect to a datatype theory $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ consists of a nonempty (abstract) *domain* $\Delta^{\mathcal{I}}$ disjoint from $\Delta^{\mathbf{D}}$, and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $C \in \mathbf{A}$ a subset of $\Delta^{\mathcal{I}}$, to each individual $o \in \mathbf{I}$ an element of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{R}_A$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each datatype role $U \in \mathbf{R}_D$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathbf{D}}$. The mapping $\cdot^{\mathcal{I}}$ is extended to all concepts and roles as usual (where $\#S$ denotes the cardinality of a set S):

- $(R^-)^{\mathcal{I}} = \{(a, b) | (b, a) \in R^{\mathcal{I}}\}$;
- $\{o_1, \ldots, o_n\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \ldots, o_n^{\mathcal{I}}\}$;
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$, $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$;
- $(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y : (x, y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$;
- $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \forall y : (x, y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$;
- $(\geq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \#(\{y | (x, y) \in R^{\mathcal{I}}\}) \geq n\}$;
- $(\leq nR)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \#(\{y | (x, y) \in R^{\mathcal{I}}\}) \leq n\}$;
- $(\exists U.D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y : (x, y) \in U^{\mathcal{I}} \wedge y \in D^{\mathbf{D}}\}$;
- $(\forall U.D)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \forall y : (x, y) \in U^{\mathcal{I}} \rightarrow y \in D^{\mathbf{D}}\}$;
- $(\geq nU)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \#(\{y | (x, y) \in U^{\mathcal{I}}\}) \geq n\}$;
- $(\leq nU)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \#(\{y | (x, y) \in U^{\mathcal{I}}\}) \leq n\}$.

Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation respect to $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$, and F an axiom. We say that $\mathcal{I}$ *satisfies* F, written $\mathcal{I} \models F$, is defined as follows: (1) $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; (2) $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$; (3) $\mathcal{I} \models \mathrm{Trans}(R)$ iff $R^{\mathcal{I}}$ is transitive; (4) $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$; (5) $\mathcal{I} \models R(a, b)$ $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ (resp., $\mathcal{I} \models U(a, v)$ iff $(a^{\mathcal{I}}, v^{\mathbf{D}}) \in U^{\mathcal{I}}$); (6) $\mathcal{I} \models a \approx b$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$ (resp., $\mathcal{I} \models a \not\approx b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$). $\mathcal{I}$ *satisfies* a DL knowledge base O, written $\mathcal{I} \models O$, if $\mathcal{I} \models F$ for any $F \in O$. In this case, we call $\mathcal{I}$ a *model* of O. An axiom F is a *logical consequence* of a DL knowledge base O, written $O \models F$, if any model of O is also a model of F.

2.2. Description logic programs

Let $\Phi = (\mathcal{P}, \mathcal{C})$ be a first-order vocabulary with nonempty finite sets $\mathcal{C}$ and $\mathcal{P}$ of constant symbols and predicate symbols respectively such that $\mathcal{P}$ is disjoint from $\mathbf{A} \cup \mathbf{R}$ and $\mathcal{C} \subseteq \mathbf{I} \cup \mathbf{V}$. *Atoms* are formed from the symbols in $\mathcal{P}$ and $\mathcal{C}$ as usual.

A *dl-atom* is an expression of the form

$$DL[S_1\ op_1\ p_1, \ldots, S_m\ op_m\ p_m; Q](\vec{t}), \quad (m \geq 0) \tag{1}$$

where

- each S_i is either a concept, a role or its negation,[1] or a special symbol in $\{\approx, \not\approx\}$;
- $op_i \in \{\uplus, \cup\!\!\!\!-, \cap\!\!\!\!-\}$;
- p_i is a unary predicate symbol in $\mathcal{P}$ if S_i is a concept, and a binary predicate symbol in $\mathcal{P}$ otherwise. The p_is are called *input predicate symbols*;
- $Q(\vec{t})$ is a *dl-query*, i.e., either (1) $C(t)$ where $\vec{t} = t$; (2) $C \sqsubseteq D$ where $\vec{t}$ is an empty argument list; (3) $R(t_1, t_2)$ where $\vec{t} = (t_1, t_2)$; (4) $t_1 \approx t_2$ where $\vec{t} = (t_1, t_2)$; or their negations, where C and D are concepts, R is a role, and $\vec{t}$ is a tuple of constants.

The precise meanings of $\{\uplus, \cup\!\!\!\!-, \cap\!\!\!\!-\}$ will be defined shortly. Intuitively, $S \uplus p$ (resp. $S \cup\!\!\!\!- p$) extends S (resp. $\neg S$) by the extension of p, and $S \cap\!\!\!\!- p$ constrains S to p.

For example, suppose the interface is such that if any individual x is registered for a course (the information from outside an ontology) then x is a student (x may not be a student by the ontology before this communication), and we query if a is a student. We can then write the dl-atom $DL[Student \uplus registered; Student](a)$. Similarly, $DL[Student \cap\!\!\!\!- registered; \neg Student \sqcap \neg Employed](a)$ queries if a is not a student nor employed, with the ontology enhancement that if we cannot show x is registered, then x is not a student.

A *ground dl-rule* (or simply a *dl-rule* or *rule*) is an expression of the form

$$A \leftarrow B_1, \ldots, B_m, not\ B_{m+1}, \ldots, not\ B_n, (n \geq m \geq 0) \tag{2}$$

[1] We allow the negation of a role for convenience, so that we can replace "$S \cup\!\!\!\!- p$" with an equivalent form "$\neg S \uplus p$" in dl-atoms. The negation of a role is not allowed in [8].

where A is a ground atom, each B_i $(1 \le i \le n)$ is a ground atom[2] or a dl-atom. We refer to A as its *head*, while the conjunction of $B_i (1 \le i \le m)$ and *not* B_j $(m+1 \le j \le n)$ is its *body*. For convenience, we may abbreviate a rule in the form (2) as

$$A \leftarrow Pos, not\ Neg \tag{3}$$

where $Pos = \{B_1, \ldots, B_m\}$ and $Neg = \{B_{m+1}, \ldots, B_n\}$. Let r be a rule of the form (3). If $Neg = \emptyset$ and $Pos = \emptyset$, r is a fact and we may write it as "A" instead of "$A \leftarrow$". A *description logic program* (*dl-program*) $\mathcal{K} = (O, P)$ consists of a DL knowledge base O and a finite set P of dl-rules. In what follows we assume the vocabulary of P is implicitly given by the constant symbols and predicate symbols occurring in P, unless stated otherwise.

Given a dl-program $\mathcal{K} = (O, P)$, the *Herbrand base* of P, denoted by HB_P, is the set of atoms occurring in P and the ones formed from the predicate symbols of $\mathcal{P}$ occurring in some dl-atoms of P and the constant symbols in $\mathcal{C}$. Thus HB_P is in polynomial size of $\mathcal{K}$. An *interpretation* I (relative to P) is a subset of HB_P. Such an I is a *model* of an atom or dl-atom A under O, written $I \models_O A$, if the following holds:

- if $A \in HB_P$, then $I \models_O A$ iff $A \in I$;
- if A is a dl-atom $DL(\lambda; Q)(\vec{t})$ of the form (1), then $I \models_O A$ iff $O(I; \lambda) \models Q(\vec{t})$ where $O(I; \lambda) = O \cup \bigcup_{i=1}^{m} A_i(I)$ and, for $1 \le i \le m$,

$$A_i(I) = \begin{cases} \{S_i(\vec{e}) | p_i(\vec{e}) \in I\}, & \text{if } op_i = \uplus; \\ \{\neg S_i(\vec{e}) | p_i(\vec{e}) \in I\}, & \text{if } op_i = \cup\!\!\!\!-; \\ \{\neg S_i(\vec{e}) | p_i(\vec{e}) \notin I\}, & \text{if } op_i = \cap\!\!\!\!-; \end{cases}$$

where $\vec{e}$ is a tuple of constants over $\mathcal{C}$. As we allow negation of role, $S \cup\!\!\!\!- p$ can be replaced with $\neg S \uplus p$ in any dl-atom. In addition, we can shorten S_1 *op* $p, \ldots, S_k$ *op* p as $(S_1 \sqcup \cdots \sqcup S_k)$ *op* p where S_i *op* p appears in λ for all i $(1 \le i \le k)$ and $op \in \{\cap\!\!\!\!-, \uplus, \cup\!\!\!\!-\}$. Thus dl-atoms can be equivalently rewritten into ones without using the operator $\cup\!\!\!\!-$, and every predicate p appears at most once for each operator $\uplus$ and $\cap\!\!\!\!-$. For instance, the dl-atom $DL[S_1 \uplus p, S_2 \uplus p, S_1 \cap\!\!\!\!- p, S_2 \cap\!\!\!\!- p, Q](\vec{t})$ can be equivalently written as $DL[(S_1 \sqcup S_2) \uplus p, (S_1 \sqcup S_2) \cap\!\!\!\!- p, Q](\vec{t})$.

An interpretation I is a *model* of a dl-rule of the form (3) iff $I \models_O B$ for any $B \in Pos$ and $I \not\models_O B'$ for any $B' \in Neg$ implies $I \models_O A$. I is a *model* of a dl-program $\mathcal{K} = (O, P)$, written $I \models_O \mathcal{K}$, iff I is a model of each rule of P. I is a *supported model* of $\mathcal{K} = (O, P)$ iff, for any $h \in I$, there is a rule $(h \leftarrow Pos, not\ Neg)$ in P such that $I \models_O A$ for any $A \in Pos$ and $I \not\models_O B$ for any $B \in Neg$.

A dl-atom A is *monotonic* relative to a dl-program $\mathcal{K} = (O, P)$ if $I \models_O A$ implies $I' \models_O A$, for all $I \subseteq I' \subseteq HB_P$, otherwise A is *nonmonotonic*. If a dl-atom does not mention $\cap\!\!\!\!-$ then it is monotonic. However, a dl-atom may be monotonic even if it mentions $\cap\!\!\!\!-$. E.g., the dl-atom $DL[S \cup\!\!\!\!- p, S \cap\!\!\!\!- p; \neg S](a)$ is monotonic (which is a tautology). Clearly, the $\cap\!\!\!\!-$ operator is the only one that may cause a dl-atom to be nonmonotonic.

We use DL_P to denote the set of all dl-atoms that occur in P, $DL_P^+ \subseteq DL_P$ to denote the set of monotonic dl-atoms, and $DL_P^? = DL_P \setminus DL_P^+$. A dl-program $\mathcal{K} = (O, P)$ is *positive* if (i) P is "not"-free, and (ii) every dl-atom is monotonic relative to $\mathcal{K}$. It is evident that if a dl-program $\mathcal{K}$ is positive, then $\mathcal{K}$ has a (set inclusion) least model.

2.3. Strong and weak answer sets

We first recall the operator $\gamma_{\mathcal{K}} : 2^{HB_P} \to 2^{HB_P}$ for a positive dl-program $\mathcal{K}$, which is called $T_{\mathcal{K}}$ in [8]: let $I \subseteq HB_P$, and

$$\gamma_{\mathcal{K}}(I) = \{h | (h \leftarrow Pos) \in P \text{ and } I \models_O A \text{ for any } A \in Pos\}. \tag{4}$$

Since $\gamma_{\mathcal{K}}$ is monotonic, so it has the least fix-point which is the unique least model of $\mathcal{K}$. Such least fix-point can be iteratively constructed as

- $\gamma_{\mathcal{K}}^0 = \emptyset$;
- $\gamma_{\mathcal{K}}^{n+1} = \gamma_{\mathcal{K}}(\gamma_{\mathcal{K}}^n)$.

Let $\mathcal{K} = (O, P)$ be a dl-program. The *strong dl-transform* of $\mathcal{K}$ relative to O and an interpretation $I \subseteq HB_P$, denoted by $\mathcal{K}^{s,I}$, is the positive dl-program (O, sP_O^I), where sP_O^I is obtained from P by deleting

- the dl-rules of the form (2) such that either $I \not\models_O B_i$ for some i $(1 \le i \le m)$ and $B_i \in DL_P^?$, or $I \models_O B_j$ for some j $(m+1 \le j \le n)$; and
- the nonmonotonic dl-atoms and *not* A from the remaining dl-rules where A is an atom or dl-atom.

The interpretation I is a *strong answer set* of $\mathcal{K}$ if it is the least model of $\mathcal{K}^{s,I}$.

The *weak dl-transform* of $\mathcal{K}$ relative to O and an interpretation $I \subseteq HB_P$, denoted by $\mathcal{K}^{w,I}$, is the positive dl-program (O, wP_O^I), where wP_O^I is obtained from P by deleting

[2] Different from [8], in this paper we consider ground atoms instead of literals for convenience.

- the dl-rules of the form (2) such that either $I \not\models_O B_i$ for some i $(1 \leq i \leq m)$ and $B_i \in DL_P$, or $I \models_O B_j$ for some j $(m+1 \leq j \leq n)$; and
- the dl-atoms and *not* A from the remaining dl-rules where A is an atom or dl-atom.

The interpretation I is a *weak answer set* of $\mathcal{K}$ if I is the least model of $\mathcal{K}^{w,I}$.

For a positive dl-program $\mathcal{K}$, $\gamma_{\mathcal{K}}$ is monotonic and it has the least fix-point, denoted by $lfp(\gamma_{\mathcal{K}})$. So $I \subseteq HB_P$ is a strong (resp., weak) answer set of a dl-program $\mathcal{K} = (O, P)$ if and only if $I = lfp(\gamma_{\mathcal{K}^{s,I}})$ (resp., $I = lfp(\gamma_{\mathcal{K}^{w,I}})$).

Proposition 2.1. *Let $\mathcal{K} = (O, P)$ be a dl-program and $\mathcal{K}'$ be the dl-program (O, P') where P' is obtained from P by replacing each "$S \cup p$" with "$\neg S \uplus p$" in the dl-atoms of P. We have that, for any interpretation $I \subseteq HB_P$, I is a weak (resp. strong) answer set of $\mathcal{K}$ if and only if I is a weak (resp. strong) answer set of $\mathcal{K}'$.*

Proof. It is evident by definition. □

Example 1. Consider the following dl-programs:

- $\mathcal{K}_0 = (O, P_0)$ where $O = \{c \sqsubseteq c'\}$ and $P_0 = \{w(a) \leftarrow DL[c \uplus p; c'](a);$ $p(a) \leftarrow\}$. For this dl-program to make some sense, let us image this situation: c' and c are classes of good conference papers and ICLP papers respectively, $p(x)$ means that x is a paper in the TPLP special issue of ICLP 2010, $w(x)$ means that x is worth reading, and a stands for "this paper".[3] Note that c and c' are concepts in O, and p and w are predicates outside of O. The communication is through the dl-rule, $w(a) \leftarrow DL[c \uplus p; c'](a)$, which says that if "this paper" is a good conference paper, given that any paper in the TPLP special issue of ICLP 2010 is an ICLP paper and ICLP papers are good conference papers (by the knowledge in O), then it is worth reading. $\mathcal{K}_0$ has exactly one strong answer set $\{p(a), w(a)\}$, which is also its unique weak answer set.
- Now, suppose someone writes $\mathcal{K}_1 = (O, P_1)$ where $O = \{c \sqsubseteq c'\}$ and $P_1 = \{p(a) \leftarrow DL[c \uplus p; c'](a)\}$. This program has a unique strong answer set $I_1 = \emptyset$ and two weak answer sets I_1 and $I_2 = \{p(a)\}$. It can be seen that there is a circular justification in the weak answer set I_2: that "this paper" is in the TPLP special issue of ICLP 2010 is justified by the fact that it is in it.

  The interested reader may verify the following. By the definition of $\uplus$, $O(I_2; c \uplus p) = O \cup \{c(a)\}$, and clearly $O \not\models c'(a)$ and $\{c(a), c \sqsubseteq c'\} \models c'(a)$. So the weak dl-transform relative to O and I_2 is $\mathcal{K}_1^{w,I_2} = (O, \{p(a) \leftarrow\})$. Since I_2 coincides with the least model of $\{p(a) \leftarrow\}$, it is a weak answer set of $\mathcal{K}_1$. Similarly, one can verify that the strong dl-transform relative to O and I_2 is $\mathcal{K}_1^{s,I_2} = (O, P_1)$. Its least model is the empty set, so I_2 is not a strong answer set of $\mathcal{K}_1$.
- $\mathcal{K}_2 = (O, P_2)$ where $O = \emptyset$ and $P_2 = \{p(a) \leftarrow DL[c \uplus p, b \cap\!\!\!\!- q; c \sqcap \neg b](a)\}$. Both $\emptyset$ and $\{p(a)\}$ are strong and weak answer sets of the dl-program. However, the atom $p(a)$ has only circular justification from $\{p(a)\}$.

These dl-programs show that strong (and weak) answer sets may not be (set inclusion) minimal. It has been shown that if a dl-program contains no nonmonotonic dl-atoms then its strong answer sets are minimal (Theorem 4.13 of [8]). However, this does not hold for weak answer sets as shown by the dl-program $\mathcal{K}_1$ above, even if it is positive. It is known that strong answer sets are always weak answer sets, but not vice versa (Theorem 4.23 of [8]).

3. Completion and loop formulas

In this section, we define completion, characterize weak and strong answer sets by loop formulas, and outline an alternative method of computing weak and strong answer sets.

3.1. Completion

Given a dl-program $\mathcal{K} = (O, P)$, we assume an underlying propositional language $\mathcal{L}_{\mathcal{K}}$, such that the propositional atoms of $\mathcal{L}_{\mathcal{K}}$ include the atoms and dl-atoms occurring in P. The *formulas* of $\mathcal{L}_{\mathcal{K}}$ are defined as usual using the connectives $\neg, \wedge, \vee, \supset$ and $\leftrightarrow$. The *dl-interpretations* (or simply *interpretations* if it is clear from context) of the language $\mathcal{L}_{\mathcal{K}}$ are the interpretations relative to P, i.e., the subsets of HB_P. For a formula ψ of $\mathcal{L}_{\mathcal{K}}$ and an interpretation I of $\mathcal{L}_{\mathcal{K}}$, we say I is a *model* of ψ relative to O, denoted $I \models_O \psi$, whenever (i) if ψ is an atom or a dl-atom, then it is defined as previous, and (ii) it is extended in the usual way to connectives $\vee, \wedge, \supset$ and so on.

