Determining Inference Semantics for Disjunctive Logic Programs (Extended Abstract)*

Yi-Dong Shen[1] and **Thomas Eiter**[2]
[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2]Institut of Logic and Computation, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria
ydshen@ios.ac.cn, eiter@kr.tuwien.ac.at

Abstract

[Gelfond and Lifschitz, 1991] introduced simple disjunctive logic programs and defined the answer set semantics called *GL-semantics*. We observed that the requirement of GL-semantics, i.e., an answer set should be a minimal model of the GL-reduct may be too strong and exclude some answer sets that would be reasonably acceptable. To address this, we present a novel and more permissive semantics, called *determining inference semantics*.

1 Introduction

In a seminal paper, [Gelfond and Lifschitz, 1991] introduced simple disjunctive logic programs, where in rule heads the disjunction operator "|" is used to express incomplete information, and defined the answer set semantics (called *GL-semantics*) based on a program transformation (called *GL-reduct*) and the minimal model requirement. Our observations reveal that the requirement of GL-semantics, i.e., an answer set should be a minimal model of rules of the GL-reduct, may sometimes be too strong and exclude some answer sets that would be reasonably acceptable, as illustrated in the following example.

Example 1. Consider the simple disjunctive program.

$$\Pi: \quad a \mid b \qquad (1)$$
$$b \leftarrow a \qquad (2)$$
$$c \leftarrow a \qquad (3)$$
$$c \leftarrow \neg c \qquad (4)$$

Intuitively, rule (1) presents two alternatives for answer set construction, namely a or b, and rules (2) and (3) infer b and c, respectively if a has already been derived. Rule (4) is a constraint stating that there is no answer set that does not contain c. We distinguish between the following two cases. Suppose that we choose a from rule (1); then by rules (2) and (3) we obtain a potential answer set $I_1 = \{a, b, c\}$. I_1 satisfies the constraint (4), so it is a candidate answer set for Π. Alternatively, suppose that we choose b from rule (1). As a is not inferred from rule (1), rules (2) and (3) are not applicable; so rules (1), (2) and (3) together infer a potential answer set $I_2 = \{b\}$. As I_2 does not satisfy the constraint (4), it is not a candidate answer set for Π. Consequently, $I_1 = \{a, b, c\}$ is a minimal candidate answer set and thus we expect it to be an answer set of Π. However, I_1 is not an answer set under GL-semantics because it is not a minimal model of Π.

To address this, we present a more permissive semantics:

(1) We present a general answer set semantics for disjunctive programs, called *determining inference semantics* (*DI-semantics* for short), which interprets the operator | in rule heads differently from the classical connective ∨, and does not require that answer sets should be minimal models. Specifically, we formalize the rule head operator | by introducing a head selection function sel, i.e., for every interpretation I and rule head $H_1 \mid \cdots \mid H_k$, $sel(H_1 \mid \cdots \mid H_k, I)$ nondeterministically selects one alternative H_i satisfied by I. Then we define answer sets as follows: (i) Given an interpretation I and a selection function sel, we transform a disjunctive program Π into a normal program Π^I_{sel}, called *disjunctive program reduct*, such that for every rule $head(r) \leftarrow body(r)$ in Π, $sel(head(r), I) \leftarrow body(r)$ is in Π^I_{sel} if I satisfies $body(r)$; (ii) given a base answer set semantics $\mathcal{X}$ for normal programs, we define I to be a *candidate answer set* w.r.t. $\mathcal{X}$ if I is an answer set of Π^I_{sel} under $\mathcal{X}$; and (iii) we define I to be an answer set w.r.t. $\mathcal{X}$ if I is a minimal candidate answer set. Such answer sets are called *DI-answer sets*.

(2) By replacing the base semantics $\mathcal{X}$ in the above general semantics with the GL_{nlp}-semantics defined by [Gelfond and Lifschitz, 1988], we induce a DI-semantics for simple disjunctive programs (definitions follow below). We show that an answer set under GL-semantics is an answer set under DI-semantics, but not vice versa; the main reason behind is that GL-semantics interprets the operator | in rule heads as the classical connective ∨ and further requires that answer sets must be minimal models; this may exclude some desired answer sets. To clearly see the essential difference of DI-semantics from GL-semantics, we also present a new characterization of GL-semantics in terms of a disjunctive program reduct Π^I_{sel}. Based on this characterization, we obtain a satisfactory solution to an open problem of [Hitzler and Seda, 1999], which was to characterize split normal derivatives of a simple disjunctive program Π.

(3) By replacing the base semantics $\mathcal{X}$ with the well-

*This paper is an extended abstract of the article [Shen and Eiter, 2019] in *Artificial Intelligence*, 277:1-28, 2019.

justified semantics defined by [Shen *et al.*, 2014], we further induce a DI-semantics for general disjunctive programs consisting of rules of the form $H_1 \mid \cdots \mid H_k \leftarrow B$, where B and every H_i are arbitrary first-order formulas. This closes the open issue of [Shen *et al.*, 2014] how to extend the well-justified semantics from general normal programs with rules of the form $H_1 \leftarrow B$ to general disjunctive programs.

(4) Finally, we show that in the propositional case deciding whether a simple disjunctive program Π has some DI-answer set is NP-complete, and deciding whether a ground literal is true in some (resp. every) DI-answer set of Π is Σ_2^p-complete (resp. Π_2^p-complete). This is in contrast to GL-semantics, where deciding whether a simple disjunctive program has GL-answer sets is Σ_2^p-complete [Eiter and Gottlob, 1995]. For general disjunctive programs, the complexity of DI-semantics increases to Σ_2^p-completeness for DI-answer set existence and to Σ_3^p-completeness and Π_3^p-completeness for brave and cautious reasoning, respectively.

For an extensive discussion of historical and philosophical background, we refer to [Shen and Eiter, 2019].

2 Disjunctive Programs

We take a first-order logic language $\mathcal{L}_\Sigma$ with equality. A *first-order theory* (or theory) is a set T of closed formulas. By $\mathcal{N}_\Sigma$ we denote the set of all ground (variable-free) terms of Σ, and by $\mathcal{H}_\Sigma$ the set of all ground atoms. An interpretation I is a subset of $\mathcal{H}_\Sigma$ such that for any ground atom A, I *satisfies* A if $A \in I$, and $\neg A$ if $A \notin I$. The notion of *satisfaction/models* of a formula/theory in I is defined as usual. A theory T *entails* a closed formula F, denoted $T \models F$, if all models of T are models of F. For an interpretation I, we let $I^- = \mathcal{H}_\Sigma \setminus I$ and $\neg I^- = \{\neg A \mid A \in I^-\}$.

Definition 1. A *general disjunctive program* (*disjunctive program* for short) is a finite set of *rules* of the form

$$H_1 \mid \cdots \mid H_k \leftarrow B \quad (1)$$

where $k > 0$, and B and the H_i's are first-order formulas.

For a rule r, we refer to B and $H_1 \mid \cdots \mid H_k$ as its body and head, denoted $body(r)$ and $head(r)$, respectively. We also refer to each H_i as a *head formula*. A *constraint* is a rule of the form $\bot \leftarrow B$. A rule $A \leftarrow \neg A$ amounts to a constraint $\bot \leftarrow \neg A$. A disjunctive program is a *general normal program* (*normal program* for short) if $k = 1$ for every rule; a *simple disjunctive program* if each H_i is an atom and B is a conjunction of literals, and a *simple normal program* if additionally $k = 1$. A *positive simple normal/disjunctive program is a simple normal/disjunctive program* without negative literals. The *grounding* of a disjunctive program Π, obtained by substituting the free variables in Π with constants in all possible ways, is denoted $ground(\Pi)$.

An interpretation I *satisfies* a rule head $H_1 \mid \cdots \mid H_k$ if it satisfies some H_i; I satisfies a rule r if it either satisfies $head(r)$ or it does not satisfy $body(r)$; I is a *model* of a disjunctive program Π if I satisfies every rule $r \in ground(\Pi)$.