Let $\mathcal{K} = (O, P)$ be a dl-program and h an atom in HB_P. The *completion* of h (relative to $\mathcal{K}$), written $COMP(h, \mathcal{K})$, is the following formula of $\mathcal{L}_{\mathcal{K}}$:

$$h \leftrightarrow \bigvee_{1 \leq i \leq n} \left(\bigwedge_{A \in Pos_i} A \wedge \bigwedge_{B \in Neg_i} \neg B \right),$$

[3] The full papers presented at ICLP 2010 are published in a TPLP special issue, and the short version of this paper appeared in that special issue.

where $(h \leftarrow Pos_1, not\ Neg_1), \ldots, (h \leftarrow Pos_n, not\ Neg_n)$ are all the rules in P whose heads are the atom h. The *completion* of $\mathcal{K}$, written $COMP(\mathcal{K})$, is the collection of completions of all atoms in HB_P.

Recall that a model $M \subseteq HB_P$ of a dl-program $\mathcal{K} = (O, P)$ is a *supported model* if for any atom $a \in M$, there is a rule in P whose head is a and whose body is satisfied by M.

Proposition 3.1. *Let $\mathcal{K} = (O, P)$ be a dl-program and I an interpretation of P. Then I is a supported model of $\mathcal{K}$ if and only if $I \models_O COMP(\mathcal{K})$.*

Proof. The interpretation I is a supported model of $\mathcal{K}$
iff, for any $h \in I$, there exists a rule $(h \leftarrow Pos, not\ Neg)$ in P such that

$$I \models_O \left(\bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B \right)$$

iff $I \models_O COMP(h, \mathcal{K})$ for any $h \in I$
iff $I \models_O COMP(\mathcal{K})$. □

Proposition 3.2. *Every weak (resp. strong) answer set of a dl-program $\mathcal{K}$ is a supported model of $\mathcal{K}$.*

Proof. (1) Let I be a strong answer set of $\mathcal{K}$. It is sufficient to show that, for any $h \in I$, $I \models_O COMP(h, \mathcal{K})$ by Proposition 3.1. We have that
$h \in I$
$\Rightarrow$ there is a dl-rule $(r' : h \leftarrow Pos_1)$ in sP^I_O such that $I \models_O A$ for any $A \in Pos_1$
$\Rightarrow$ there is a dl-rule $(r : h \leftarrow Pos_1, Pos_2, not\ Neg)$ in P such that r' is obtained from r by the strong dl-transformation, where Pos_2 is a set of nonmonotonic dl-atoms, i.e., (i) $I \models_O B$ for any $B \in Pos_2$, and (ii) $I \not\models_O B'$ for any $B' \in Neg$
$\Rightarrow I \models_O \bigwedge_{A \in Pos_1 \cup Pos_2} A \wedge \bigwedge_{B \in Neg} \neg B$.

Consequently, I is a supported model of $\mathcal{K}$.

(2) The proof is similar when I is a weak answer set of $\mathcal{K}$. □

3.2. Weak loop formulas

In order to capture weak answer sets of dl-programs using completion and loop formulas, we define weak loops. Formally, let $\mathcal{K} = (O, P)$ be a dl-program. The *weak positive dependency graph* of $\mathcal{K}$, written $G^w_{\mathcal{K}}$, is the directed graph (V, E), where $V = HB_P$ (note that a dl-atom is not in V), and $(u, v) \in E$ if there is a dl-rule of the form (2) in P such that $A = u$ and $B_i = v$ for some i $(1 \le i \le m)$. A nonempty subset L of HB_P is a *weak loop* of $\mathcal{K}$ if there is a cycle in $G^w_{\mathcal{K}}$ which goes through only and all the nodes in L. The nodes on the cycle may repeat. A loop L is *maximal* if there is no loop L' such that $L \subset L'$, i.e., a strongly connected component of $G^w_{\mathcal{K}}$. A maximal loop L is *terminating* if there is no path from one node of L to any other maximal loop.

Given a weak loop L of a dl-program $\mathcal{K} = (O, P)$, the *weak loop formula* of L (relative to $\mathcal{K}$), written $wLF(L, \mathcal{K})$, is the following formula of $\mathcal{L}_{\mathcal{K}}$:

$$\bigvee L \supset \bigvee_{1 \le i \le n} \left(\bigwedge_{A \in Pos_i} A \wedge \bigwedge_{B \in Neg_i} \neg B \right)$$

where $(h_1 \leftarrow Pos_1, not\ Neg_1), \ldots, (h_n \leftarrow Pos_n, not\ Neg_n)$ are all the rules in P such that $h_i \in L$ and $Pos_i \cap L = \emptyset$ for any i $(1 \le i \le n)$.

Theorem 3.3. *Let $\mathcal{K} = (O, P)$ be a dl-program and I an interpretation of P. Then I is a weak answer set of $\mathcal{K}$ if and only if $I \models_O COMP(\mathcal{K}) \cup wLF(\mathcal{K})$, where $wLF(\mathcal{K})$ is the set of weak loop formulas of all weak loops of $\mathcal{K}$.*

Proof. $(\Rightarrow)$ By Proposition 3.2, we only need to show that $I \models_O wLF(L, \mathcal{K})$ for any weak loop L of $\mathcal{K}$. Suppose $I \not\models_O wLF(L, \mathcal{K})$, i.e.,

$$I \models_O \bigvee L \quad \text{and} \quad I \not\models_O \left(\bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B \right) \tag{5}$$

for any rule $(h \leftarrow Pos, not\ Neg)$ in P such that $h \in L$ and $Pos \cap L = \emptyset$. It implies that $I \cap L \neq \emptyset$. Without loss of generality, suppose $L = \{h_1, \ldots, h_k\}$ and $h_1 \in I \cap L$. Because I is a weak answer set of $\mathcal{K}$, $I = lfp(\gamma_{\mathcal{K}^{w,I}})$. It follows that $h_1 \in lfp(\gamma_{\mathcal{K}^{w,I}})$. Let k_1 be the least number such that $h_1 \in \gamma^{k_1+1}_{\mathcal{K}^{w,I}}$. Thus wP^I_O must have a rule

$$r_1 : h_1 \leftarrow Pos_1$$

such that $\gamma^{k_1}_{\mathcal{K}^{w,I}} \models_O A$ for any $A \in Pos_1$. Suppose r_1 is obtained from the following rule

$$h_1 \leftarrow Pos_1, Adl_1, not\ Neg_1$$

in P by the weak dl-transformation, where Adl_1 is a set of dl-atoms. Thus $I \models_O A$ for any $A \in Adl_1$ and $I \not\models_O B$ for any $B \in Neg_1$. By (5) we have $Pos_1 \cap L \neq \emptyset$. From $h_1 \notin Pos_1$, it follows $(L \setminus \{h_1\}) \cap Pos_1 \neq \emptyset$. Without loss of generality, suppose $h_2 \in Pos_1$. Similarly, there exists the least number k_2 such that $h_2 \in \gamma^{k_2+1}_{\mathcal{K}^{w,I}}$. Using this construction, we can get a sequence $(k_1, k_2, \dots,)$ of natural numbers and a sequence $(h_1, h_2 \dots,)$ of atoms in $L \cap I$ such that, for any $i \geq 1$,

- k_i is the smallest number such that $h_i \in \gamma^{k_i+1}_{\mathcal{K}^{w,I}}$,
- $(h_i \leftarrow Pos_i)$ is a rule in wP^I_O such that $Pos_i \subseteq \gamma^{k_i}_{\mathcal{K}^{w,I}}$, and
- $k_i > k_j$ for any $j > i$.

Since $I \cap L$ is finite, there must be some i, j $(0 \leq i < j)$ such that $h_i = h_j$. This implies that $k_i = k_j$. This is a paradox. Thus $I \models_O wLF(L, \mathcal{K})$.

($\Leftarrow$) First, we show $I \subseteq lfp(\gamma_{\mathcal{K}^{w,I}})$. Let Γ be the set of rules in wP^I_O whose bodies are satisfied by I. Since I is a supported model of $\mathcal{K}$, the heads of rules in Γ are also satisfied by I. Moreover, I is the set of atoms occurring in Γ. Let $I^* = lfp(\gamma_{\mathcal{K}^{w,I}_\Gamma})$, where $\mathcal{K}_\Gamma = (O, \Gamma)$. Let $I^- = I \setminus I^*$ and Γ_{I^-} be the set of rules in Γ whose heads are in I^-. If $I^- = \emptyset$, i.e., $\Gamma_{I^-} = \emptyset$ then $I \subseteq lfp(\gamma_{\mathcal{K}^{w,I}})$ since $I \subseteq I^*$ and $I^* \subseteq lfp(\gamma_{\mathcal{K}^{w,I}})$. Suppose $I^- \neq \emptyset$. We show that (O, Γ_{I^-}) has at least one terminating loop.

For any rule $(r : h \leftarrow Pos)$ in Γ_{I^-}, $Pos \subseteq I$ since $I \models A$ for any $A \in Pos$ and wP^I_O mentions only atoms. However $Pos \setminus I^* \neq \emptyset$ otherwise $I^* \models A$ for any $A \in Pos$ and then $r \notin \Gamma_{I^-}$. It follows that

$$Pos \cap (I \setminus I^*) \neq \emptyset.$$

Suppose $h' \in Pos \cap I^-$. Then there is an edge (h, h') in the weak positive dependency graph of (O, Γ_{I^-}). So we can construct a sequence of atoms

$$(h_1, h_2, \dots, h_i, \dots)$$

such that $h_i \in I^-$ for any $i \geq 1$ and (h_i, h_{i+1}) is an edge of the weak positive dependency graph of (O, Γ_{I^-}). Since I^- is finite, the above sequence must contain a loop. If a graph has a loop, then it has at least one terminating loop. Now suppose $L = \{h_1, \dots, h_k\}$ is a terminating loop of Γ_{I^-}. We further claim that, for any rule $(h \leftarrow Pos)$ in Γ_{I^-} such that $h \in L$:

$$I^- \cap Pos \subseteq L.$$

Otherwise, we can construct another path $(h, h', \dots)$ in the positive weak dependency graph of (O, Γ_{I^-}) such that $h' \in I \cap Pos$ and $h' \notin L$. Thus we have a path from L to another maximal loop of the weak dependency graph of (O, Γ_{I^-}), which contradicts the fact that L is a terminating loop.

Because L is also a weak loop of $\mathcal{K}$, $I \models_O wLF(L, \mathcal{K})$ and $L \subseteq I^-$, it follows that P should have at least one rule

$$r' : h' \leftarrow Pos', not\ Neg'$$

such that $h' \in L$, $Pos' \cap L = \emptyset$, $I \models_O A$ for any $A \in Pos'$ and $I \not\models_O B$ for any $B \in Neg'$. Suppose $(r^* : h' \leftarrow Pos^*)$ is the rule obtained from r' by the weak dl-transformation. Evidently, $r^* \in \Gamma$. Furthermore $r^* \in \Gamma_{I^-}$ since $h' \in L \subseteq I^-$. This implies that $I^- \cap Pos^* \subseteq L$ which contradicts $Pos^* \cap L = \emptyset$ since $I^- \cap Pos^* \neq \emptyset$. Consequently, $I^- = \emptyset$ and then $I \subseteq I^* = lfp(\gamma_{\mathcal{K}^{w,I}_\Gamma})$. It follows that $I \subseteq lfp(\gamma_{\mathcal{K}^{w,I}_\Gamma}) \subseteq lfp(\gamma_{\mathcal{K}^{w,I}})$ by $\Gamma \subseteq wP^I_O$.

Second, we prove $lfp(\gamma_{\mathcal{K}^{w,I}}) \subseteq I$. Let $I' = lfp(\gamma_{\mathcal{K}^{w,I}}) \setminus I$. Suppose $I' \neq \emptyset$. Let h be an arbitrary atom in I'. There is the least number k such that $h \in \gamma^{k+1}_{\mathcal{K}^{w,I}}$. So there exists a rule $(r' : h \leftarrow Pos')$ in wP^I_O such that $Pos' \subseteq \gamma^k_{\mathcal{K}^{w,I}}$. Since $h \notin I$ and $I \models_O COMP(h, \mathcal{K})$, we have that, for any rule $(h \leftarrow Pos, not\ Neg)$ in P,

$$I \not\models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B.$$

It follows $I \setminus Pos \neq \emptyset$. Thus there exists an atom $h' \in Pos$ such that $h \in \gamma^k_{\mathcal{K}^{w,I}} \setminus I$. So we can construct a sequence of numbers $(k_1, k_2, \dots)$ and a sequence $(h_1, h_2, \dots)$ of atoms in I' such that, for any $i \geq 1$,

- k_i is the least number such that $h_i \in \gamma^{k_i+1}_{\mathcal{K}^{w,I}}$,
- $(h_i \leftarrow Pos_i)$ is a rule in wP^I_O such that $Pos_i \subseteq \gamma^{k_i}_{\mathcal{K}^{w,I}}$, and
- $k_i > k_j$ for any $j > i$.

Since I' is finite, there exists $i, j (0 \leq i < j)$ such that $h_i = h_j$ which implies that $k_i = k_j$. It contradicts $k_i > k_j$. Thus $I' = \emptyset$, i.e., $lfp(\gamma_{\mathcal{K}^{w,I}}) \subseteq I$.

Consequently I is a weak answer set of $\mathcal{K}$. $\square$

3.3. Strong loop formulas

Let $\mathcal{K} = (O, P)$ be a dl-program. The *strong positive dependency graph* of $\mathcal{K}$, denoted by $G^s_{\mathcal{K}}$, is the directed graph (V, E), where $V = HB_P$ and $(p(\vec{c}), q(\vec{c}\,')) \in E$ if there is a rule of the form (2) in P such that, (1) $A = p(\vec{c})$ and, (2) for some $i\ (1 \le i \le m)$, either

- $B_i = q(\vec{c}\,')$, or
- B_i is a monotonic dl-atom mentioning the predicate q and $\vec{c}\,'$ is a tuple of constants matching the arity of q. (If this condition is ignored then it becomes the definition of weak positive dependency graph.)

A nonempty subset L of HB_P is a *strong loop* of $\mathcal{K}$ if there is a cycle in $G^s_{\mathcal{K}}$ which passes only and all the nodes in L.

To define strong loop formulas of a dl-program $\mathcal{K} = (O, P)$, we need to extend the vocabulary Φ, such that, for any predicate symbol p and a nonempty set of atoms L, Φ contains the predicate symbol p_L that has the same arity as that of p. Given an interpretation I of $\mathcal{K}$, the *loop extension* of I relative to $\mathcal{K}$, written $LE(I, \mathcal{K})$, is the set of atoms in I and $p_L(\vec{c})$ where p is a predicate occurring in some dl-atom of P, $L \subseteq HB_P$ and $L \neq \emptyset$, and $p(\vec{c}) \in I \setminus L$. In other words, $LE(I, \mathcal{K}) = I \cup \{p_L(\vec{c}) | p(\vec{c}) \in I \setminus L, L \neq \emptyset \text{ and } L \subseteq HB_P\}$.

Let L be a nonempty set of atoms and let $A = DL[\lambda; Q](\vec{t})$ be a dl-atom. The *irrelevant formula* of A relative to L, written by $IF(A, L)$ is $DL[\lambda_L; Q](\vec{t})$ where λ_L is obtained from λ by replacing each predicate symbol p with p_L if $p(\vec{c}) \in L$ for some $\vec{c}$.

We are now in a position to define strong loop formulas. Let L be a strong loop of $\mathcal{K} = (O, P)$. The *strong loop formula* of L (relative to $\mathcal{K}$), written $sLF(L, \mathcal{K})$, is the following formula of $\mathcal{L}_{\mathcal{K}}$:

$$\bigvee L \supset \bigvee_{1 \le i \le n} \left(\bigwedge_{A \in Pos_i} \gamma(A, L) \wedge \bigwedge_{B \in Neg_i} \neg B \right)$$

where

- $(h_1 \leftarrow Pos_1, not\ Neg_1), \ldots, (h_n \leftarrow Pos_n, not\ Neg_n)$ are all the rules in P such that $h_i \in L$ and $Pos_i \cap L = \emptyset$ for all $i\ (1 \le i \le n)$,
- $\gamma(A, L) = IF(A, L)$ if $A \in DL^+_P$, and A otherwise.

In general, we have to recognize the monotonicity of dl-atoms in order to construct strong loops of dl-programs. In this sense, the strong loops and strong loop formulas are defined semantically. If a dl-atom does not mention the operator $\cap\!\!\!\!-$ then it is monotonic. Thus for the class of dl-programs in which no monotonic dl-atoms mention $\cap\!\!\!\!-$, the strong loops and strong loop formulas are given syntactically, since it is sufficient to determine the monotonicity of a dl-atom by checking whether it contains the operator $\cap\!\!\!\!-$.

Example 2. Let $\mathcal{K} = (\emptyset, P)$ be a dl-program where P consists of

$$p(a) \leftarrow DL[c \uplus p; c](a); \qquad p(a) \leftarrow not\ DL[c \uplus p; c](a).$$

The dl-program $\mathcal{K}$ has a unique strong loop $L = \{p(a)\}$, but does not have any weak loops. Its completion is the formula

$$p(a) \leftrightarrow DL[c \uplus p; c](a) \vee \neg DL[c \uplus p; c](a)$$

which is equal to the formula $p(a) \leftrightarrow \top$, i.e., $p(a)$. The strong loop formula $sLF(L, \mathcal{K})$ is the formula

$$p(a) \supset DL[c \uplus p_L; c](a) \vee \neg DL[c \uplus p; c](a).$$

The interpretation $I = \{p(a)\}$ is a model of $COMP(\mathcal{K})$ relative to the knowledge base $O = \emptyset$. However, $LE(I, \mathcal{K}) \not\models_O sLF(L, \mathcal{K})$ since $p(a) \in LE(I, \mathcal{K})$ and $p_L(a) \notin LE(I, \mathcal{K})$.

Lemma 3.4. *Let $\mathcal{K} = (O, P)$ be a dl-program, $I \subseteq HB_P$ and L be an arbitrary nonempty set of atoms. Then we have if A is a dl-atom in DL_P then $LE(I, \mathcal{K}) \models_O IF(A, L)$ iff $I \setminus L \models_O A$.*

Proof. Let $I' = LE(I, \mathcal{K})$. We have that $p(\vec{c}) \in I$ iff $p(\vec{c}) \in I'$ for any $p(\vec{c}) \in HB_P$. Furthermore, for any atom $p_L(\vec{c})$, $p_L(\vec{c}) \in I'$ iff $p(\vec{c}) \in I \setminus L$. For clarity and without loss of generality, let $A = DL[S \uplus p, S' \cap\!\!\!\!-\, q; Q](\vec{t})$. The statement in (ii) obviously holds if the predicates p and q do not occur in L, since $IF(A, L) = A$. Let us assume that the predicates p and q appear in L.
$I' \models_O IF(A, L)$
$\Leftrightarrow I' \models_O DL[S \uplus p_L, S' \cap\!\!\!\!-\, q_L; Q](\vec{t})$
$\Leftrightarrow O \cup \{S(\vec{e}) | p_L(\vec{e}) \in I'\} \cup \{\neg S'(\vec{e}) | q_L(\vec{e}) \notin I'\} \models Q(\vec{t})$
$\Leftrightarrow O \cup \{S(\vec{e}) | p(\vec{e}) \in I \setminus L\} \cup \{\neg S'(\vec{e}) | q(\vec{e}) \notin I \setminus L\} \models Q(\vec{t})$
$\Leftrightarrow I \setminus L \models_O DL[S \uplus p, S' \cap\!\!\!\!-\, q; Q](\vec{t})$
$\Leftrightarrow I \setminus L \models_O A$.