Let Π be a simple disjunctive program and I an interpretation. The *GL-reduct* of Π w.r.t. I, written as Π^I, is obtained from $ground(\Pi)$ by (1) removing all rules whose bodies contain some $\neg C_i$ with $C_i \in I$, and (2) removing from the remaining rules all $\neg C_i$. The *GL-semantics* defines I to be an answer set of Π (referred to as *GL-answer set*) if I is a minimal model of Π^I [Gelfond and Lifschitz, 1991]. When Π is a simple normal program, the *GL$_{nlp}$-semantics* defines I to be an answer set of Π if I is the least model of Π^I. For simple normal programs, GL- and GL$_{nlp}$-semantics coincide.

Remark 1. If we replace $\mid$ with $\vee$ in rule heads and let $\Pi_\vee^I$ be Π^I with all occurrences of $\mid$ replaced by $\vee$, then Π^I has the same minimal models as $\Pi_\vee^I$. Thus I is an answer set of Π under GL-semantics iff I is a minimal model of Π^I iff I is a minimal model of $\Pi_\vee^I$. This means that in GL-semantics the use of $\mid$ in rule heads amounts in essence to disjunction $\vee$.

3 Determining Inference (DI) Semantics

Under the constructive view of the operator $\mid$ as a nondeterministic inference operator, every rule head $\mathcal{H} = H_1 \mid \cdots \mid H_k$ in a disjunctive program can be viewed as a set $\{H_1, \cdots, H_k\}$ of alternatives. As these alternatives may have different variants (i.e., every H_i can be expressed as different yet logically equivalent formulas) and appear in different orders in rule heads, we introduce a notion of variant rule heads.

Definition 2. Rule heads $\mathcal{H}_1 = E_1 \mid \cdots \mid E_k$ and $\mathcal{H}_2 = F_1 \mid \cdots \mid F_l$, where the E_i's and F_j's are closed formulas, are *variant rule heads* if for every E_i in $\mathcal{H}_1$ some F_j in $\mathcal{H}_2$ exists with $E_i \equiv F_j$, and vice versa for every F_j in $\mathcal{H}_2$ some E_i in $\mathcal{H}_1$ exists with $E_i \equiv F_j$.

Intuitively, variant rule heads $\mathcal{H}_1$ and $\mathcal{H}_2$ represent the same set of alternatives and should be treated the same. If $k > l$, then $\mathcal{H}_1$ must have some head formulas that are logically equivalent. Moreover, rule heads in a simple disjunctive program are variant rule heads iff they have the same atoms.

Definition 3. Let Π be a disjunctive program and $\mathcal{I}$ the collection of all interpretations. Let $\mathcal{HD}_\Pi$ be the set of all rule heads in $ground(\Pi)$, and $\mathcal{HF}_\Pi$ the set of all head formulas in $\mathcal{HD}_\Pi$. A *head selection* for Π is a function $sel : \mathcal{HD}_\Pi \times \mathcal{I} \rightarrow \mathcal{HF}_\Pi \cup \{\bot\}$ such that for every interpretation $I \in \mathcal{I}$ and every rule $r \in ground(\Pi)$,

$$sel(head(r), I) = \begin{cases} F_i, & \text{if } head(r) \text{ has some head formula} \\ & F_i \text{ that is satisfied by } I \\ \bot, & \text{otherwise,} \end{cases}$$

such that for every variant rule heads $\mathcal{H}_1$ and $\mathcal{H}_2$ in $\mathcal{HD}_\Pi$, $sel(\mathcal{H}_1, I) \equiv sel(\mathcal{H}_2, I)$.

A head selection function sel formalizes the operator $\mid$ as a nondeterministic operator; i.e., for any interpretation I, $sel(F_1 \mid \cdots \mid F_k, I)$ returns from a rule head $F_1 \mid \cdots \mid F_k$ one of the alternatives F_i satisfied by I, or it returns $\bot$ if there is no F_i that is satisfied by I. For variant rule heads, it returns logically equivalent alternatives that are satisfied by I.