The other two cases, namely (a) p appears in L but not q, and (b) q appears in L but not p, can be similarly proved. The proof can be easily extended to the case $A = DL[S_1 \uplus p_1, \ldots, S_m \uplus p_m, S'_1 \cap\!\!\!\!-\, q_1, \ldots, S'_n \cap\!\!\!\!-\, q_n; Q](\vec{t})$. $\Box$

Lemma 3.5. *Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$ such that $I \models_O COMP(\mathcal{K})$. Then we have that $lfp(\gamma_{\mathcal{K}^{s,I}}) \subseteq I$.*

Proof. Let $I' = lfp(\gamma_{\mathcal{K}^{s,I}})$ and $I^- = I' \setminus I$. Suppose $I' \not\subseteq I$ then $I^- \neq \emptyset$. For any $h \in I^-$, there exists the least natural number k and a rule $(r : h \leftarrow Pos)$ in sP^I_O such that $\gamma^k_{\mathcal{K}^{s,I}} \models_O A$ for any $A \in Pos$. But we know that $h \notin I$ and $I \models_O COMP(h, \mathcal{K})$ which implies that, for any rule $(r' : h \leftarrow Pos', not\ Neg')$ in P:

$$I \not\models_O \bigwedge_{A \in Pos'} A \wedge \bigwedge_{B \in Neg'} \neg B.$$

It follows that $I \not\models_O A$ for some $A \in Pos$. It follows that either

(i) there is an atom $h' \in Pos \cap \gamma^k_{\mathcal{K}^{s,I}}$ such that $h' \notin I$, or
(ii) there is a monotonic dl-atom $A = DL[\lambda; Q](\vec{t})$ in Pos such that, for some $S \uplus p$ (or $S \cup\!\!\!\!- p$) occurring in λ, there is an atom $h' = p(\vec{c})$ such that $h' \in \gamma^k_{\mathcal{K}^{s,I}}$ and $p(\vec{c}) \notin I$.

It is evident that $h' \neq h$ and $h' \in I^-$. Thus we can construct a sequence $(k_1, k_1, \ldots)$ of natural numbers and a sequence $(h_1, h_2, \ldots)$ of atoms in I^- such that, for any $i \geq 1$,

- k_i is the least number such that $h_i \in \gamma^{k_i+1}_{\mathcal{K}^{s,I}}$,
- $(h_i \leftarrow Pos_i)$ is in sP^I_O such that $\gamma^{k_i}_{\mathcal{K}^{s,I}} \models_O A$ for any $A \in Pos_i$, and
- $k_i > k_j$ for any $j > i$.

Since I^- is finite, in the above sequence of atoms there must exist i, j $(0 \leq i < j)$ such that $h_i = h_j$. It follows that $k_i = k_j$ which contradicts $k_i > k_j$. Consequently, $I^- = \emptyset$, i.e., $I' \subseteq I$. $\Box$

Theorem 3.6. *Let $\mathcal{K} = (O, P)$ be a dl-program and I an interpretation of P. Then we have that I is a strong answer set of $\mathcal{K}$ if and only if $LE(I, \mathcal{K}) \models_O COMP(\mathcal{K}) \cup sLF(\mathcal{K})$, where $sLF(\mathcal{K})$ is the set of strong loop formulas of all strong loops of $\mathcal{K}$.*

Proof. Let $I' = LE(I, \mathcal{K})$. Clearly $I' \cap HB_P = I$.

($\Rightarrow$) By Proposition 3.2 we have that I is a supported model, and by Proposition 3.1 follows $I \models_O COMP(\mathcal{K})$ and then clearly $LE(I, \mathcal{K}) \models_O COMP(\mathcal{K})$. It thus remains to prove that, for any strong loop L of $\mathcal{K}$, $I' \models_O sLF(L, \mathcal{K})$. Suppose $L = \{h_1, \ldots, h_k\}$ is a strong loop of $\mathcal{K}$ and $I' \not\models_O sLF(L, \mathcal{K})$, i.e.,

$$I' \models_O \bigvee L \quad \text{and} \quad I' \not\models_O \bigvee_{1 \leq i \leq n} \left(\bigwedge_{A \in Pos_i} \gamma(A, L) \wedge \bigwedge_{B \in Neg_i} \neg B \right)$$

where $(h_1 \leftarrow Pos_1, not\ Neg_1), \ldots, (h_n \leftarrow Pos_n, not\ Neg_n)$ are all the rules in P such that $h_i \in L$ and $Pos_i \cap L = \emptyset$ for any i $(1 \leq i \leq n)$. It follows that, for any i $(1 \leq i \leq n)$,

$$I' \not\models_O \bigwedge_{A \in Pos_i} \gamma(A, L) \wedge \bigwedge_{B \in Neg_i} \neg B. \tag{6}$$

Since $I' \models_O \bigvee L$, we have that $I' \cap L \neq \emptyset$ and thus $I \cap L \neq \emptyset$. Without loss of generality, let us assume $h_1 \in I \cap L$. Recall that I is a strong answer set of $\mathcal{K}$, i.e., $I = lfp(\gamma_{\mathcal{K}^{s,I}})$. Thus there is the least number k_1 such that $h_1 \in \gamma^{k_1+1}_{\mathcal{K}^{s,I}}$. So there is a rule $(r_1 : h_1 \leftarrow Pos_1)$ in sP^I_O such that $\gamma^{k_1}_{\mathcal{K}^{s,I}} \models_O A$ for any $A \in Pos_1$. It is evident that $h_1 \notin Pos_1$. It follows that P has a rule

$$r'_1 : h_1 \leftarrow Pos_1, Ndl_1, not\ Neg_1,$$

where Ndl_1 is a set of nonmonotonic dl-atoms, such that r_1 is obtained from r'_1 by the strong dl-transformation, i.e., $I \models_O A$ for any $A \in Ndl_1$ and $I \not\models_O B$ for any $B \in Neg_1$. It is clear that, $I' \models_O A$ for each $A \in Ndl_1$ and $I' \not\models_O B$ for any $B \in Neg_1$ due to $I' \cap HB_P = I$. By (6), at least one of the following two cases holds:

- $Pos_1 \cap L \neq \emptyset$. In this case, there is an atom $h \in Pos_1 \cap L$ and $h \neq h_1$.
- $I' \not\models_O IF(A, L)$ for some monotonic dl-atom $A = DL[\lambda; Q](\vec{t})$ in Pos_1. By Lemma 3.4, we have $I \setminus L \not\models_O A$. Since A is monotonic, then we further have $\gamma^{k_1}_{\mathcal{K}^{s,I}} \setminus L \not\models_O A$. But we know that $\gamma^{k_1}_{\mathcal{K}^{s,I}} \models_O A$. It follows that, there exists an atom $p(\vec{c}) \in L \cap \gamma^{k_1}_{\mathcal{K}^{s,I}}$, $p(\vec{c}) \neq h_1$ and $S \uplus p$ (or $S \cup\!\!\!\!- p$) appears in L for some S.

By the above analysis, we can have a sequence of natural numbers $(k_1, k_2, \ldots,)$ and a sequence $(h_1, h_2, \ldots)$ of atoms in L such that, for any $i \geq 1$,

- k_i is the least natural number such that $h_i \in \gamma^{k_i+1}_{\mathcal{K}^{s,I}}$,
- $(h_i \leftarrow Pos_i)$ is the rule in sP^I_O such that $\gamma^{k_i+1}_{\mathcal{K}^{s,I}} \models_O A$ for any $A \in Pos_i$, and
- $k_i > k_j$ for any $j > i$.

Since L is finite, there must be some i, j ($1 \le i < j$) such that $h_i = h_j$, which implies that $k_i = k_j$. This is a contradiction. Consequently, $I' \models_O sLF(L, \mathcal{K})$.

($\Leftarrow$) Let $I = I' \cap HB_P$. By Proposition 3.1, I is a supported model of $\mathcal{K}$. Let Γ be the set of rules in sP_O^I whose bodies are satisfied by I relative to O. Clearly, for any rule $(h \leftarrow Pos)$ in Γ, $h \in I$. And conversely, for any $h \in I$, there exists at least one rule $(h \leftarrow Pos)$ in Γ. Let $I^* = lfp(\gamma_{\mathcal{K}_\Gamma^{s,I}})$ where $\mathcal{K}_\Gamma = (O, \Gamma)$. Evidently, $I^* \subseteq I$. Let $I^- = I \setminus I^*$. Suppose $I^- \neq \emptyset$. Let Γ_{I^-} be the set of rules in Γ whose heads belong to I^-. We claim that the dl-program (O, Γ_{I^-}) must have one terminating loop.

Let $h \in I^-$ and suppose $(h \leftarrow Pos)$ is a rule in Γ_{I^-}. We have that

$$I^* \not\models_O \bigwedge_{A \in Pos} A \quad \text{and} \quad I^* \cup I^- \models_O \bigwedge_{A \in Pos} A.$$

It follows that there is an atom or dl-atom A in Pos such that $I^* \not\models_O A$. This implies that at least one of the following cases holds:

- there is an atom $h' \in Pos$, $h' \in I^-$;
- there exists a monotonic dl-atom $A = DL[\lambda; Q](\vec{t})$ in Pos such that $I^* \not\models_O A$, which implies that there exists some $S \uplus p$ (or $S \cup\!\!\!\!- \, p$) appearing in λ and $p(\vec{c}) \in I \setminus I^*$ for some atom $p(\vec{c})$ since $I \models_O A$, otherwise $I^* \models_O A$.

It follows that there exists an edge (h, h') in the positive strong dependency graph G of the dl-program (O, Γ_{I^-}) where $h' \in I^-$. Consequently, we can construct a sequence

$$(h_0, h_1, \ldots, h_i, \ldots)$$

of atoms in I^- such that, for any $i \ge 0$, (h_i, h_{i+1}) is an edge of G. Since I^- is finite, the constructed sequence must contain a loop. Furthermore, G has at least one terminating loop. Let L be a terminating loop of (O, Γ_{I^-}), $h \in L$ and

$$r : h \leftarrow Pos$$

be an arbitrary rule in Γ_{I^-}. Obviously $L \subseteq I^-$. Because L is a terminating loop of (O, Γ_{I^-}), it follows that the following cases hold:

- $I^- \cap Pos \subseteq L$, and
- for every monotonic dl-atom $DL[\lambda; Q](\vec{t})$ in Pos, if $S \uplus p$ (or $S \cup\!\!\!\!- \, p$) appear in λ for some S then we have $p(\vec{c}) \in I^-$ implies $p(\vec{c}) \in L$.

Clearly L is also a strong loop of $\mathcal{K}$. Due to $I' \models_O sLF(L, \mathcal{K})$, $L \subseteq I^-$, and $I^- \subseteq I'$, we have $I' \models_O \bigvee L$. Thus, P has at least one rule

$$r' : h' \leftarrow Pos', not\ Neg'$$

such that $h' \in L$, $Pos' \cap L = \emptyset$ and

$$I' \models_O \left(\bigwedge_{A \in Pos'} \gamma(A, L) \wedge \bigwedge_{B \in Neg'} \neg B \right).$$

It follows that $I \models_O A$ for each nonmonotonic dl-atom $A \in Pos'$ and $I \not\models_O B$ for each $B \in Neg'$. Let $(r'' : h' \leftarrow Pos'')$ be the rule in sP_O^I that is obtained from r' by the strong dl-transformation. Clearly, $r'' \in \Gamma$ by Lemma 3.4. Furthermore, due to $h' \in L \subseteq I^-$, we have

$$r'' \in \Gamma_{I^-}.$$

By $Pos'' \cap I^- \subseteq L$ we have $Pos' \cap I^- \subseteq L$. It follows that $Pos' \cap I^- = \emptyset$ by $Pos' \cap L = \emptyset$. So we have $Pos' \cap HB_P \subseteq I^*$ since $I \models A$ for every $A \in Pos' \cap HB_P$. It follows that $Pos'' \cap HB_P \subseteq I^*$. Since $r'' \in \Gamma_{I^-}$, Pos'' must have a monotonic dl-atom $A = DL[\lambda; Q](\vec{t})$ such that $I^* \not\models_O A$, i.e., $I \setminus I^- \not\models_O A$. By Lemma 3.4, we have $I \setminus L \models_O A$ since $I' \models_O \gamma(A, L)$. Thus there must exist an atom $p(\vec{c}) \in (I \setminus L) \setminus (I \setminus I^-)(= I^- \setminus L)$ and $S \uplus p$ (or $S \cup\!\!\!\!- \, p$) appears in λ since A is monotonic. Recall that $p(\vec{c}) \in I^-$ implies $p(\vec{c}) \in L$. It follows that $I \setminus L \not\models_O A$ by $I \setminus I^- \not\models_O A$. It is a contradiction.

Consequently, $I \setminus I^- = \emptyset$. It follows $I \subseteq I^*$. We have that $lfp(\gamma_{\mathcal{K}_\Gamma^{s,I}}) \subseteq lfp(\gamma_{\mathcal{K}^{s,I}})$ by $\Gamma \subseteq sP_O^I$. It follows that $I \subseteq lfp(\gamma_{\mathcal{K}^{s,I}})$. By Lemma 3.5, $lfp(\gamma_{\mathcal{K}^{s,I}}) \subseteq I$ since $I \models_O COMP(\mathcal{K})$. Consequently, $I = lfp(\gamma_{\mathcal{K}^{s,I}})$. Thus I is a strong answer set of $\mathcal{K}$. $\Box$

Since a weak loop of a dl-program $\mathcal{K}$ is also a strong loop of $\mathcal{K}$, as a by-product, our loop formula characterizations yield an alternative proof that strong answer sets are also weak answer sets.

Proposition 3.7. *Let $\mathcal{K} = (O, P)$ be a dl-program, I an interpretation of P and L a weak loop of $\mathcal{K}$. Then we have $LE(I, \mathcal{K}) \models_O sLF(L, \mathcal{K}) \supset wLF(L, \mathcal{K})$.*

Proof. Let $I' = LE(I, \mathcal{K})$. Suppose $I' \models_O sLF(L, \mathcal{K})$ and $I' \not\models_O wLF(L, \mathcal{K})$. We have that $I' \cap L \neq \emptyset$ and, for each dl-rule $(h \leftarrow Pos, not\ Neg)$ in P such that $h \in L$ and $Pos \cap L = \emptyset$,

$$I' \not\models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B.$$

As L is also a strong loop of $\mathcal{K}$ and $I' \models_O sLF(L, \mathcal{K})$, there exists at least one rule $(h' \leftarrow Pos', not\ Neg')$ in P such that $h' \in L$, $Pos' \cap L = \emptyset$ and

$$I' \models_O \bigwedge_{A' \in Pos'} \gamma(A', L) \wedge \bigwedge_{B' \in Neg'} \neg B'.$$

For each formula ψ of $\mathcal{L}_\mathcal{K}$, $I' \models_O \psi$ implies that $I \models_O \psi$ since ψ mentions only the predicates occurring in $\mathcal{K}$. Notice further that if A' is a monotonic dl-atom, then $I \setminus L \models_O A$ by Lemma 3.4, thus by monotonicity $I \models_O A$. It follows that

$$I \models_O \bigwedge_{A' \in Pos'} A' \wedge \bigwedge_{B' \in Neg'} \neg B'$$

which contradicts $I \not\models_O wLF(L, \mathcal{K})$. $\square$

3.4. An alternative method of computing weak and strong answer sets

Theorems 3.3 and 3.6 serve as the basis for an alternative method of computing weak and strong answer sets using a SAT solver, along with a dl-reasoner $\mathcal{R}$ with the following property: $\mathcal{R}$ is sound, complete, and terminating for entailment checking. Let $\mathcal{K} = (O, P)$ be a dl-program and $T = COMP(\mathcal{K})$. We replace all dl-atoms in T with new propositional atoms to produce T'. Let ξ_A be the new atom in T', for the dl-atom A in T, and X be the set of all such new atoms in T'. Below, we outline an algorithm to compute the weak answer sets of $\mathcal{K}$ (here we only describe how to compute first such an answer set). To compute a strong answer set, replace the word weak with strong.

(i) Generate a model I of T; if there is none, then there is no weak answer set.
(ii) Check if I is a weak answer set of $\mathcal{K}$,
  (a) if yes, return I as a weak answer set of $\mathcal{K}$.
  (b) if no, add a weak loop formula into T that is not satisfied by I relative to O, and goto (i).

To generate a model of T, we compute a model M of T' using a SAT solver, and then use $\mathcal{R}$ to check the entailment: for each dl-atom A in T, if $M \models \xi_A$ then $M \models_O A$ otherwise $M \not\models_O A$. Let $M' = M/X$. Recall that X is the set of new atoms in T'. It is not difficult to verify that M' is a model of $\mathcal{K}$.

Proposition 3.8. *Let ψ be a formula of $\mathcal{L}_\mathcal{K}$, M a set of atoms of $\mathcal{L}_\mathcal{K}$ and*

$$M' = M \cup \{\xi_A | A \text{ is a dl-atom occurring in } \psi \text{ and } M \models_O A\}.$$

Then $M \models_O \psi$ iff $M' \models \psi'$ where ψ' is the formula obtained from ψ by replacing every occurrence of a dl-atom A with a fresh atom ξ_A, i.e., different dl-atoms are replaced with different propositional variables.

Proof. We prove this by induction on the structure of ψ.

- ψ is an atom $p(\vec{t})$. It is evident that $M \models \psi$ iff $M' \models \psi'$ by $\psi' = \psi$, M' and M contain the same atoms of $\mathcal{L}_\mathcal{K}$.
- ψ is a dl-atom A. Since $\psi' = \xi_A$, $M' \models \psi'$ iff $\xi_A \in M'$ iff $M \models_O A$.
- $\psi = \neg\varphi$. We have that $M \models_O \neg\varphi$ iff $M \not\models_O \varphi$ iff $M' \not\models \varphi'$ (by the inductive assumption) iff $M' \models \neg\varphi'$.
- $\psi = \psi_1 \vee \psi_2$. We have that $M \models_O \psi_1 \vee \psi_2$ iff $M \models_O \psi_1$ or $M \models_O \psi_2$ iff $M' \models \psi_1'$ or $M \models_O \psi_2'$ (by the inductive assumption) iff $M' \models \psi_1' \vee \psi_2'$ iff $M' \models (\psi_1 \vee \psi_2)'$.

The proof for the other connectives is similar. $\square$

The strong and weak answer set semantics of dl-programs have been implemented in the prototype system SWLP,[4] which is done by a guess-and-check approach. Informally, given a dl-program $\mathcal{K} = (O, P)$, the algorithm below can be used to compute weak answer sets of $\mathcal{K}$:

(1) replace each dl-atom α in P by a fresh atom a_α;
(2) add to the result of Step (1) all the rules of the form, for each dl-atom α,

$$a_\alpha \leftarrow not\ \neg a_\alpha, \quad \text{and} \quad \neg a_\alpha \leftarrow not\ a_\alpha.$$

The resulting program is denoted by P_{guess}.

[4] https://www.mat.unical.it/ianni/swlp/.

(3) For each answer set I of P_{guess} and each dl-atom α in DL_P, check whether $a_\alpha \in I$ iff $I \models_O \alpha$. If the condition holds, then $I \cap HB_P$ is a weak answer set of $\mathcal{K}$.

In Step (2), classical negation is introduced, which can be handled easily in answer set solvers like DLV [16]. In Step (3), any answer set solver can be applied to enumerate the answer sets of P_{guess} and a dl-reasoner can be used to check the condition $I \models_O \alpha$. The reader can refer to [8] for the details of the implementation and interesting dl-programs, using the ASP solver DLV and a dl-reasoner.

In terms of our approach, we use a SAT solver to compute a model I of the completion of P_{guess} such that $I \models a_\alpha$ iff $I \models_O \alpha$, and then check whether I is a weak answer set of $\mathcal{K}$. Thus a main difference in the method outlined here is that we use a SAT solver to generate candidate models, which allows taking the advantages of the state-of-the-art SAT technology.

For strong answer sets, the construction of a strong loop formula requires checking monotonicity of dl-atoms. However, for the class of dl-programs mentioning no $\cap\!\!\!\!-$, this checking is not needed and the construction of a strong loop formula is hence tractable.

4. Canonical answer sets

4.1. Motivation: the problem of self-support

As commented by Eiter et al. [8], some weak answer sets may be considered counterintuitive because of "self-supporting" loops. For instance, consider the weak answer set $\{p(a)\}$ of the dl-program $\mathcal{K}_1$ in Example 1. The evidence of the truth of $p(a)$ is inferred by means of a self-supporting loop: "$p(a) \Leftarrow DL[c \uplus p; c'](a) \Leftarrow p(a)$", which involves not only the dl-atom $DL[c \uplus p; c'](a)$ but the DL knowledge base O. Thus the truth of $p(a)$ depends on the truth of itself. This self-support is excluded by the strong loop formula of the loop $L = \{p(a)\}$.

Let us consider the dl-program $\mathcal{K}_2$ in Example 1 again. Note that $\{p(a)\}$ is a strong answer set of $\mathcal{K}_2$. The truth of the atom $p(a)$ depends on the truth of $[c \sqcap \neg b](a)$ which depends on the truth of $p(a)$ and $\neg q(a)$. Thus the truth of $p(a)$ depends on the truth of itself. The self-supporting loop is: "$p(a) \Leftarrow DL[c \uplus p, b \cap\!\!\!\!- q; c \sqcap \neg b](a) \Leftarrow (p(a) \wedge \neg q(a))$". In this sense, some strong answer sets may be considered counterintuitive as well.

The notion of "circular justification" was formally defined by [18] to characterize self-supports for LPARSE [29] programs, which was motivated by the notion of *unfoundedness* for logic programs [30] and logic programs with aggregates [7]. With slight modifications, we extend the concept of circular justification to dl-programs. Formally, let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$ be a supported model of $\mathcal{K}$. I is said to be *circularly justified* (or simply *circular*) if there is a nonempty subset M of I such that

$$I \setminus M \not\models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B \tag{7}$$

for every dl-rule $(h \leftarrow Pos, not\ Neg)$ in P with $h \in M$ and $I \models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B$. Otherwise, we say that I is *noncircular*. Intuitively speaking, Condition (7) means that the atoms in M have no support from outside of M, i.e., they have to depend on themselves.

Example 3. Let $\mathcal{K} = (\emptyset, P)$ where P consists of

$$p(a) \leftarrow not\ DL[b \cap\!\!\!\!- p; \neg b](a).$$

It is not difficult to verify that $\mathcal{K}$ has two weak answer sets $\emptyset$ and $\{p(a)\}$. They are strong answer sets of $\mathcal{K}$ as well. In terms of the above definition, $\{p(a)\}$ is circular.

It is interesting to note that weak answer sets allow self-supporting loops involving any dl-atoms (either monotonic or nonmonotonic), while strong answer sets allow self-supporting loops only involving nonmonotonic dl-atoms and their default negations. These considerations motivate us to define a new semantics which is free of circular justifications.

4.2. Canonical answer sets by loop formulas

Let $\mathcal{K} = (O, P)$ be a dl-program. The *canonical dependency graph* of $\mathcal{K}$, written as $G^c_{\mathcal{K}}$, is the directed graph (V, E), where $V = HB_P$ and $(u, v) \in E$ if there is a rule of the form (2) in P such that $A = u$ and there exists an interpretation $I \subseteq HB_P$ such that either of the following two conditions holds:

(1) $I \not\models_O B_i$ and $I \cup \{v\} \models_O B_i$, for some i $(1 \le i \le m)$. In this case, we say that v is a *positive monotonic* (resp., *nonmonotonic*) dependency of B_i if B_i is a monotonic (resp., nonmonotonic) dl-atom. Intuitively, the truth of B_i may depend on that of v while the truth of u may depend on that of B_i. Thus the truth of u may depend on that of v.
(2) $I \models_O B_j$ and $I \cup \{v\} \not\models_O B_j$, for some j $(1 + m \le j \le n)$. Clearly, B_j must be nonmonotonic. In this case, we say that v is a *negative nonmonotonic dependency* of B_j. Intuitively, the truth of u may depend on that of "*not* B_j", while its truth may depend on that of v. Thus the truth of u may depend on that of v.

Algorithm 1 *Psup*(A, I_1, I_2)

```
1: I* ← I_2
2: M ← I_2 \ I_1
3: for all h ∈ M ∩ I* do
4:     h* ← h
5:     if I* \ {h*} ⊨_O A then
6:         I* ← I* \ {h*}
7:         continue
8:     end if
9:     break
10: end for
11: return (I*, h*)
```

A nonempty subset L of HB_P is a *canonical loop* of $\mathcal{K}$ if there is a cycle in $G^c_{\mathcal{K}}$ that goes through only and all the nodes in L. It is clear that if $B_i = v$ then the interpretation $I = \{v\}$ satisfies v while $I \setminus \{v\}$ does not. Thus the notion of canonical loops is a generalization of that of weak loops given in Section 3.2, and a generalization of the notion of loops for normal logic programs [17].

Note further that the canonical dependency graph is not a generalization of the strong positive dependency graph, since some strong loops are not canonical loops. For example, with the dl-program $\mathcal{K} = (\emptyset, P)$, where $P = \{p(a) \leftarrow DL[c \uplus p, c \cap\!\!\!\!- p, \neg c](a)\}$, the dl-atom $A = DL[c \uplus p, c \cap\!\!\!\!- p, \neg c](a)$ is equivalent to $\top$. So it is monotonic. It follows that $L = \{p(a)\}$ is a strong loop of $\mathcal{K}$. However, L is not a canonical loop of $\mathcal{K}$, because there is no interpretation I such that $I \not\models_O A$ and $I \cup \{p(a)\} \models_O A$.

Due to the two kinds of dependences in a canonical dependency graph defined above, to define canonical loop formulas, we need two kinds of irrelevant formulas: let L be a set of atoms and $A = DL[\lambda; Q](\vec{t})$ a nonmonotonic dl-atom. The *positive canonical irrelevant formula*[5] of A with respect to L, written as $pCF(A, L)$, is $DL[\lambda_L; Q](\vec{t})$ where λ_L is obtained from λ by replacing each predicate p with p_L if L contains an atom $p(\vec{c})$ which is a positive nonmonotonic dependency of A. The *negative canonical irrelevant formula* of A with respect to L, written as $nCF(A, L)$, is $DL[\lambda_L; Q](\vec{t})$ where λ_L is obtained from λ by replacing each predicate p with p_L if L contains an atom $p(\vec{c})$ which is a negative nonmonotonic dependency of A.

Lemma 4.1. *Let $\mathcal{K} = (O, P)$ be a dl-program, $A \in DL_P$ and $I_1 \subset I_2 \subseteq HB_P$.*

(1) *If $I_1 \not\models_O A$ and $I_2 \models_O A$ then there exists an interpretation I^* and an atom $h^* \in I_2 \setminus I_1$ such that $I_1 \subset I^* \subseteq I_2$, $I^* \models_O A$ and $I^* \setminus \{h^*\} \not\models_O A$.*
(2) *If A is nonmonotonic, $I_1 \models_O A$ and $I_2 \not\models_O A$ then there exists an interpretation I^* and an atom $h^* \in I_2 \setminus I_1$ such that $I_1 \subseteq I^* \subset I_2$, $I^* \cup \{h^*\} \not\models_O A$ and $I^* \models_O A$.*

Proof. (1) Clearly $I_2 \setminus I_1 \neq \emptyset$ by the assumption. We construct the interpretation I^* by Algorithm 1: *Psup*(A, I_1, I_2). Since both I_2 and M are finite, the algorithm definitely terminates. Since M is a nonempty subset of I_2, the **forall** loop will run at least once. Suppose *Psup*(A, I_1, I_2) is executed and terminated. There are only two possible cases leading to its termination:

- There is no $h \in M \cap I^*$ (line 3). It follows that $I^* = I_1$ and $I^* \models_O A$. The latter contradicts $I_1 \not\models_O A$. Thus this case is impossible.
- The "break" is executed (line 9). It follows that $I^* \subseteq I_2$ and $I^* \setminus \{h^*\} \not\models_O A$.

Thus the above algorithm returns (I^*, h^*) such that $I^* \models_O A$ and $I^* \setminus \{h^*\} \not\models_O A$.

(2) We have Algorithm 2 for this purpose. Similarly, since both M and I_2 are finite then the algorithm *Nsup*(A, I_1, I_2) definitely terminates and the **forall** loop will be executed at least once. Suppose *Nsup*(A, I_1, I_2) is executed and terminated. If *Nsup*(A, I_1, I_2) terminates because of $M \setminus I^* = \emptyset$ in the **forall** loop then we have $I^* = I_2$ and $I^* \models_O A$. The latter contradicts $I_2 \not\models_O A$. Thus the only case leading to the termination of *Nsup*(A, I_1, I_2) is the "break" (line 10). In that case, we have that $I^* \cup \{h^*\} \not\models_O A$ and $I^* \models_O A$. It is obvious $I_1 \subseteq I^*$. $\square$

Lemma 4.2. *Let $\mathcal{K} = (O, P)$ be a dl-program, $I \subseteq HB_P$, L a set of atoms and $A = DL[\lambda; Q](\vec{t})$ a nonmonotonic dl-atom in $DL^?_P$.*

(1) *If $LE(I, \mathcal{K}) \models_O pCF(A, L)$ then $I \setminus L \models_O A$,*
(2) *If $LE(I, \mathcal{K}) \not\models_O nCF(A, L)$ then $I \setminus L \not\models_O A$.*

Proof. Let $I' = LE(I, \mathcal{K})$, and for clarity and without loss of generality, suppose $\lambda = (S_1 \uplus p_1, S_2 \cap\!\!\!\!- p_2)$.

(1) Suppose $p_1 \neq p_2$. There is no atom $p_2(\vec{c})$ which is a positive nonmonotonic dependency of A. If there is no atom $p_1(\vec{c}) \in L$ such that $p_1(\vec{c})$ is a positive nonmonotonic dependency of A then $pCF(A, L) = A$. It follows that $I \models_O A$ since $I' \models_O A$ and $I' \cap HB_P = I$. Suppose $I \setminus L \not\models_O A$. From (1) of Lemma 4.1, there is an atom $h \in I \setminus (I \setminus L)$, i.e., $h \in L$, and an

[5] This is different from the original definition in [32], but it is equivalent to the original one under loop extended interpretations.

Algorithm 2 *Nsup*(A, I_1, I_2)

$I^* \leftarrow I_1$
$M \leftarrow I_2 \setminus I_1$
for all $h \in M \setminus I^*$ **do**
  if $I^* \cup \{h\} \models_O A$ **then**
    $I^* \leftarrow I^* \cup \{h\}$
    continue
  end if
  $h^* \leftarrow h$
  $I^* \leftarrow I^* \cup \{h^*\}$
  break
end for
return (I^*, h^*)

interpretation I^* such that $I^* \not\models_O A$ and $I^* \cup \{h\} \models_O A$. It is evident that h must mention the predicate p_1. It follows that h is a positive nonmonotonic dependency of A which contradicts the assumption. Thus $I \setminus L \models_O A$.

Suppose there is an atom $p_1(\vec{c}) \in L$ such that $p_1(\vec{c})$ is a positive nonmonotonic dependency of A. By $p_{1_L}(\vec{c}) \in I'$ iff $p_1(\vec{c}) \in I \setminus L$, we have that
$I' \models_O pCF(A, L)$
$\Rightarrow O \cup \{S_1(\vec{e}) | p_{1_L}(\vec{c}) \in I'\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I'\} \models Q(\vec{t})$
$\Rightarrow O \cup \{S_1(\vec{e}) | p_1(\vec{c}) \in I \setminus L\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I\} \models Q(\vec{t})$
$\Rightarrow O \cup \{S_1(\vec{e}) | p_1(\vec{c}) \in I \setminus L\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I \setminus L\} \models Q(\vec{t})$
$\Rightarrow I \setminus L \models_O DL[S_1 \uplus p_1, S_2 \cap\!\!\!- p_2; Q](\vec{t})$
$\Rightarrow I \setminus L \models_O A$.

It is similar to show that $I \setminus L \models_O A$ for the case $p_1 = p_2$.

(2) Suppose $p_1 \neq p_2$. There is no atom $p_1(\vec{c})$ which is a negative nonmonotonic dependency of A. If there is no atom $p_2(\vec{c}) \in L$ such that $p_2(\vec{c})$ is a negative nonmonotonic dependency of A then $nCF(A, L) = A$. It follows that $I \not\models_O A$ since $I' \not\models_O A$ and $I' = LE(I, \mathcal{K})$. Suppose $I \setminus L \models_O A$. By (2) of Lemma 4.1, there is an atom $h \in I \setminus (I \setminus L)$, i.e., $h \in L$, and an interpretation I^* such that $I^* \models_O A$ and $I^* \cup \{h\} \not\models_O A$. As h must mention the predicate p_2, it follows that h is a negative nonmonotonic dependency of A which contradicts the assumption. Thus $I \setminus L \not\models_O A$.

Suppose there is an atom $p_2(\vec{c}) \in L$ such that $p_2(\vec{c})$ is a negative nonmonotonic dependency of A. By $p_{2_L}(\vec{c}) \in I'$ iff $p_2(\vec{c}) \in I \setminus L$, we have that
$I' \not\models nCF(A, L)$
$\Rightarrow O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in I'\} \cup \{\neg S_2(\vec{e}) | p_{2_L}(\vec{e}) \notin I'\} \not\models Q(\vec{t})$
$\Rightarrow O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in I\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I \setminus L\} \not\models Q(\vec{t})$
$\Rightarrow O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in I \setminus L\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I \setminus L\} \not\models Q(\vec{t})$
$\Rightarrow I \setminus L \not\models_O DL[S_1 \uplus p_1, S_2 \cap\!\!\!- p_2; Q](\vec{t})$
$\Rightarrow I \setminus L \not\models_O A$.

It is similar to show that $I \setminus L \not\models_O A$ for the case $p_1 = p_2$. $\square$

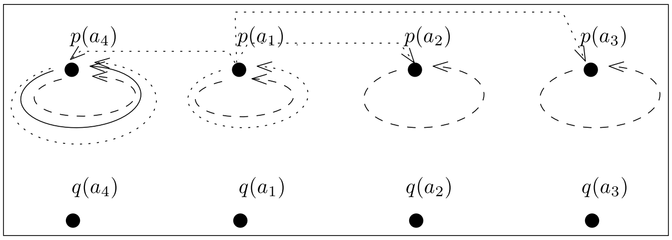
Please note that the converses of (1) and (2) in the above lemma do not generally hold. For example, let $\mathcal{K} = (O, P)$ be a dl-program where $O = \emptyset$, $A = DL[S_1 \uplus p_1, S_2 \cap\!\!\!- p_2; S_1 \sqcap \neg S_2](a)$ occurring in P, $I_1 = \{p_1(a), p_2(a)\}$, $I_2 = \{p_1(a)\}$, $L_1 = \{p_2(a)\}$ and $L_2 = \{p_1(a)\}$. Because there is no interpretation I such that $I \not\models_O A$ and $I \cup L_1 \models_O A$, it follows $pCF(A, L_1) = A$. Similarly, we have $nCF(A, L_2) = A$. Note that $I_1 \setminus L_1 \models_O A$. However $LE(I_1, \mathcal{K}) \not\models_O pCF(A, L_1)$ since $I_1 \not\models_O A$. Similarly, we have that $I_2 \setminus L_2 \not\models_O A$ and $LE(I_2, \mathcal{K}) \models_O nCF(A, L_2)$ since $I_2 \not\models_O A$. Even if we add the condition $I \models_O A$, the converse of (1) does not hold either. For instance, let $O = \emptyset$, $A = DL[S_1 \uplus p_1, S_2 \cap\!\!\!- p_2; S_1 \cup \neg S_2](a)$ and $I = L = \{p_1(a), p_2(a)\}$. We can verify that $I \models_O A$ and $I \setminus L \models_O A$, however $I' \not\models_O pCF(A, L)$.