Definition 4. Let Π be a disjunctive program, I an interpretation and sel a head selection function. The *reduct* of Π w.r.t. I and sel is $\Pi_{sel}^I = \{sel(head(r), I) \leftarrow body(r) \mid r \in ground(\Pi)$ s.t. I satisfies $body(r)\}$.

A reduct Π_{sel}^I is a normal program; therefore we can apply any existing answer set semantics for normal programs to compute answer sets of Π_{sel}^I. Intuitively I is a candidate answer set of Π if I is an answer set of Π_{sel}^I, and I is an answer set of Π if I is minimal among all candidate answer sets.

Definition 5. Let I be a model of a disjunctive program Π, and $\mathcal{X}$ be an answer set semantics for normal programs. Then I is an answer set of Π w.r.t. $\mathcal{X}$ if (1) for some head selection function sel, I is an answer set of Π^I_{sel} under $\mathcal{X}$, and (2) Π has no model $J \subset I$ satisfying condition (1).

Due to the use of head selection functions, the above semantics interprets the disjunctive rule head operator | differently from the classical connective $\vee$. Let $\mathcal{H}_1 = E_1 \mid \cdots \mid E_k$ and $\mathcal{H}_2 = E_1 \vee \cdots \vee E_k$ be two rule heads and let I be an interpretation that satisfies $\mathcal{H}_2$. Then there may be up to k head selection functions for $\mathcal{H}_1$, each selecting one alternative E_i that is satisfied by I, which leads to at most k disjunctive program reducts; in contrast, there is only one head selection function for $\mathcal{H}_2$, i.e., $sel(\mathcal{H}_2, I) = \mathcal{H}_2$, which leads to only one disjunctive program reduct. Different reducts may lead to different candidate answer sets and thus disjunctive programs with rule heads like $\mathcal{H}_1$ are different from programs with rule heads like $\mathcal{H}_2$.

Moreover, the above semantics does not require that answer sets should be minimal models; it only requires answer sets to be minimal among all candidate answer sets.

In order to stress the intuition that candidate answer sets are determined by means of a chosen head selection function for applying rules $H_1 \mid \cdots \mid H_k \leftarrow Body$, where one alternative H_i from the head is inferred when $Body$ is satisfied, we refer to the above answer set semantics as *determining inference (DI) semantics* for disjunctive programs; we call answer sets of DI-semantics ***DI-answer sets*** and models satisfying condition (1) of Definition 5 *candidate DI-answer sets*.

4 DI-Semantics for Simple Disjunctive Programs

By replacing the base semantics $\mathcal{X}$ in Definition 5 with GL_{nlp}-semantics we induce a DI-answer set semantics for simple disjunctive programs.

Definition 6. A model I of a simple disjunctive program Π is a *DI-answer set* of Π, if (1) for some head selection function sel, I is an answer set of Π^I_{sel} under GL_{nlp}-semantics, and (2) Π has no model $J \subset I$ satisfying condition (1).

A DI-answer set is not necessarily a GL-answer set, but for simple normal programs and positive simple disjunctive programs, DI-semantics agrees with GL-semantics.

Theorem 1. *Let Π be a simple normal program or a positive simple disjunctive program. Then an interpretation I is a DI-answer set of Π iff I is a GL-answer set of Π.*

It is particularly interesting to observe that GL-semantics can also be characterized using the disjunctive program reduct Π^I_{sel} of Definition 4 simply by requiring that for *every* (instead of *some*) head selection function sel, I is an answer set of Π^I_{sel} under GL_{nlp}-semantics. *This reveals the essential difference between DI-semantics and GL-semantics.*

Theorem 2. *A model I of a simple disjunctive program Π is a GL-answer set of Π iff for every head selection function sel, I is an answer set of Π^I_{sel} under GL_{nlp}-semantics.*

As GL-answer sets of a simple disjunctive program Π are minimal models of Π, the following corollary is immediate.

Corollary 1. *Let Π be a simple disjunctive program. If I is a GL-answer set, then I is a DI-answer set.*

5 DI-Semantics for General Programs

General normal programs consist of rules of the form $H \leftarrow B$, where H and B are first-order formulas. To overcome the problem of circular justifications with those answer set semantics for general normal programs such as those in [Pearce, 2006; Truszczynski, 2010; Bartholomew *et al.*, 2011; Faber *et al.*, 2011; Ferraris *et al.*, 2011] based on classical logic, [Shen *et al.*, 2014] presented the *well-justified semantics* whose answer sets have a level mapping and thus are free of circular justifications, in analogy to the level mapping of GL_{nlp}-semantics for simple normal programs [Fages, 1994]. [Shen *et al.*, 2014] left extending the well-justified semantics to general disjunctive programs as an open problem; we can elegantly close it by replacing the base semantics $\mathcal{X}$ in Definition 5 with the well-justified semantics.

The well-justified semantics is based on the one-step provability operator $T_\Pi(O, N)$, which extends the well-known immediate consequence operator [van Emden and Kowalski, 1976] from Horn programs to general normal programs.

Definition 7 ([Shen *et al.*, 2014])**.** Let Π be a general normal program, and let O and N be two first-order theories. Then

$$T_\Pi(O, N) = \{head(r) \mid r \in ground(\Pi), O \cup N \models body(r)\}.$$

Informally, $T_\Pi(O, N)$ collects all heads of grounded rules whose bodies are entailed by $O \cup N$. For fixed N, the entailment $\models$ is monotone in O, so $T_\Pi(O, N)$ is monotone w.r.t. O, i.e., for any theories $O_1 \subseteq O_2$, we have $T_\Pi(O_1, N) \subseteq T_\Pi(O_2, N)$. As moreover $T_\Pi(O, N)$ is finitary, the inference sequence $\langle T^i_\Pi(\emptyset, N)\rangle^\infty_{i=0}$, where $T^0_\Pi(\emptyset, N) = \emptyset$ and for $i \geq 0$ $T^{i+1}_\Pi(\emptyset, N) = T_\Pi(T^i_\Pi(\emptyset, N), N)$, will converge to a least fixpoint, denoted $lfp(T_\Pi(\emptyset, N))$.

The *well-justified (WJ) semantics* is then defined in terms of $lfp(T_\Pi(\emptyset, \neg I^-))$, i.e., derivability under the closed-world assumption applied to candidate answer I, as follows.

Definition 8 ([Shen *et al.*, 2014])**.** Let I be a model of a general normal program Π. Then I is a *WJ-answer set* of Π if $lfp(T_\Pi(\emptyset, \neg I^-)) \cup \neg I^- \models A$ for every $A \in I$.

By replacing $\mathcal{X}$ in Definition 5 with WJ-semantics we induce a DI-answer set semantics for general programs.

Definition 9. A model I of a general disjunctive program Π is a *DI-answer set* of Π if (1) for some head selection function sel, I is a WJ-answer set of Π^I_{sel}, and (2) Π has no model $J \subset I$ satisfying condition (1).

Intuitively, a DI-answer set is a model that is minimal among all models that can be nondeterministically (by means of a head selection function) inferred by iteratively applying rules via a bottom up fixpoint sequence.

Corollary 2. *For a general normal program, I is a DI-answer set iff I is a WJ-answer set. For a simple disjunctive program, I is a DI-answer set under Definition 9 iff I is a DI-answer set under Definition 6.*

6 Computational Complexity

We address the computational complexity of propositional logic programs, where we focus on the DI-semantics with the well-justified semantics [Shen *et al.*, 2014] as the base semantics and refer to it as *DI-WJ answer set semantics*.