We are now in the position to define canonical loop formulas. Let $\mathcal{K} = (O, P)$ be a dl-program, $M \subseteq HB_P$ and L a loop of $\mathcal{K}$. The *canonical loop formula* of L relative to $\mathcal{K}$ under M, written as $cLF(L, M, \mathcal{K})$, is the following formula

$$\bigvee L \supset \bigvee_{1 \le i \le n} \left(\bigwedge_{A \in Pos_i} \delta_1(A, L) \wedge \bigwedge_{B \in Neg_i} \neg \delta_2(B, L) \right)$$

where

- $(h_1 \leftarrow Pos_1, not\ Neg_1), \ldots, (h_n \leftarrow Pos_n, not\ Neg_n)$ are all the rules in P such that $h_i \in L$, $Pos_i \cap L = \emptyset$, $M \models_O A$ for every $A \in Pos_i$ and $M \not\models_O B$ for every $B \in Neg_i$ for all i $(1 \le i \le n)$,
- $\delta_1(A, L) = pCF(A, L)$ if $A \in DL_P^?$, and $\gamma(A, L)$ otherwise,
- $\delta_2(B, L) = nCF(B, L)$ if $B \in DL_P^?$, and B otherwise.



——: weak dependency; · · · : strong dependency; – – –: canonical dependency;

Fig. 1. The positive dependency relations on HB_P.

Given a dl-program $\mathcal{K} = (O, P)$ and an interpretation $I \subseteq HB_P$. We call I a *canonical answer set* of $\mathcal{K}$ if $LE(I, \mathcal{K}) \models_O COMP(\mathcal{K}) \cup cLF(I, \mathcal{K})$, where $cLF(I, \mathcal{K}) = \{cLF(L, I, \mathcal{K}) | L$ is a canonical loop of $\mathcal{K}\}$. It is not difficult to prove that every canonical answer set of a dl-program $\mathcal{K}$ is a supported model of $\mathcal{K}$.

Example 4. Consider the dl-program $\mathcal{K}_2$ in Example 1, i.e., $\mathcal{K}_2 = (O, P_2)$ where $O = \emptyset$ and $P_2 = \{p(a) \leftarrow DL[c \uplus p, b \Cap q; c \sqcap \neg b](a)\}$. The dl-atom $DL[c \uplus p, b \Cap q; c \sqcap \neg b](a)$ is nonmonotonic. We have that $\emptyset \not\models_O DL[c \uplus p, b \Cap q; c \sqcap \neg b](a)$ and $\{p(a)\} \models_O DL[c \uplus p, b \Cap q; c \sqcap \neg b](a)$. Thus $L = \{p(a)\}$ is a canonical loop of $\mathcal{K}_2$. Let $I = \{p(a)\}$. The canonical loop formula $cLF(L, I, \mathcal{K})$ is the following formula

$$p(a) \supset DL[c \uplus p_L, b \Cap q; c \sqcap \neg b](a).$$

It is not difficult to see that $LE(I, \mathcal{K}) = I \cup \{p_{L'}(a)\}$ where $L' = \{q(a)\}$. Evidently, $LE(I, \mathcal{K}) \not\models_O cLF(L, I, \mathcal{K})$ due to $p_L(a) \notin LE(I, \mathcal{K})$. So that I is not a canonical answer set of $\mathcal{K}_2$, even if $I \models_O COMP(\mathcal{K}_2)$.

The next example demonstrates the difference among the positive dependency graphs of dl-programs.

Example 5. Let $\mathcal{K} = (O, P)$ be a dl-program where $O = \emptyset$ and P consists of the following rules:

$$p(a_1) \leftarrow DL[c \uplus p, c](a_1), \qquad p(a_3) \leftarrow not\ DL[c \Cap p, \neg c](a_3),$$
$$p(a_2) \leftarrow DL[c \uplus p, b \Cap q; c \sqcap \neg b](a_2), \qquad p(a_4) \leftarrow p(a_4).$$

The only weak positive dependency on HB_P is $(p(a_4), p(a_4))$, the strong positive dependency includes $(p(a_1), p(a_1))$ besides the weak one, while the canonical positive dependency contains $(p(a_2), p(a_2))$ and $(p(a_3), p(a_3))$ in addition to the strong ones. Fig. 1 depicts the various dependency relations on HB_P. The weak positive dependency graph is $G^w_{\mathcal{K}} = (V, E)$ where $V = \{p(a_i), q(a_i) | 1 \le i \le 4\}$ and $E = \{(p(a_4), p(a_4))\}$, while the strong one is $G^s_{\mathcal{K}} = (V, E')$ where $E' = E \cup \{(p(a_1), p(a_i)) | 1 \le i \le 4\}$. The canonical dependency graph is $G^c_{\mathcal{K}} = (V, E'')$ where $E'' = \{(p(a_i), p(a_i)) | 1 \le i \le 4\}$.

Comparing with the previous definitions of loop formulas, in addition to the irrelevant formulas of nonmonotonic dl-atoms, the definition of canonical loop formulas has a notable distinction: it is given under a set M of atoms whose purpose is to restrict that the support of any atom in L come from the rules whose bodies are satisfied by M (relative to a knowledge base). The next proposition shows that the canonical loops and canonical loop formulas for dl-programs are indeed a generalization of loops and loop formulas for normal logic programs [17] respectively.

Proposition 4.3. *Let $\mathcal{K} = (O, P)$ be a dl-program where P is a normal logic program and $O = \emptyset$, $L \subseteq HB_P$ and M a model of the completion of P.*

(1) *L is a loop of P if and only if L is a canonical loop of $\mathcal{K}$.*
(2) *$M \models LF(L, P)$ if and only if $M \models_O cLF(L, M, P)$, where $LF(L, P)$ is the loop formula associated with L under P [17].*

Proof. (1) Clearly, since for each atom h there always exists an interpretation $I = \{h\}$ such that $I \models_O h$ and $I \setminus \{h\} \not\models_O h$.

(2) Note that $M \models_O cLF(L, I, \mathcal{K})$ iff there is a rule $(r : h \leftarrow Pos, not\ Neg)$ in P such that $h \in L$, $Pos \cap L = \emptyset$, $M \models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B$ and

$$M \models_O \bigwedge_{A \in Pos} \delta_1(A, L) \wedge \bigwedge_{B \in Neg} \neg\delta_2(B, L). \tag{8}$$

Since r mentions no dl-atoms, it implies that $\delta_1(A, L) = A$ and $\delta_2(B, L) = B$. Thus Eq. (8) holds iff $M \models \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B$. Consequently, $M \models_O cLF(L, I, \mathcal{K})$ iff $M \models LF(L, P)$. □

Proposition 4.4. *Let $\mathcal{K} = (O, P)$ be a dl-program and I a canonical answer set of $\mathcal{K}$. Then I is minimal in the sense that $\mathcal{K}$ has no canonical answer set I' such that $I' \subset I$.*

Proof. Suppose there is a canonical answer set I_1 of $\mathcal{K}$ such that $I_1 \subset I$. Let $M = I \setminus I_1$. Since $I \models_O COMP(\mathcal{K})$ and $I_1 \models_O COMP(\mathcal{K})$, for each atom $h \in M$, there is no rule $(h \leftarrow Pos, not\ Neg)$ in P such that

$$I_1 \models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B. \tag{9}$$

Similarly, there is at least one rule $(h \leftarrow Pos', not\ Neg')$ in P such that

$$I \models_O \bigwedge_{A \in Pos'} A \wedge \bigwedge_{B \in Neg'} \neg B.$$

It follows that at least one of the following conditions holds:

- There is an atom $h' \in Pos'$ such that $h' \in M$.
- There is a dl-atom $A \in Pos'$ such that $I_1 \not\models_O A$. By $I \models_O A$, there is an atom $h' \in I \setminus I_1$, i.e., $h' \in M$, and an interpretation I^* such that $I^* \not\models_O A$ and $I^* \cup \{h'\} \models_O A$ by (1) of Lemma 4.1.
- There is a nonmonotonic dl-atom $B \in Neg'$ such that $I_1 \models_O B$. By $I \not\models_O B$, there is an atom $h' \in I \setminus I_1$, i.e., $h' \in M$, and an interpretation I^* such that $I^* \models_O B$ and $I^* \cup \{h'\} \not\models_O B$ by (2) of Lemma 4.1.

It follows that (h, h') is an edge of $G^c_{\mathcal{K}}$. Due to that h is an arbitrary atom in M and M is finite, there must exist a canonical loop L of $\mathcal{K}$ such that $L \subseteq M$. We can further assume L is terminating, i.e., (a) L is a maximal subset of M and (b) L is a canonical loop of $\mathcal{K}$ and (c) $G^c_{\mathcal{K}}$ has no path from one atom of L to an atom of another maximal canonical loop L' of $\mathcal{K}$ with $L' \subseteq M$. Since $LE(I, \mathcal{K}) \models_O cLF(L, I, \mathcal{K})$, there is at least one rule $(h \leftarrow Pos'', not\ Neg'')$ in P such that $h \in L$, $Pos'' \cap L = \emptyset$,

$$I \models_O \bigwedge_{A \in Pos''} A \wedge \bigwedge_{B \in Neg''} \neg B \text{ and } LE(I, \mathcal{K}) \models_O \bigwedge_{A \in Pos''} \delta_1(A, L) \wedge \bigwedge_{B \in Neg''} \neg\delta_2(B, L).$$

By Lemmas 3.4 and 4.2, we have that

$$I \setminus L \models_O \bigwedge_{A \in Pos''} A \wedge \bigwedge_{B \in Neg''} \neg B. \tag{10}$$

If $L \subset M$ then $I_1 \subset I \setminus L$. In terms of the previous analysis, there is an atom $h'' \in (I \setminus L) \setminus I_1$, i.e., $h'' \in M \setminus L$, such that (h, h'') is an edge of $G^c_{\mathcal{K}}$. Thus $G^c_{\mathcal{K}}$ must have a path from h to another canonical loop L'' of $\mathcal{K}$, where $L'' \subseteq M$, which contradicts with L is a terminating canonical loop. So we have $L = M$. According to Eq. (10), we have $I_1 \models_O \bigwedge_{A \in Pos''} A \wedge \bigwedge_{B \in Neg''} \neg B$ which contradicts the condition (9). Consequently, I_1 cannot be a canonical answer set of $\mathcal{K}$. This completes the proof. $\square$

The following two propositions show that the canonical answer sets of dl-programs are noncircular strong answer sets. Thus canonical answer sets are weak answer sets as well.

Proposition 4.5. *Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$ a canonical answer set of $\mathcal{K}$. Then I is noncircular.*

Proof. Suppose I is circular, i.e., there exists a nonempty subset M of I such that, for each $(h \leftarrow Pos, not\ Neg)$ in P with $h \in M$ and $I \models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B$, the following condition holds:

$$I \setminus M \not\models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B. \tag{11}$$

Without loss of generality, we assume M is such a minimal one. It follows that at least one of the following cases hold:

- $Pos \cap M \neq \emptyset$ which implies that there is an atom $h' \in Pos \cap M$.
- There is a dl-atom $A \in Pos$ such that $I \setminus M \not\models_O A$. Knowing that $I \models_O A$, it follows that there is an interpretation $I^* \subseteq I$ and an atom $h' \in I \setminus (I \setminus M)$, i.e., $h' \in M$ such that $I^* \models_O A$ and $I^* \setminus \{h'\} \not\models_O A$ by (1) of Lemma 4.1. So that h' is a positive nonmonotonic dependency of A.
- There is a nonmonotonic dl-atom $B \in Neg$ such that $I \setminus M \models_O B$. Knowing that $I \not\models_O B$, it follows that there is an interpretation I^* and an atom $h' \in I \setminus (I \setminus M)$, i.e., $h' \in M$ such that $I^* \models_O B$ and $I^* \cup \{h'\} \not\models_O B$ by (2) of Lemma 4.1. So that h' is a negative nonmonotonic dependency of A.

Thus we have that (h, h') is an edge of the canonical dependency graph of $\mathcal{K}$. Because $h \in M$ is arbitrary and M is finite, there is a terminating canonical loop in the generated subgraph of $G^c_{\mathcal{K}}$ on M, i.e., the graph $G' = (V, E)$ where $V = M$ and $(u, v) \in E$ if (u, v) is an edge of $G^c_{\mathcal{K}}$. Let $L \subseteq M$ be such a terminating canonical loop.

By $LE(I, \mathcal{K}) \models_O cLF(L, I, \mathcal{K})$ and $L \subseteq I$, we have that there is at least one rule $(h \leftarrow Pos', not\ Neg')$ in P such that $h \in L$, $L \cap Pos' = \emptyset$,

$$I \models_O \bigwedge_{A \in Pos'} A \wedge \bigwedge_{B \in Neg'} \neg B \text{ and } LE(I, \mathcal{K}) \models_O \bigwedge_{A \in Pos'} \delta_1(A, L) \wedge \bigwedge_{B \in Neg'} \neg\delta_2(B, L).$$

It follows that, by Lemmas 3.4 and 4.2,

$$I \setminus L \models_O \bigwedge_{A \in Pos'} A \wedge \bigwedge_{B \in Neg'} \neg B.$$

Thus $M \neq L$ and hence by Eq. (11) $I \setminus M \subset I \setminus L$. However, by a similar analysis, we have that G' has a path from one atom in L to another loop of G'. It contradicts the fact that L is a terminating canonical loop of G'. Thus I must be noncircular. □

Unfortunately, the notion of noncircular justification is not a complete characterization of self-supportedness as illustrated by the next example.

Example 6. Let $\mathcal{K} = (O, P)$ where $O = \{\neg S_1(b), S_2(a)\}$ and P consists of

$$p(b) \leftarrow p(a),$$
$$p(a) \leftarrow DL[S_1 \uplus p, S_2 \cap\!\!\!\!- p; S_2](b).$$

All the interpretations $\emptyset$, $\{p(b)\}$ and $I = \{p(a), p(b)\}$ satisfy (relative to O) the dl-atom $DL[S_1 \uplus p, S_2 \cap\!\!\!\!- p; S_2](b)$ while $\{p(a)\}$ does not. So that $p(b)$ is the only positive nonmonotonic dependency of that dl-atom and then the only canonical (and strong) loop is $L = \{p(a), p(b)\}$. The canonical loop formula $cLF(L, I, \mathcal{K})$ is

$$p(a) \vee p(b) \supset DL[S_1 \uplus p_L, S_2 \cap\!\!\!\!- p_L; S_2](b).$$

We can verify that $LE(I, \mathcal{K}) \models_O cLF(L, I, \mathcal{K})$ and $I \models_O COMP(\mathcal{K})$. Consequently, I is a canonical answer set of $\mathcal{K}$ and also a strong answer set of $\mathcal{K}$. It is not difficult to check that I is not circularly justified.

Proposition 4.6. *Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$ a canonical answer set of $\mathcal{K}$. Then I is a strong answer set of $\mathcal{K}$.*

Proof. Let $I' = LE(I, \mathcal{K})$. Suppose I is not a strong answer set of $\mathcal{K}$. Since $I \models_O COMP(\mathcal{K})$, there must exist a strong loop L of $\mathcal{K}$ such that $I' \not\models_O sLF(L, \mathcal{K})$. It follows that, $I' \models_O \bigvee L$ and

$$I' \not\models_O \bigwedge_{A \in Pos} \gamma(A, L) \wedge \bigwedge_{B \in Neg} \neg B \tag{12}$$

for every rule $(h \leftarrow Pos, not\ Neg)$ in P with $Pos \cap L = \emptyset$. Without loss of generality, we assume L is a minimal one such that $I' \not\models_O sLF(L, \mathcal{K})$.

Let $M = L \cap I$. It is evident that $M \neq \emptyset$ and $I \setminus M = I \setminus L$. Let h' be an atom in M. Because $h' \in I$, there exists at least one rule $(h' \leftarrow Pos', not\ Neg')$ in P such that

$$I \models_O \bigwedge_{A \in Pos'} A \wedge \bigwedge_{B \in Neg'} \neg B.$$

It follows that at least one of the following conditions holds:

- $Pos' \cap L \neq \emptyset$. It shows that there is an atom $h'' \in Pos' \cap M$.
- There is a monotonic dl-atom $A \in Pos'$ such that $I' \not\models_O IF(A, L)$. It shows that $I \setminus L \not\models_O A$ by Lemma 3.4, i.e., $I \setminus M \not\models_O A$. Since $I \models_O A$, there must exist an interpretation I^* and an atom $h'' \in I \setminus (I \setminus M)$, i.e., $h'' \in M$ such that $I^* \models_O A$ and $I^* \setminus \{h''\} \not\models_O A$ by (1) of Lemma 4.1.

So that (h', h'') is an edge of the canonical dependency graph of $\mathcal{K}$. As h' is arbitrary and M is finite, the generated subgraph G' of $G^c_{\mathcal{K}}$ on M must have a terminating canonical loop M'. It is clear that $M' \subseteq M$. Since $M' \subseteq I$ and $I' \models_O cLF(M', I, \mathcal{K})$, there is at least one rule $(h'' \leftarrow Pos'', not\ Neg'')$ in P such that $h'' \in M'$, $Pos'' \cap M' = \emptyset$,

$$I \models_O \bigwedge_{A \in Pos''} A \wedge \bigwedge_{B \in Neg''} \neg B \text{ and } I' \models_O \bigwedge_{A \in Pos''} \delta_1(A, M') \wedge \bigwedge_{B \in Neg''} \neg\delta_2(B, M').$$

It follows that, by Lemmas 3.4 and 4.2,

$$I \setminus M' \models_O \bigwedge_{A \in Pos''} A \wedge \bigwedge_{B \in Neg''} \neg B.$$

However, by Eq. (12) at least one of the following conditions hold:

- $Pos'' \cap L \cap I \neq \emptyset$, i.e., $Pos'' \cap M \neq \emptyset$. It follows that there is an atom $h^* \in Pos'' \cap M$ such that $h^* \in M \setminus M'$.
- There is a monotonic dl-atom $A \in Pos''$ such that $I \setminus M \not\models_O A$. But we know that $I \setminus M' \models_O A$ by Lemma 4.2. It shows that there is an atom $h^* \in M \setminus M'$ such that $I^* \not\models_O A$ and $I^* \cup \{h^*\} \models_O A$ for some interpretation I^* by Lemma 4.1.

It follows that (h'', h^*) is also an edge of $G^c_{\mathcal{K}}$. Since M is finite, G' must have a path from h'' to another canonical loop of G'. It contradicts the fact that M' is a terminating canonical loop of G'. Consequently, I is a strong answer set of $\mathcal{K}$. □

The following example shows that there are noncircular strong answer sets that are not canonical ones.