Theorem 3. *Given a propositional simple (resp. general) disjunctive program* Π *and a ground literal L, deciding whether (i)* Π *has some DI-WJ answer set is* NP*-complete (resp.* Σ_2^p*-complete), (ii) L is true in every DI-WJ answer set of* Π *is* Π_2^p*-complete (resp.* Π_3^p*-complete), and (iii) L is true in some DI-WJ answer set of* Π *is* Σ_2^p*-complete (resp.* Σ_3^p*-complete).*

Analogous results hold for other semantics such as FLP-semantics [Faber *et al.*, 2011]. Compared to GL-/FLP-semantics, the complexity of brave and cautious reasoning increases under DI-semantics by one level of PH, thus offering higher problem solving capacity. Computing a DI-WJ answer set is complete for the NP- resp. Σ_2^p-functions and feasible with bounded many witness oracle calls [Buss *et al.*, 1993; Janota and Marques-Silva, 2016] in polynomial time.

7 Difference between Disjunctive Rule Heads and Choice Constructs

Like disjunctive rule heads, *choice constructs* [Simons *et al.*, 2002; Ferraris and Lifschitz, 2005; Calimeri *et al.*, 2012] are also used to express a set of alternatives. However, a disjunctive rule head $a_1 \mid \cdots \mid a_m$ and a choice construct of the form $u_1\{a_1, \cdots, a_m\}u_2$, where $m > 0$, $0 \le u_1 \le u_2 \le m$, and the a_i's are ground atoms, are essentially different.

Let $\alpha = \{a_1, \cdots, a_m\}$ and $\beta = \{\gamma \mid \gamma \subseteq \alpha \text{ and } u_1 \le |\gamma| \le u_2\}$. The choice construct $u_1\{a_1, \cdots, a_m\}u_2$ says that any $\gamma \in \beta$ can be chosen as answer.

For a logic program Π, let $AS(\Pi)$ denote the set of answer sets of Π. Let Π' be Π extended with a choice construct $u_1\{a_1, \cdots, a_m\}u_2$. Then the set of answer sets of Π' is

$$AS(\Pi') = \bigcup_{\gamma \in \beta} AS(\Pi \cup \{a \mid a \in \gamma\} \cup \{\neg b \mid b \in (\alpha \setminus \gamma)\}).$$

Example 2. Let $\Pi = \{b\}$ and $\Pi' = \Pi \cup \{1\{a, b\}2\}$. Then $AS(\Pi') = AS(\Pi \cup \{a, \neg b\}) \cup AS(\Pi \cup \{\neg a, b\}) \cup AS(\Pi \cup \{a, b\})$. $\Pi \cup \{a, \neg b\}$ has no model and thus no answer set, $\Pi \cup \{\neg a, b\}$ has a single answer set $\{b\}$, and $\Pi \cup \{a, b\}$ has a single answer set $\{a, b\}$. Therefore, Π' has in total two answer sets, $\{b\}$ and $\{a, b\}$.

A disjunctive rule head $a_1 \mid \cdots \mid a_m$ infers one atom a_i from α and differs essentially from a choice construct $1\{a_1, \cdots, a_m\}u$. When $u = 1$, the choice construct $1\{a_1, \cdots, a_m\}1$ in a logic program Π enforces every answer set of Π to contain exactly one a_i from α. In contrast, though $a_1 \mid \cdots \mid a_m$ infers only one a_i from α, a DI-answer set may contain other atoms $a_j \in \alpha$, which are inferred by other rules in a disjunctive program. When $u > 1$, the choice construct $1\{a_1, \cdots, a_m\}u$ allows for answer sets I and J with $I \subset J$. This will not happen with $a_1 \mid \cdots \mid a_m$ for DI-answer sets.

8 Relation to Split and Fork Programs

For simple disjunctive programs Π, [Hitzler and Seda, 1999] proposed to split Π into a collection of simple normal programs, called *normal derivatives* $P(\Pi)$, which informally are obtained from $ground(\Pi)$ by replacing every rule $A_1 | \cdots | A_k \leftarrow body(r)$, $k \ge 2$, arbitrarily with one or more rules $A_i \leftarrow body(r)$, $1 \le i \le k$. E.g., $\Pi = \{p \mid q \leftarrow \neg s\}$ has three normal derivatives: $P_1(\Pi) = \{p \leftarrow \neg s\}$, $P_2(\Pi) = \{q \leftarrow \neg s\}$, and $P_3(\Pi) = \{p \leftarrow \neg s, q \leftarrow \neg s\}$.