Example 7. Let $\mathcal{K} = (\emptyset, P)$ be a dl-program where P consists of

$$q(a) \leftarrow DL[c_1 \uplus p, c_2 \cap\!\!\!\!- q; c_1 \sqcup \neg c_2](a),$$
$$p(a) \leftarrow q(a).$$

Consider the interpretation $I = \{p(a), q(a)\}$ which is the unique strong answer set of $\mathcal{K}$. We have that I is the least model of $\mathcal{K}^{s,I}$, so that I is a strong answer set of $\mathcal{K}$. It is not difficult to check that I is not circularly justified. However, we know that $L = I$ is a canonical loop of $\mathcal{K}$. Because $I \models_O DL[c_1 \uplus p, c_2 \cap\!\!\!\!- q; c_1 \sqcup \neg c_2](a)$ and $I \setminus \{p(a)\} \not\models_O DL[c_1 \uplus p, c_2 \cap\!\!\!\!- q; c_1 \sqcup \neg c_2](a)$, $p(a)$ is a positive nonmonotonic dependency of $q(a)$. The canonical loop formula $cLF(L, I, \mathcal{K})$ is the following formula

$$p(a) \vee q(a) \supset DL[c_1 \uplus p_L, c_2 \cap\!\!\!\!- q; c_1 \sqcup \neg c_2](a).$$

It is evident that $LE(I, \mathcal{K}) \not\models_O cLF(L, I, \mathcal{K})$. Thus I is not a canonical answer set of $\mathcal{K}$. In fact, $\mathcal{K}$ has no canonical answer set.

The above example, together with Propositions 4.4 and 4.6, implies the following.

Corollary 4.7. *Each canonical answer set of a dl-program $\mathcal{K}$ is a minimal strong answer set of $\mathcal{K}$, but not vice versa.*

The following proposition, together with Proposition 4.5, implies that the existence of nonmonotonic dl-atoms is the only cause that a strong answer set of a dl-program is circular.

Proposition 4.8. *Let $\mathcal{K} = (O, P)$ be a dl-program in which P does not mention any nonmonotonic dl-atoms. Then $I \subseteq HB_P$ is a canonical answer set of $\mathcal{K}$ if and only if I is a strong answer set of $\mathcal{K}$.*

Proof. Let $I' = LE(I, \mathcal{K})$. By Proposition 4.6, it is sufficient to show that if I is a strong answer set of $\mathcal{K}$ then I is a canonical answer set of $\mathcal{K}$. Suppose I is a strong answer set of $\mathcal{K}$ but I is not a canonical answer set of $\mathcal{K}$. Since $I \models_O COMP(\mathcal{K})$, it follows that there exists at least one canonical loop L of $\mathcal{K}$ such that $I' \not\models_O cLF(L, I, \mathcal{K})$.

Recall that all dl-atoms appearing in P are monotonic. If A is a monotonic dl-atom and there is an atom $p(\vec{c})$ and an interpretation I^* such that $I^* \not\models_O A$ and $I^* \cup \{p(\vec{c})\} \models_O A$ then A must contain $S \uplus p$ (or $S \cup\!\!\!\!- p$) for some S. It follows that L is also a strong loop of $\mathcal{K}$ and then $I' \models_O sLF(L, \mathcal{K})$ by Theorem 3.6, i.e., P has at least one rule $(h \leftarrow Pos, not\ Neg)$ such that $h \in L$, $Pos \cap L = \emptyset$ and

$$I' \models_O \bigwedge_{A \in Pos} \gamma(A, L) \wedge \bigwedge_{B \in Neg} \neg B.$$

Please note that again there are no nonmonotonic dl-atoms in $Pos \cup Neg$. It follows that

$$I' \models_O \bigwedge_{A \in Pos} \delta_1(A, L) \wedge \bigwedge_{B \in Neg} \neg \delta_2(B, L) \ \text{ and } \ I \models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B$$

by Lemma 3.4. Thus we have $I' \models_O cLF(L, I, \mathcal{K})$. This is a contradiction. It contradicts $I' \not\models_O cLF(L, I, \mathcal{K})$. Consequently I is a canonical answer set of $\mathcal{K}$. $\square$

5. Some other semantical considerations

There are two well-known generalizations of dl-programs, HEX-programs [9] and constraint programs [28]. Thus there exist two more different semantics for dl-programs borrowed from HEX-programs and constraint programs respectively. We investigate the relationship among those semantics for dl-programs below.

5.1. FLP-semantics for dl-programs

Dl-programs had been extended to HEX programs that combines answer set programs with higher-order atoms and external atoms [9]. In particular, the external atoms can refer, as dl-atoms in dl-programs, to concepts belonging to a classical knowledge base or ontology. In such a case one can compare the semantics of the HEX program with that of the corresponding dl-program. The semantics of HEX programs is based on the notion of FLP-reduct [10]. We also note that the semantics of dl-programs has been investigated from the perspective of the quantified logic of here-and-there [13]. For the purpose of comparison, let us rephrase the FLP-answer set semantics of dl-programs below.

Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$. The *FLP-reduct* of $\mathcal{K}$ relative to I, written as $\mathcal{K}^{f,I}$, is the dl-program (O, fP_O^I) where fP_O^I is the set of all rules of P whose bodies are satisfied by I relative to O. An interpretation I is an FLP-answer set of a dl-program $\mathcal{K}$ if I is a minimal model of fP_O^I (relative to O).

Proposition 5.1. *The FLP-answer sets of a dl-program are incomparable, i.e., if a set I_2 is an FLP-answer set of $\mathcal{K} = (O, P)$ then I_1 is not an FLP-answer set of $\mathcal{K}$ for any $I_2 \subset I_1$.*

Proof. Suppose $I_1 \subset I_2 \subseteq HB_P$ are two FLP-answer sets of a dl-program $\mathcal{K}$. It is evident that, for every rule $(r : h \leftarrow Pos, not\ Neg)$ in P, if $I_1 \models_O A$ for each $A \in Pos$ and $I_1 \not\models_O B$ for each $B \in Neg$ then $I_1 \models_O h$. Thus I_1 is a model of $fP_O^{I_2}$ which contradicts that I_2 is a minimal model of $fP_O^{I_2}$ due to $I_1 \subset I_2$. $\square$

Proposition 5.2. *Every minimal strong answer set of a dl-program is an FLP-answer set of the dl-program.*

Proof. Let I be a minimal strong answer set of the dl-program $\mathcal{K} = (O, P)$. We have that I satisfies every rule of P (relative to O) and thus I is a model of fP^I_O. Suppose I is not an FLP-answer set of $\mathcal{K}$. It follows that there exists $I' \subset I$ such that I' satisfies each rule in fP^I_O (relative to O). Evidently, we have $s[fP^I_O]^I_O \subseteq sP^I_O$ due to $fP^I_O \subseteq P$ and, for any rule $(h \leftarrow Pos)$ in $sP^I_O \setminus s[fP^I_O]^I_O$, $I \not\models_O A$ for some $A \in Pos$. It follows that $I' \not\models_O A$ since A is an atom or monotonic dl-atom and thus I' satisfies each rule in sP^I_O (relative to O). It implies that I' is a model of sP^I_O. This contradicts that I is a minimal strong answer set of $\mathcal{K}$. □

Recall that canonical answer sets of a dl-program are minimal by Proposition 4.4, and canonical answer sets of a dl-program are strong answer sets by Proposition 4.6. The below proposition follows from Proposition 5.2.

Proposition 5.3. *Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$ be a canonical answer set of $\mathcal{K}$. Then we have that I is also an FLP-answer set of $\mathcal{K}$.*

Let us consider the dl-program $\mathcal{K}$ in Example 7 again. It is not difficult to see that $fP^I_O = P$ and the minimal model of P is I. Thus I is an FLP-answer set of $\mathcal{K}$. We can also verify that I is the unique strong answer set of $\mathcal{K}$. However we know that I is not a canonical answer set of $\mathcal{K}$. It shows that there are some FLP-answer sets of a dl-program that are not canonical answer sets of the dl-program.

It has been shown that the FLP-answer set semantics coincides with the strong answer set semantics of dl-programs that contain no nonmonotonic dl-atoms (Theorem 5 of [9]). The following Proposition asserts that FLP-answer sets of a dl-program are strong answer sets of the dl-program.

Proposition 5.4. *Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$ an FLP-answer set of $\mathcal{K}$. Then we have that I is a strong answer set of $\mathcal{K}$.*

Proof. It is sufficient to show that $I = lfp(\gamma_{\mathcal{K}^{s,I}})$. Let $I' = lfp(\gamma_{\mathcal{K}^{s,I}})$.

($\supseteq$) It is sufficient to prove that $\gamma^k_{\mathcal{K}^{s,I}} \subseteq I$ for every $k \geq 0$. We prove this by induction on k.

Base: It trivially holds for $k = 0$.

Step: Suppose it holds for $k = n$, i.e., $\gamma^n_{\mathcal{K}^{s,I}} \subseteq I$. For each atom $h \in \gamma^{n+1}_{\mathcal{K}^{s,I}}$, there exists a rule $(r : h \leftarrow Pos, Ndl, not\ Neg)$ in P such that (i) $\gamma^n_{\mathcal{K}^{s,I}} \models_O A$ for each $A \in Pos$, (ii) $I \models_O B$ for every $B \in Ndl$ and, (iii) $I \not\models_O C$ for every $C \in Neg$ where Pos is the set of atoms and monotonic dl-atoms occurring in the body of the rule, and Ndl is the set of nonmonotonic dl-atoms occurring in the rule. It follows that $I \models_O A$ for each $A \in Pos$. Thus the rule r is in fP^I_O. Since I is a model of fP^I_O, we have that $h \in I$. Consequently $\gamma^{n+1}_{\mathcal{K}^{s,I}} \subseteq I$ and then $I' \subseteq I$.

($\subseteq$) To show $I \subseteq I'$, it is sufficient to prove that I' is a model of fP^I_O since I is a minimal model of fP^I_O. Let us consider an arbitrary rule $(r : h \leftarrow Pos, not\ Neg)$ of fP^I_O. Note that the rule $(r' : h \leftarrow Pos')$ obtained from r by the strong dl-transformation is in sP^I_O where Pos' is the set of atoms and monotonic dl-atoms in Pos. Recall that I' satisfies the rule r' (relative to O) since $I' = lfp(\gamma_{\mathcal{K}^{s,I}})$, i.e., if I' satisfies (relative to O) the atoms and dl-atoms in Pos' then I' satisfies h. It follows that I' satisfies r (relative to O). Thus I' is a model of fP^I_O. □

If a dl-atom mentions no operator $\cap\!\!\!\!-$ then the dl-atom is monotonic, and if a dl-program $\mathcal{K}$ mentions no nonmonotonic dl-atoms then the FLP-answer sets of $\mathcal{K}$ coincide with the strong answer sets of $\mathcal{K}$. Together with Proposition 4.6, we have the following corollary.

Corollary 5.5. *Let $\mathcal{K} = (O, P)$ be a dl-program such that $DL_P = DL^+_P$ and $I \subseteq HB_P$. Then we have that I is a strong answer set of $\mathcal{K}$ iff I is a canonical answer set of $\mathcal{K}$ iff I is an FLP-answer set of $\mathcal{K}$.*

5.2. Dl-atoms as abstract constraints

Logic programs with abstract constraint atoms (or constraint programs) [21] is a quite general framework for answer set programming [28] and we can translate dl-programs into constraint programs. In the following, we compare the semantics of dl-programs with that of corresponding constraint programs, starting with recalling the basic notations of constraint programs [28].

An *abstract constraint atom (c-atom)* is of the form (D, C), where D is a finite set of atoms and $C \subseteq 2^D$. For a c-atom $A = (D, C)$, we use A_d and A_c to refer to D and C respectively. A is said to be *elementary* if it is of the form $(\{a\}, \{\{a\}\})$, which may be just written as a. We use $\overline{A}$ to denote the *complement* of A, i.e., $\overline{A}_d = A_d$ and $\overline{A}_c = 2^{A_d} \setminus A_c$. Let M be a set of atoms and A a c-atom. *M satisfies A*, written as $M \models A$, if $M \cap A_d \in A_c$.

Let M and S be sets of atoms. The set S *conditionally satisfies* a c-atom A w.r.t. M, denoted by $S \models_M A$ if $S \models A$ and for every I with $S \cap A_d \subseteq I \subseteq M \cap A_d$, $I \in A_c$. Intuitively, $S \models_M A$ implies that $S' \models A$ for every S' such that $S \subseteq S' \subseteq M$.

Formally, a *logic program with c-atoms*, also called a *constraint program* (or just a *program*), is a finite set of rules of the form

$$A \leftarrow A_1, \ldots, A_k, not\ A_{k+1}, \ldots, not\ A_n \tag{13}$$

where A and A_i's are arbitrary c-atoms. The literals $not\ A_j$ are called *negative c-atoms*. For a rule r of the form (13), we define $Head(r) = A$ and $Body(r) = Pos(r) \cup not\ Neg(r)$ where $Pos(r) = \{A_i | 1 \le i \le k\}$ and $Neg(r) = \{A_j | k+1 \le j \le n\}$ and $not\ S = \{not\ A | A \in S\}$ for a set S of c-atoms.

A program is *positive* if it does not mention *not* in its rules. A program is *basic* if, for each rule of the form (13) of P, A is either elementary or $\bot$.[6] The above satisfaction relation for c-atoms is easily extended for programs.

For a set of atoms S and a positive basic program P, we define the operator T_P as following

$$T_P(S, M) = \{p | \exists r \in P, Head(r) = p \neq \bot, S \models_M Body(r)\}.$$

A model M of a positive basic P is an *answer set* of P if $M = T_P^\infty(\emptyset, M)$, where

- $T_P^0(\emptyset, M) = \emptyset$,
- $T_P^{i+1}(\emptyset, M) = T_P(T_P^i(\emptyset, M), M)$.

If P is a positive basic program then the least fixpoint of $T_P(\emptyset, M)$ exists for each $M \subseteq HB_P$. M is an *answer set* of P if M is the least fixpoint of $T_P(\emptyset, M)$. For a basic program P and M a set of atoms, M is an *answer set of P by complement* if M is an answer set of the program P', which is obtained from the rules of P by replacing each $not\ A$ with $\overline{A}$.

Let P be a positive basic program, the completion of P, written as $Comp(P)$, consists of the following formulas:

- $\bigwedge_{A \in Pos} A \supset a$ for every rule $(a \leftarrow Pos)$ in P;
- $a \supset \bigvee_{(a \leftarrow Pos) \in P} \bigwedge_{A \in Pos} A$.

Given a dl-program $\mathcal{K} = (O, P)$, and A an atom or dl-atom appearing in P, we define $\tau(A)$ to be the c-atom where $\tau(A)_d = HB_P$ and $\tau(A)_c = \{M \subseteq HB_P | M \models_O A\}$. In what follows, we denote by $P_{\mathcal{K}}$ the positive basic program obtained from the rules of P by replacing each atom and dl-atom A with $\tau(A)$, and then $not\ \tau(A)$ with $\overline{\tau(A)}$.

Example 8 (*Continued from Example 7*). We have that $P_{\mathcal{K}}$ consists of

$$q(a) \leftarrow (\{p(a), q(a)\}, \{\emptyset, \{p(a)\}, \{p(a), q(a)\}\});$$
$$p(a) \leftarrow q(a).$$

Consider the interpretation $I = \{p(a), q(a)\}$ again. It is not difficult to see that $\emptyset$ is the least fixpoint of $T_{P_{\mathcal{K}}}(\emptyset, I)$. Thus I is not an answer set of $P_{\mathcal{K}}$. It shows that the strong answer sets of dl-programs of a dl-program $\mathcal{K}$ do not correspond to the answer sets (by complement) of $P_{\mathcal{K}}$.

The following lemma is evident.

Lemma 5.6. *Let $\mathcal{K} = (O, P)$ be a dl-program, A an atom or dl-atom appearing in P and $I \subseteq HB_P$. Then $I \models_O A$ iff $I \models \tau(A)$ iff $I \in [\tau(A)]_c$ iff $I \not\models \overline{\tau(A)}$.*

Proposition 5.7. *Let $\mathcal{K} = (O, P)$ be a dl-program and $M \subseteq HB_P$. Then we have $M \models_O COMP(\mathcal{K})$ iff $M \models Comp(P_{\mathcal{K}})$.*

Proof. Note that $I \models_O COMP(\mathcal{K})$ iff, for each $h \in HB_P$, $I \models_O COMP(h, \mathcal{K})$, i.e., $I \models_O h$ iff there is a rule $(h \leftarrow Pos, not\ Neg)$ in P such that

$$I \models_O \bigwedge_{A \in Pos} A \wedge \bigwedge_{B \in Neg} \neg B.$$

By Lemma 5.6, the above holds iff

$$I \models \bigwedge_{A \in Pos} \tau(A) \wedge \bigwedge_{B \in Neg} \overline{\tau(B)}$$

iff

$$I \models \left(\bigwedge_{A \in Pos'} \tau(A) \wedge \bigwedge_{B \in Neg'} \overline{\tau(B)} \right) \supset h$$

for any rule $(h \leftarrow Pos', not\ Neg')$ in P, and

$$I \models h \supset \bigvee_{1 \le i \le m} \left(\bigwedge_{A \in Pos_i} \tau(A) \wedge \bigwedge_{B \in Neg_i} \overline{\tau(B)} \right)$$

where $(h \leftarrow Pos_1, not\ Neg_1), \ldots, (h \leftarrow Pos_m, not\ Neg_m)$ are all the rules in P whose head is h, iff $I \models Comp(P_{\mathcal{K}})$. $\square$

---

[6] The symbol $\bot$ is understood as shorthand for a c-atom of the form $(D, \emptyset)$.

Algorithm 3 *PMin*(A, I, L)

//precondition: not *derivable*$(A, I \setminus L, I)$, $I \models_O A$ and $I \setminus L \models_O A$

$I^* \leftarrow I$
for all $h \in I^* \cap L$ s.t not *derivable*$(A, I \setminus L, I^* \setminus \{h\})$ **do**
  $I^* \leftarrow I^* \setminus \{h\}$
  // assert: not *derivable*$(A, I \setminus L, I^*)$
end for
return I^*

Lemma 5.8. *Let $\mathcal{K} = (O, P)$ be a dl-program, let $A \in DL^?_P$, and let L a nonempty set of atom in HB_P and $I \subseteq HB_P$.*

(1) *If $I \models_O A$, $I \setminus L \models_O A$ and $LE(I, \mathcal{K}) \not\models_O pCF(A, L)$ then there is a set I^* with $I \setminus L \subset I^* \subset I$ such that $I^* \not\models_O A$.*
(2) *If $I \not\models_O A$, $I \setminus L \not\models_O A$ and $LE(I, \mathcal{K}) \models_O nCF(A, L)$ then there is a set I^* with $I \setminus L \subset I^* \subset I$ such that $I^* \models_O A$.*

Proof. Let $I' = LE(I, \mathcal{K})$, and for clarity and without loss of generality, suppose $A = DL[S_1 \uplus p_1, S_2 \cap\!\!\!\!- \, p_2; Q](\vec{t})$.