[Hitzler and Seda, 1999] aimed to use normal derivatives to characterize GL-semantics of a simple disjunctive program Π. They showed that every answer set of Π under GL-semantics is an answer set of some normal derivative of Π under GL_{nlp}-semantics. E.g. for the program Π from above, $I = \{p\}$ is an answer set of Π under GL-semantics and an answer set of $P_1(\Pi)$ under GL_{nlp}-semantics. However, they left a precise characterization open, stated as the problem to determine for every interpretation I some normal derivatives such that I is an answer set of Π under GL-semantics iff I is an answer set of these normal derivatives under GL_{nlp}-semantics.

The characterization of GL-semantics by the disjunctive program reduct (Theorem 2) enables us to provide a solution for this open problem. For a logic program Π, an interpretation I and a head selection sel on I, let

$$P_{sel}(\Pi, I) = \{sel(head(r), I) \leftarrow body(r) \mid r \in ground(\Pi)\},$$
$$ND(\Pi, I) = \{P_{sel}(\Pi, I) \mid sel \text{ is a head selection on } I\}.$$

Note that $ND(\Pi, I)$ is the collection of normal derivatives obtained by applying every head selection on I. Thus for any head selection sel on a model I, we have

$$\begin{aligned} P_{sel}(\Pi, I) &= \{sel(head(r), I) \leftarrow body(r) \mid r \in ground(\Pi)\} \\ &= \Pi_{sel}^I \cup \{sel(head(r), I) \leftarrow body(r) \mid r \in ground(\Pi) \\ &\qquad \text{and } body(r) \text{ is not satisfied by } I\}. \end{aligned}$$

A solution to the above open problem is then as follows.

Theorem 4. *An interpretation I is an answer set of a simple disjunctive program* Π *under GL-semantics iff I is an answer set of every* $P(\Pi) \in ND(\Pi, I)$ *under* GL_{nlp}*-semantics.*

In independent work and parallel to ours, [Aguado *et al.*, 2019] proposed a new construct "|" for answer programs called *fork*, which aims at overcoming problems with omitting auxiliary atoms in choice constructs. Informally, under fork semantics the answer sets of $\{E \mid F\} \cup \Pi$ are the answer sets of $\{E\} \cup \Pi$ plus the answer sets of $\{F\} \cup \Pi$. Accordingly, $P = \{a \mid b, b \mid c\}$ has the fork-answer sets $\{a, b\}$, $\{a, c\}$, $\{b\}$ and $\{b, c\}$, while its DI-answer sets are $\{a, c\}$ and $\{b\}$. They diverge as DI-semantics operates in a sense globally on alternatives in different rules (by item (2) in Definition 5), while fork semantics operates locally treating them independently.

Notably, selection functions similar to ours were used in [Vennekens *et al.*, 2004] to define probabilistic semantics for logic programs with annotated disjunctions. However, they do not depend on an interpretation and result (disregarding probabilities) in all normal derivatives picking always a single rule $A_i \leftarrow body$ (in the example, $P_1(\Pi)$ and $P_2(\Pi)$). Thus like fork semantics, this semantics has a local flavor.

Acknowledgments

This work has been supported in part by NSFC grants 61976205 and 61379043, and the Austrian Science Fund (FWF) grant P27730.

References

[Aguado *et al.*, 2019] Felicidad Aguado, Pedro Cabalar, Jorge Fandinno, David Pearce, Gilberto Pérez, and Concepción Vidal. Forgetting auxiliary atoms in forks. *Artif. Intell.*, 275:575–601, 2019.

[Bartholomew *et al.*, 2011] M. Bartholomew, J. Lee, and Y. Meng. First-order extension of the FLP stable model semantics via modified circumscription. In *Proc. 22nd Int'l Joint Conference on Artificial Intelligence (IJCAI-11)*, pages 724–730, 2011.

[Buss *et al.*, 1993] Samuel Buss, Jan Krajìček, and Gaisi Takeuti. On provably total functions in bounded arithmetic theories. In Peter Clote and Jan Krajìček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 116–61. Oxford University Press, 1993.

[Calimeri *et al.*, 2012] Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Francesco Ricca, and Torsten Schaub. ASP-Core-2: Input language format, 2012. https://www.mat.unical.it/aspcomp2013/files/ASP-CORE-2.01c.pdf.

[Eiter and Gottlob, 1995] T. Eiter and G. Gottlob. On the computational cost of disjunctive logic programming: Propositional case. *Annals of Mathematics and Artificial Intelligence*, 15(3-4):289–323, 1995.

[Faber *et al.*, 2011] W. Faber, G. Pfeifer, and N. Leone. Semantics and complexity of recursive aggregates in answer set programming. *Artificial Intelligence*, 175(1):278–298, 2011.

[Fages, 1994] François Fages. Consistency of clark's completion and existence of stable models. *Journal of Methods of Logic in Computer Science*, 1:51–60, 1994.

[Ferraris and Lifschitz, 2005] Paolo Ferraris and Vladimir Lifschitz. Mathematical foundations of answer set programming. In Sergei N. Artëmov, Howard Barringer, Artur S. d'Avila Garcez, Luís C. Lamb, and John Woods, editors, *We Will Show Them! Essays in Honour of Dov Gabbay, Volume One*, pages 615–664. College Publications, 2005.

[Ferraris *et al.*, 2011] P. Ferraris, J. Lee, and V. Lifschitz. Stable models and circumscription. *Artificial Intelligence*, 175(1):236–263, 2011.

[Gelfond and Lifschitz, 1988] Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium*, pages 1070–1080, 1988.

[Gelfond and Lifschitz, 1991] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.

[Hitzler and Seda, 1999] Pascal Hitzler and Anthony Karel Seda. Multivalued mappings, fixed-point theorems and disjunctive databases. In *Proceedings of the 3rd Irish Conference on Formal Methods, Galway, Eire*, pages 113–131. British Computer Society, 1999.

[Janota and Marques-Silva, 2016] Mikolás Janota and Joao Marques-Silva. On the query complexity of selecting minimal sets for monotone predicates. *Artif. Intell.*, 233:73–83, 2016.

[Pearce, 2006] D. Pearce. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence*, 47(1-2):3–41, 2006.

[Shen and Eiter, 2019] Y. D. Shen and T. Eiter. Determining inference semantics for disjunctive logic programs. *Artificial Intelligence*, 277:1–28, 2019.

[Shen *et al.*, 2014] Y. D. Shen, K. Wang, T. Eiter, M. Fink, C. Redl, T. Krennwallner, and J. Deng. FLP answer set semantics without circular justifications for general logic programs. *Artificial Intelligence*, 213:1–41, 2014.

[Simons *et al.*, 2002] P. Simons, I. Niemela, and T. Soininen. Extending and implementing the stable model semantics. *Artificial Intelligence*, 138(1-2):181–234, 2002.

[Truszczynski, 2010] M. Truszczynski. Reducts of propositional theories, satisfiability relations, and generalizations of semantics of logic programs. *Artificial Intelligence*, 174(16-17):1285–1306, 2010.

[van Emden and Kowalski, 1976] M. H. van Emden and R. A. Kowalski. The semantics of predicate logic as a programming language. *Journal of the ACM*, 23(4):733–742, 1976.

[Vennekens *et al.*, 2004] Joost Vennekens, Sofie Verbaeten, and Maurice Bruynooghe. Logic programs with annotated disjunctions. In Bart Demoen and Vladimir Lifschitz, editors, *Logic Programming, 20th International Conference, ICLP 2004, Saint-Malo, France, September 6-10, 2004, Proceedings*, volume 3132 of *Lecture Notes in Computer Science*, pages 431–445. Springer, 2004.