(1) If there is no atom $h \in L$ such that h is a positive nonmonotonic dependency of A, then $pCF(A, L) = A$. In this case, we obviously have $I' \models_O pCF(A, L)$ since $I \models_O A$ which contradicts $I' \not\models_O pCF(A, L)$. Thus there is at least one atom in L which is a positive nonmonotonic dependency of A. If $p_1 = p_2$ then we have that $pCF(A, L) = DL[S_1 \uplus p_{1_L}, S_2 \cap\!\!\!\!- \, p_{2_L}; Q](\vec{t})$. From $I' \not\models_O pCF(A, L)$, we have $I \setminus L \not\models_O A$ which is a contradiction. So $p_1 \neq p_2$ and we have that

$$O \cup \{S_1(\vec{e}) | p_{1_L}(\vec{e}) \in I'\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I'\} \not\models Q(\vec{t})$$

which implies that, by the fact that $p_{1_L}(\vec{e}) \in I'$ iff $p_1(\vec{e}) \in I \setminus L$,

$$O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in I \setminus L\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I\} \not\models Q(\vec{t}). \tag{14}$$

Recall that $I \models_O A$ and $I \setminus L \models_O A$. It follows that L must contain two atoms $p_1(\vec{c})$ and $p_2(\vec{c}\,')$ in I. For clarity and without loss of generality, we assume both I and L mention only the atoms of the form $p_1(\vec{c})$ and $p_2(\vec{c})$ for some $\vec{c}$. Let M and N be two sets of atoms. We denote *derivable*(A, M, N) for

$$O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in M\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin N\} \models Q(\vec{t}).$$

Consider the set I^* returned by Algorithm 3: *PMin*(A, I, L). It is evident that *derivable*$(A, I \setminus L, I^*)$ is always *false* in the **forall** loop, $I \setminus L \subset I^*$ since $I \setminus L \models_O A$, $I^* \subset I$ since $I \models_O A$ and, I^* contains no atom $p_1(\vec{c})$ which belongs to $I \cap L$. It follows that, for each atom $p_1(\vec{c}) \in I^*$, $p_1(\vec{c}) \in I$ and $p_1(\vec{c}) \notin L$, i.e., $p_1(\vec{c}) \in I \setminus L$. So that *derivable*$(A, I^*, I^*)$ is *false*, i.e., $I^* \not\models_O A$.

(2) If there is no atom $h \in L$ such that h is a negative nonmonotonic dependency of A, then $nCF(A, L) = A$. In this case, we obviously have $I' \not\models_O nCF(A, L)$ since $I \not\models_O A$ which is a contradiction. Thus there is at least one atom in L which is a negative nonmonotonic dependency of A. If $p_1 = p_2$ then we have that $nCF(A, L) = DL[S_1 \uplus p_{1_L}, S_2 \cap\!\!\!\!- \, p_{2_L}; Q](\vec{t})$. From $I' \models_O nCF(A, L)$, we have $I \setminus L \models_O A$ which is a contradiction too. So that $p_1 \neq p_2$ and we have that

$$O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in I'\} \cup \{\neg S_2(\vec{e}) | p_{2_L}(\vec{e}) \notin I'\} \models Q(\vec{t})$$

which implies that, by $p_{2_L}(\vec{e}) \in I'$ iff $p_2(\vec{e}) \in I \setminus L$,

$$O \cup \{S_1(\vec{e}) | p_1(\vec{e}) \in I\} \cup \{\neg S_2(\vec{e}) | p_2(\vec{e}) \notin I \setminus L\} \models Q(\vec{t}). \tag{15}$$

Recall that $I \not\models_O A$ and $I \setminus L \not\models_O A$. It follows that L must contain two atoms $p_1(\vec{c})$ and $p_2(\vec{c}\,')$. For clarity and without loss of generality, we assume both I and L mention only the atoms of the form $p_1(\vec{c})$ and $p_2(\vec{c})$ for some $\vec{c}$. Consider the set I^* returned by Algorithm 4: *NMin*(A, I, L). It is evident that *derivable*$(A, I^*, I \setminus L)$ is always *true* in the **forall** loop since $I' \models_O nCF(A, L)$, $I \setminus L \subset I^*$ since $I \setminus L \not\models_O A$, $I^* \subset I$ since $I \not\models_O A$, and I^* contains no atom $p_2(\vec{c})$ which belongs to $I \cap L$. It follows that, for each atom $p_2(\vec{c}) \in I^*$, $p_2(\vec{c}) \in I \setminus L$. It follows that *derivable*(A, I^*, I^*) is *true*, i.e., $I^* \models_O A$. □

Theorem 5.9. *Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$. Then we have that I is a canonical answer set of $\mathcal{K}$ if I is an answer set of $P_{\mathcal{K}}$ by complement.*

Algorithm 4 *NMin*(A, I, L)

// precondition: *derivable*$(A, I, I \setminus L)$, $I \not\models_O A$ and $I \setminus L \not\models_O A$

$I^* \leftarrow I$
for all $h \in I^* \cap L$ s.t *derivable*$(A, I^* \setminus \{h\}, I \setminus L)$ **do**
  $I^* \leftarrow I^* \setminus \{h\}$
  // assert: *derivable*$(A, I^*, I \setminus L)$
end for
return I^*

Proof. Let $I' = LE(I, \mathcal{K})$ and $I^* = T^{\infty}_{P_{\mathcal{K}}}(\emptyset, I)$. By Proposition 5.7, $I \models_O COMP(\mathcal{K})$ since $I \models Comp(P_{\mathcal{K}})$. It is sufficient to prove that $I' \models_O cLF(L, I, \mathcal{K})$ for every canonical loop L of $\mathcal{K}$. Suppose there is a canonical loop L of $\mathcal{K}$ such that $I' \not\models_O cLF(L, I, \mathcal{K})$, i.e., $I' \models_O \bigvee L$ and

$$I' \not\models_O \bigwedge_{A \in Pos} \delta_1(A, L) \wedge \bigwedge_{B \in Neg} \neg\delta_2(B, L) \tag{16}$$

for every rule $(h \leftarrow Pos, not\ Neg)$ in P such that $h \in L$, $Pos \cap L = \emptyset$, $I \models_O A$ for every $A \in Pos$ and $I \not\models_O B$ for each $B \in Neg$.

Let $M = L \cap I$. We have $I \setminus M = I \setminus L$. Since $I = I^*$, if follows that, for each $h \in M$, there is the least number k_1 such that $h \in T^{k_1+1}_{P_{\mathcal{K}}}(\emptyset, I)$, i.e., P has a rule $(h \leftarrow Pos', not\ Neg')$ such that

$$T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \models_I \bigwedge_{A \in Pos'} \tau(A) \wedge \bigwedge_{B \in Neg'} \overline{\tau(B)}. \tag{17}$$

As I is a model of $P_{\mathcal{K}}$, $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \subseteq I$ by Corollary 1 of [28]. By the definition of conditional satisfaction and (17), we have that, for any set $S \subseteq HB_P$ with $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \subseteq S \subseteq I$, $S \models \tau(A)$ for every $A \in Pos'$, and $S \models \overline{\tau(B)}$ for each $B \in Neg'$, it follows that $S \models_O \bigwedge_{A \in Pos'} A \wedge \bigwedge_{B \in Neg'} \neg B$ by Lemma 5.6. By (16) and (17), at least one of the following cases holds:

- $Pos' \cap L \neq \emptyset$ which implies $Pos' \cap M \neq \emptyset$. So that there is an atom $h_1 \in Pos' \cap M$ such that $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \cap M$.
- There is a dl-atom $A \in Pos'$ such that $I' \not\models_O \delta_1(A, L)$. If A is monotonic then $\delta_1(A, L) = IF(A, L)$. It follows $I \setminus L \not\models_O A$ by Lemma 3.4, i.e. $I \setminus M \not\models_O A$. Thus $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\subseteq I \setminus M$ since $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \models_I \tau(A)$, i.e. $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \models_O A$ by Lemma 5.6. So there is an atom $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \cap M$. It is evident that $h_1 \neq h$.

  If A is nonmonotonic then $\delta_1(A, L) = pCF(A, L)$. In the case $I \setminus L \not\models_O A$, we have $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\subseteq I \setminus M$ since $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \models_I \tau(A)$. As the above discussion, there is an atom $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \cap M$ and $h_1 \neq h$. In the case $I \setminus L \models_O A$, due to $I \models_O A$ and $I' \not\models_O pCF(A, L)$, it follows that there is a set I'' with $I \setminus L \subset I'' \subset I$ such that $I'' \not\models_O A$ by Lemma 5.8. We then have $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\subseteq I \setminus L$, i.e., $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\subseteq I \setminus M$. Thus there is an atom $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \setminus (I \setminus M)$, i.e., $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \cap M$. Obviously, $h_1 \neq h$.
- There is a nonmonotonic dl-atom $B \in Neg'$ such that $I' \models_O nCF(B, L)$. Evidently $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \models_I \overline{\tau(B)}$. It follows that $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\models_O B$ and $I \not\models_O B$ by Lemma 5.6 and the definition of conditional satisfaction. If $I \setminus L \models_O A$ then $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\subseteq I \setminus M$. If $I \setminus L \not\models_O A$ then there is a set I'' with $I \setminus L \subset I'' \subset I$ such that $I'' \models_O A$. This implies that $T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \not\subseteq I \setminus L$. Thus there is an atom $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \setminus (I \setminus M)$, i.e., $h_1 \in T^{k_1}_{P_{\mathcal{K}}}(\emptyset, I) \cap M$ and $h_1 \neq h$.

It follows that there is the least number k_2 such that $h_1 \in T^{k_2+1}_{P_{\mathcal{K}}}(\emptyset, I)$ and $k_2 < k_1$. Consequently, we can construct a sequence $(h_0(= h), h_1, \ldots)$ of atoms in M and a sequence $(k_1, k_2, \ldots)$ of integers such that

- k_i is the least number such that $h_{i-1} \in T^{k_i+1}_{P_{\mathcal{K}}}(\emptyset, I)$, and
- $k_i > k_j$ if $i < j$.

Since M is finite, there exist some i, j $(1 \leq i < j)$ such that $h_i = h_j$. It follows that $k_i = k_j$. This is a contradiction. Thus $I' \models_O cLF(L, I, \mathcal{K})$ and then I is a canonical answer set of $\mathcal{K}$. □

However, as illustrated by the next example, the converse of Theorem 5.9 does not hold in general.

Example 9 (*Continued from Example 6*). We have that $P_{\mathcal{K}}$ consists of

$$p(b) \leftarrow p(a),$$
$$p(a) \leftarrow (\{p(a), p(b)\}, \{\emptyset, \{p(a)\}, \{p(a), p(b)\}\}).$$

The only models of $P_{\mathcal{K}}$ are $\{p(b)\}$ and $\{p(a), p(b)\}$. However, $lfp(T_{P_{\mathcal{K}}}(\emptyset, \{p(b)\})) = lfp(T_{P_{\mathcal{K}}}(\emptyset, \{p(a), p(b)\})) = \emptyset$. Thus neither $\{p(b)\}$ nor $\{p(a), p(b)\}$ is an answer set of $P_{\mathcal{K}}$. Consequently, $P_{\mathcal{K}}$ has no answer set, though $\{p(a), p(b)\}$ is a canonical answer set of $\mathcal{K}$.

The above example motivates us to consider the difference between canonical loops (resp. canonical loop formulas) of dl-programs and loops (resp. loop formulas) of constraint logic programs defined in [33]. To relate canonical loops and canonical loop formulas of dl-programs with those of constraint programs, let us recall some basic notions of constraint programs from [33]. Let A be a c-atom. A pair of sets $\langle B, T\rangle$, where $B \subseteq T \subseteq A_d$, is called a *local power set (LPS)* of A, if for every set M such that $B \subseteq M \subseteq T$, $M \in A_c$. Local power sets are called *prefixed power sets* in [26]. A local power set $\langle B, T\rangle$ of a c-atom A is *maximal* if there is no other power set $\langle B', T'\rangle$ such that $B' \subseteq B$ and $T \subseteq T'$. The LPS representation of A, denoted by A^*, is (A_d, A_c^*), where $A_c^* = \{\langle B, T\rangle | \langle B, T\rangle$ is a maximal LPS of $A\}$. In the following, we focus on basic and positive constraint programs.

Let P be a basic and positive constraint program. The dependency graph of P, written as G^a_P, is the directed graph (V, E) where V is the set of atoms occurring in P and $(u, v) \in E$ if there is a rule $(u \leftarrow Body)$ in P such that $v \in B$ for some c-atom $A \in Body$ and some $\langle B, T\rangle \in A^*_c$. A set L of atoms is a loop of P if there is a cycle in G^a_P which goes through only and all the nodes in L.

Let P be a basic and positive constraint program, A a c-atom, and $L \subseteq HB_P$. The *restriction of* A^* *to* L, written as $A^*_{|L}$, is the c-atom $(A_d, A^*_{c|L})$, where

$$A^*_{c|L} = \{\langle B, T\rangle \in A^*_c | L \cap B = \emptyset\}. \tag{18}$$

By $\pi(A, L)$ we denote the formula $\bigvee_{\langle B,T\rangle \in A^*_{c|L}} \left(\bigwedge B \wedge \bigwedge_{\beta \in (A_d \setminus T)} \neg\beta\right)$.

The *loop formula* of a loop L of a basic and positive constraint P, written as $LP(L, P)$, is the formula

$$\bigvee L \supset \bigvee_{1 \le i \le n} \left(\bigwedge_{A \in Body_i} \pi(A, L)\right), \tag{19}$$

where $(h_i \leftarrow Body_i)(1 \le i \le n)$ are all the rules in P such that $h_i \in L$. It has been shown that a set of atoms M is an answer set by complement of a constraint program P if and only if M is a model of its completion and loop formulas (cf. Theorem 2 of [33]).

Proposition 5.10. *Let $\mathcal{K} = (O, P)$ be a dl-program and $L \subseteq HB_P$. L is a canonical loop of $\mathcal{K}$ if and only if L is a loop of $P_{\mathcal{K}}$ in the sense of [33].*

Proof. $(\Rightarrow)$ (u, v) is an edge of $G^c_{\mathcal{K}}$ implies that there is a rule $(u \leftarrow Pos, not\ Neg)$ in P such that at least one of the following conditions hold:

(1) There is an atom $v \in Pos$. This implies (u, v) is also an edge of $G^a_{P_{\mathcal{K}}}$.
(2) There is a dl-atom $A \in Pos$ such that v is a positive monotonic (or nonmonotonic) dependency of A. It follows that there is an $I^* \subseteq HB_P$ such that $I^* \models_O A$ and $I^* \setminus \{v\} \not\models_O A$. Thus there is a minimal such I^*, i.e., $\langle I^*, T\rangle \in [\tau(A)]^*_c$ for some $T \subseteq HB_P$. It follows that (u, v) is also an edge of $G^a_{P_{\mathcal{K}}}$.
(3) There is a nonmonotonic dl-atom $B \in Neg$ such that v is a negative nonmonotonic dependency of B. It follows that there is an $I^* \subseteq HB_P$ such that $I^* \setminus \{v\} \models_O B$ and $I^* \not\models_O B$. Thus there is a maximal such I^*. It follows that $I^* \in \overline{[\tau(B)]_c}$. By $I^* \setminus \{v\} \notin \overline{[\tau(B)]_c}$, there is a minimal subset I'' of I^* such that, for every I' with $I'' \subseteq I' \subseteq I^*$, $I' \not\models_O B$. It is evident that $v \in I''$ (otherwise we have $I' = I^* \setminus \{v\}$ with $I'' \subseteq I' \subseteq I^*$ such that $I' \models_O B$). It shows that there is $\langle I'', T\rangle \in \overline{[\tau(B)]}^*_c$ such that $v \in I''$. Thus (u, v) is also an edge of $G^a_{P_{\mathcal{K}}}$.

$(\Leftarrow)$ (u, v) is an edge of $G^a_{P_{\mathcal{K}}}$ implies that there is a rule $(u \leftarrow Pos, not\ Neg)$ in P such that at least one of the following conditions hold:

(1) There is some $A \in Pos$ such that $v \in I^*$ for some $\langle I^*, T\rangle \in [\tau(A)]^*_c$. If A is an atom then (u, v) is also an edge of $G^c_{\mathcal{K}}$. Suppose A is a dl-atom. From $I^* \setminus \{v\} \notin A_c$ we have that $I^* \setminus \{v\} \not\models_O A$ by Lemma 5.6. Thus v is a positive monotonic (or nonmonotonic) dependency of A and then (u, v) is an edge of $G^c_{\mathcal{K}}$.
(2) There is some $B \in Neg$ such that $v \in I^*$ for some $\langle I^*, T\rangle \in \overline{[\tau(B)]}^*_c$. It follows that $I^* \not\models_O B$ and $I^* \setminus \{v\} \models_O B$ by Lemma 5.6. If B is an atom then, for every set I of atoms such that $B \subseteq I \subseteq HB_P$, $I \in [\tau(B)]_c$. It shows that the only $\langle I^*, T\rangle \in [\tau(B)]^*_c$ is $I^* = \emptyset$ and $T = HB_P \setminus \{B\}$. Thus B cannot be an atom. If B is a monotonic dl-atom then, for every $I \models_O B$, $I \cup M \models_O B$ for every $M \subseteq HB_P$. It shows that, for every $\langle I^*, T\rangle \in [\tau(B)]^*_c$, $T = HB_P$. Thus, for any $\langle I^*, T\rangle \in \overline{[\tau(B)]}^*_c$, $I^* = \emptyset$. Thus B cannot be nonmonotonic. Consequently, v is a negative monotonic dependency of B and then (u, v) is an edge of $G^c_{\mathcal{K}}$.

It follows that the canonical dependency graph $G^c_{\mathcal{K}}$ is identical to the dependency graph G^a_P. Thus L is a canonical loop of $\mathcal{K}$ if and only L is a loop of $P_{\mathcal{K}}$. This completes the proof. □

The above proposition, together with Theorem 5.9 Theorem 2 of [33], implies that loop formulas cause the difference between canonical answer set semantics of dl-programs and the answer set semantics of the corresponding constraint programs. For instance, for the dl-program $\mathcal{K}$ in Example 6 (whose translation to a constraint program is given in Example 9), we have that $I = L = \{p(a), p(b)\}$ is a canonical loop of $\mathcal{K}$ and a loop of $P_{\mathcal{K}}$. The loop formula $LP(L, P_{\mathcal{K}})$ is

$$p(a) \vee p(b) \supset \neg p(b)$$

since

$$\tau(A)^*_c = \{\langle\emptyset, \{p(a)\}\rangle, \langle\{p(a)\}, \{p(a), p(b)\}\rangle\}$$

where $A = DL[S_1 \uplus p, S_2 \cap\!\!\!\!- p; S_2](a)$. It is evident that $I \not\models LP(L, P_{\mathcal{K}})$.

Lemma 5.11. *Let $\mathcal{K} = (O, P)$ be a dl-program, A an atom or monotonic dl-atom, $L \subseteq HB_P$ and $I \subseteq HB_P$. Then we have that*

(i) *If $I \setminus L \models_O A$ then $I \models \pi(\tau(A), L)$.*
(ii) *If $I \not\models_O A$ then $I \models \pi(\overline{\tau(A)}, L)$.*

Proof. (i) $I \setminus L \models_O A$
$\Rightarrow (I \setminus L) \cap \tau(A)_d \in \tau(A)_c$
$\Rightarrow (I \setminus L) \in \tau(A)_c$
$\Rightarrow$ there exists $\langle B, HB_P\rangle \in \tau(A)^*_c$ such that $B \subseteq I \setminus L$ since A is monotonic
$\Rightarrow B \cap L = \emptyset$ and $I \models \bigwedge B$
$\Rightarrow I \models \pi(\tau(A), L)$.
(ii) $I \not\models_O A$
$\Rightarrow I \models \overline{\tau(A)}$
$\Rightarrow$ there exists $\langle \emptyset, I'\rangle \in \overline{\tau(A)}^*_c$ such that $I \subseteq I'$ since A is monotonic
$\Rightarrow I \models \bigwedge_{\beta\in\tau(A)_d\setminus I'} \neg\beta$
$\Rightarrow I \models \pi(\overline{\tau(A)}, L)$ since $\emptyset \cap L = \emptyset$. $\square$

The next proposition shows that if a dl-program $\mathcal{K}$ has no nonmonotonic dl-atoms then the canonical answer sets of $\mathcal{K}$ coincide with the answer sets of $P_{\mathcal{K}}$.

Proposition 5.12. *Let $\mathcal{K} = (O, P)$ be a dl-program such that $DL_P = DL_P^+$ and $I \subseteq HB_P$. Then we have that I is a canonical answer set $\mathcal{K}$ if and only if I is an answer set of $P_{\mathcal{K}}$.*

Proof. ($\Leftarrow$) It follows from Theorem 5.9.

($\Rightarrow$) Suppose I is not an answer set of $P_{\mathcal{K}}$ but I is a canonical answer set of $\mathcal{K}$. It follows that there is a loop L of $P_{\mathcal{K}}$ such that $I \not\models LP(L, P_{\mathcal{K}})$ by Theorem 2 of [33]. It follows that $I \models \bigvee L$ and, for each rule ($h \leftarrow Pos, not\ Neg$) of P with $h \in L$,

$$I \not\models \bigwedge_{A\in Pos} \pi(\tau(A), L) \wedge \bigwedge_{B\in Neg} \pi(\overline{\tau(B)}, L). \tag{20}$$

By Proposition 5.10, we have L is also a canonical loop of $\mathcal{K}$ and then $LE(I, \mathcal{K}) \models_O cLF(L, I, \mathcal{K})$ since I is a canonical answer set of $\mathcal{K}$. It follows that there exists at least one rule ($r : h' \leftarrow Pos', not\ Neg'$) in P such that $Pos' \cap L = \emptyset$, $I \models_O A'$ for each $A' \in Pos'$, $I \not\models_O B'$ for each $B' \in Neg'$ and

$$LE(I, \mathcal{K}) \models_O \bigwedge_{A'\in Pos'} \delta_1(A', L) \wedge \bigwedge_{B'\in Neg'} \delta_2(B', L). \tag{21}$$

Recall that there is no nonmonotonic dl-atoms in $Pos' \cup Neg'$. By Eq. (20) we have that

$$LE(I, \mathcal{K}) \models_O \bigwedge_{A'\in Pos'} IF(A', L) \wedge \bigwedge_{B'\in Neg'} \neg B'$$

which implies that, by Lemma 3.4,

$$I \setminus L \models_O \bigwedge_{A'\in Pos'} A' \ \text{ and } \ I \models_O \bigwedge_{B'\in Neg'} \neg B'.$$

which further implies that, by Lemma 5.11,

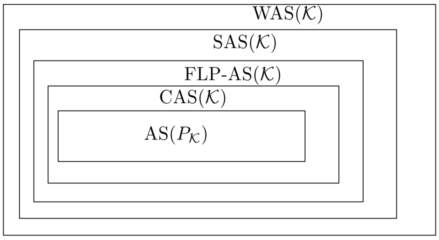
$$I \models \bigwedge_{A'\in Pos'} \pi(\tau(A'), L) \wedge \bigwedge_{B'\in Neg'} \pi(\overline{\tau(B')}, L).$$

This contradicts Eq. (20). Consequently, I is an answer set of $P_{\mathcal{K}}$. $\square$

It is straightforward to see that the FLP semantics is applicable to constraint programs. Additionally, Ferraris proposes an answer set semantics for arbitrary proposition theories, which is also applicable to constraint programs by taking c-atoms as formulas [11,12]. Of course, different treatment of c-atoms as formulas may lead to different semantics. The two semantics coincide for positive basic constraint programs [27], thus they coincide for dl-programs mentioning no *not* dl-atoms, i.e., no dl-atom α occurs in the form *not* α.

As we know, given a dl-program $\mathcal{K}$, every strong answer set of $\mathcal{K}$ is a weak answer set of $\mathcal{K}$ (Theorem 4.23 of [8]). Together with Propositions 5.3 and 5.4, as well as Theorem 5.9, the relationship among these semantics of dl-programs is summarized in Fig. 2.

Please note that Propositions 4.8 and 5.4 imply that for dl-programs containing no nonmonotonic dl-atoms, the canonical answer set semantics coincides with the FLP-answer set semantics. From Theorem 5 of [9] and Proposition 5.12, it follows that if a dl-program $\mathcal{K}$ contains no nonmonotonic dl-atoms then SAS($\mathcal{K}$)=FLP-AS($\mathcal{K}$)=CAS($\mathcal{K}$)=AS($P_{\mathcal{K}}$) where SAS($\mathcal{K}$), FLP-AS($\mathcal{K}$), CAS($\mathcal{K}$) and AS($P_{\mathcal{K}}$) denotes the set of answer sets of $\mathcal{K}$ as illustrated in Fig. 2.



WAS($\mathcal{K}$) – The set of weak answer sets of $\mathcal{K}$; SAS($\mathcal{K}$) – The set of strong answer sets of $\mathcal{K}$; FLP-AS($\mathcal{K}$) – The set of FLP-answer sets of $\mathcal{K}$; CAS($\mathcal{K}$) – The set of canonical answer sets of $\mathcal{K}$; AS($P_{\mathcal{K}}$) – The set of answer sets of $P_{\mathcal{K}}$.

Fig. 2. The relationship among the answer sets of a dl-program $\mathcal{K}$.

6. Concluding remarks and future work

Integrating ASP with description logics has attracted a great deal of attention recently. The existing approaches can be roughly classified into three categories. The first is to adopt a nonmonotonic formalism that covers both ASP and first-order logic (if not for the latter, then extend it to the first-order case) [5,19,22], where ontologies and rules are written in the same language, resulting in a tight coupling. The second is a loose approach: an ontology knowledge base and the rules share the same constants but not the same predicates, and the communication is via a well-defined interface, such as dl-atoms [8]. The third is to combine ontologies with hybrid rules [6,24,25], where predicates in the language of ontologies are interpreted classically, whereas those in the language of rules are interpreted nonmonotonically.

Although each approach above has its own merits, the loose approach possesses some unique advantages. In many situations, we would like to combine existing knowledge bases, possibly under different logics. In this case, a notion of interface is natural and necessary. The loose approach seems particularly intuitive, as it does not rely on the use of modal operators nor on a multi-valued logic. One notices that dl-programs share similar characteristics with another recent interest, *multi-context systems*, in which knowledge bases of arbitrary logics communicate through *bridge rules* [4].

However, the relationships among these different approaches are currently not well understood. For example, although we know how to translate a dl-program without the nonmonotonic operator $\cap\!\!\!\!\!-$ to an MKNF theory while preserving the strong answer set semantics [22], when $\cap\!\!\!\!\!-$ is involved, no such a translation is known. Similarly, although a variant of Quantified Equilibrium Logic (QEL) captures the existing hybrid approaches, as shown by [6], it is not clear how one would apply the loop formulas for logic programs with arbitrary sentences [15] to dl-programs, since, to the best of our knowledge, there is no syntactic, semantics-preserving translation from dl-programs to logic programs with arbitrary sentences or to QEL.

In fact, the loop formulas for dl-programs are more involved than any previously known loop formulas, due to mixing ASP with classical first-order logic. This is evidenced by the fact that weak loop formulas permit self-supports, strong loop formulas eliminate certain kind of self-supports, and canonical loop formulas remove more self-supports.

In this paper, we have characterized the weak and strong answer sets of dl-programs by program completion and loop formulas. Although these loop formulas also provide an alternative mechanism for computing answer sets, building such a system presents itself as an interesting future work. We have also proposed the canonical answer sets for dl-programs, which are minimal and noncircular in a formal sense. From the perspective of loop formulas, we see a notable distinction among the weak, strong and canonical answer sets: the canonical answer sets permit no circular justifications in the sense that canonical answer sets are always noncircular, the strong answer sets permit circular justifications involving nonmonotonic dl-atoms but not monotonic ones, whereas the weak answer sets permit circular justifications that may involve any dl-atoms but not atoms.

Unfortunately, as illustrated by Example 6, the canonical answer set semantics does not exclude all self-supports which also means that the notion of circular justification proposed in the paper is not sufficient to capture the phenomena of self-supports. Whether there exists a syntactic style definition of loop formulas for dl-programs that excludes all self-supports is worthy of further study, in addition to looking for a more restricted notion of circular justification that can capture the phenomena of self-supports.

The more recently adopted semantics for dl-programs is the FLP-answer set semantics. We proved that canonical answer sets are FLP-answer sets and the FLP-answer set semantics permits some self-supports that are excluded in the canonical answer set semantics. Since logic programs with abstract constraints is a general formalism for answer set programming, we have shown that dl-programs can be intuitively mapped to positive and basic logic programs with abstract constraints and proved that, for a dl-program $\mathcal{K}$, the answer sets of the corresponding logic programs with abstract constraints are canonical answer sets of $\mathcal{K}$, but not vice versa. This reveals some interesting relationships among the semantics for dl-programs considered in this paper. We have also revealed that for the dl-programs containing no nonmonotonic dl-atoms, all the semantics coincide with each other except for the weak answer set one.

We remark that, for a given dl-program $\mathcal{K} = (O, P)$, to decide whether a set $M \subseteq HB_P$ is a strong or canonical loop and to construct the strong or canonical loop formula of M are generally quite difficult, since we have to decide the monotonicity of the dl-atoms occurring in P. The exact complexity of deciding whether a set of atoms is a strong or canonical loop requires further research, in addition to the complexity of deciding whether a given dl-program has a canonical answer set.

Acknowledgements

We thank the reviewers for their detailed comments, which helped improve the presentation of this paper. Yisong Wang and Mingyi Zhang are partially supported by NSFC grants 60963009 and 90718009 and the Fund of Science and Technology of Guizhou: 2008[2119]; Yisong is also partially supported by the Fund of Education Department of Guizhou Province: 2008[011]. The work of Jia-Huai You and Li Yan Yuan were carried out under the support by NSERC discovery grants. Yi-Dong Shen is supported in part by NSFC grant 60970045.

References

[1] M. Abadi, L. Cardelli, A Theory of Objects, Springer, 1996.
[2] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, The Description Logic Handbook: Theory, Implementation, and Applications, 2nd ed., Cambridge University Press, New York, NY, 2007.
[3] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American Magazine 284 (2001) 34–43.
[4] G. Brewka, T. Eiter, Equilibria in heterogeneous nonmonotonic multi-context systems, in: Proceedings of the Twenty-second AAAI Conference on Artificial Intelligence, AAAI 2007, AAAI Press, Vancouver, British Columbia, Canada, 2007, pp. 385–390.
[5] J. de Bruijn, T. Eiter, A. Polleres, H. Tompits, Embedding non-ground logic programs into autoepistemic logic for knowledge-base combination, in: Proceedings of International Joint Conference On Artificial Intelligence, IJCAI-07, Hyderabad, India, 2007, pp. 304–309.
[6] J. de Bruijn, D. Pearce, A. Polleres, A. Valverde, Quantified equilibrium logic and hybrid rules, in: Web Reasoning and Rule Systems, First International Conference, RR 2007, in: Lecture Notes in Computer Science, vol. 4524, Springer, Innsbruck, Austria, 2007, pp. 58–72.
[7] F. Calimeri, W. Faber, N. Leone, S. Perri, Declarative and computational properties of logic programs with aggregates, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, IJCAI 2005, Professional Book Center, Edinburgh, Scotland, UK, 2005, pp. 406–411.
[8] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, H. Tompits, Combining answer set programming with description logics for the semantic web, Artifical Intelligence 172 (2008) 1495–1539.
[9] T. Eiter, G. Ianni, R. Schindlauer, H. Tompits, A uniform integration of higher-order reasoning and external evaluations in answer-set programming, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), Professional Book Center, Edinburgh, Scotland, UK, 2005, pp. 90–96.
[10] W. Faber, N. Leone, G. Pfeifer, Recursive aggregates in disjunctive logic programs: Semantics and complexity, in: Logics in Artificial Intelligence, 9th European Conference, JELIA 2004, in: Lecture Notes in Computer Science, vol. 3229, Springer, Lisbon, Portugal, 2004, pp. 200–212.
[11] P. Ferraris, Answer sets for propositional theories, in: Logic Programming and Nonmonotonic Reasoning, 8th International Conference, LPNMR 2005, in: Lecture Notes in Computer Science, vol. 3662, Springer, Diamante, Italy, 2005, pp. 119–131.
[12] P. Ferraris, Logic programs with propositional connectives and aggregates, ACM Transactions on Computational Logic (TOCL), 12 25 (2011) 1–25:40.
[13] M. Fink, D. Pearce, A logical semantics for description logic programs, in: Logics in Artificial Intelligence - 12th European Conference, JELIA 2010, in: Lecture Notes in Computer Science, vol. 6341, Springer, Helsinki, Finland, 2010, pp. 156–168.
[14] I. Horrocks, P.F. Patel-Schneider, A proposal for an OWL rules language, in: Proceedings of the 13th international conference on World Wide Web (WWW 2004), ACM, New York, NY, USA, 2004, pp. 723–731.
[15] J. Lee, Y. Meng, On loop formulas with variables, in: Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008), AAAI Press, Sydney, Australia, 2008, pp. 444–453.
[16] N. Leone, G. Pfeifer, W. Faber, T. Eiter, G. Gottlob, S. Perri, F. Scarcello, The dlv system for knowledge representation and reasoning, ACM Transactions on Computational Logic 7 (2006) 499–562.
[17] F. Lin, Y. Zhao, ASSAT: computing answer sets of a logic program by SAT solvers, Artificial Intelligence 157 (2004) 115–137.
[18] G. Liu, J.-H. You, Lparse programs revisited: Semantics and representation of aggregates, in: Logic Programming, 24th International Conference (ICLP 2008), in: Lecture Notes in Computer Science, vol. 5366, Springer, Udine, Italy, 2008, pp. 347–361.
[19] T. Lukasiewicz, A novel combination of answer set programming with description logics for the semantic web, IEEE Transactions on Knowledge and Data Engineering 22 (2010) 1577–1592.
[20] V.W. Marek, M. Truszczynski, Stable models and an alternative logic programming paradigm, in: K. Apt, V. Marek, M. Truszczynski, D. Warren (Eds.), The Logic Programming Paradigm: A 25-Year Perspective, Springer-Verlag, Berlin, 1999, pp. 375–398.
[21] V.W. Marek, M. Truszczynski, Logic programs with abstract constraint atoms, in: Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence (AAAI 2004), AAAI Press, San Jose, California, USA, 2004, pp. 86–91.
[22] B. Motik, R. Rosati, Reconciling description logics and rules, Journal of the ACM 57 (2010) 1–62.
[23] I. Niemelä, Logic programs with stable model semantics as a constraint programming paradigm, Annals of Mathematics and Artificial Intelligence 25 (1999) 241–273.
[24] R. Rosati, On the decidability and complexity of integrating ontologies and rules, Journal of Web Semantics 3 (2005) 61–73.
[25] R. Rosati, DL+log: Tight integration of description logics and disjunctive datalog, in: Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR2006), AAAI Press, Lake District of the United Kingdom, 2006, pp. 68–78.
[26] Y.-D. Shen, J.-H. You, A generalized gelfond-lifschitz transformation for logic programs with abstract constraints, in: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, AAAI Press, Vancouver, British Columbia, 2007, pp. 483–488.
[27] Y.-D. Shen, J.-H. You, L.Y. Yuan, Characterizations of stable model semantics for logic programs with arbitrary constraint atoms, Theory and Practice of Logic Programming 9 (2009) 529–564.
[28] T.C. Son, E. Pontelli, P.H. Tu, Answer sets for logic programs with arbitrary abstract constraint atoms, Journal of Artificial Intelligence Research 29 (2007) 353–389.
[29] T. Syrjänen, Lparse 1.0 user's manual, 2000, http://www.tcs.hut.fi/Software/smodels/.
[30] A. Van Gelder, K.A. Ross, J.S. Schlipf, The well-founded semantics for general logic programs, Journal of the ACM 38 (1991) 620–650.
[31] W3C OWL Working Group, 2009, OWL 2 Web Ontology Language: Document Overview, http://www.w3.org/TR/owl2-overview/.
[32] Y. Wang, J.-H. You, L. Yuan, Y.-D. Shen, Loop formulas for description logic programs, in: Theory and Practice of Logic Programming, 26th Int'l. Conference on Logic Programming (ICLP'10) Special Issue, 10, 2010, pp. 531–545.
[33] J.-H. You, G. Liu, Loop formulas for logic programs with arbitrary constraint atoms, in: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, AAAI Press, Chicago, Illinois, USA, 2008, pp. 584–589.