Paracoherent Answer Set Programming*

Thomas Eiter and **Michael Fink**
Institut für Informationssysteme
Technische Universität Wien, Austria
{eiter,fink}@kr.tuwien.ac.at

João Moura
Centro de Inteligência Artificial
Universidade Nova de Lisboa, Portugal
joaomoura@yahoo.com

Abstract

We study the problem of reasoning from incoherent answer set programs, i.e., from logic programs that do not have an answer set due to cyclic dependencies of an atom from its default negation. As a starting point we consider so-called semi-stable models which have been developed for this purpose building on a program transformation, called epistemic transformation. We give a model-theoretic characterization of this semantics, considering pairs of two-valued interpretations of the original program, rather than resorting to its epistemic transformation. Moreover, we show some anomalies of semi-stable semantics with respect to basic epistemic properties and propose an alternative semantics satisfying these properties. In addition to a model-theoretic and a transformational characterization of the alternative semantics, we prove precise complexity results for main reasoning tasks under both semantics.

Introduction

Answer Set Programming (ASP) is a prime formalism for nonmonotonic reasoning and knowledge representation, mainly because of the existence of efficient solvers and well-established relationships to common nonmonotonic logics. It is a declarative programming paradigm with a model-theoretic semantics, where problems are encoded into a logic program using rules, and its models, called answer sets (or stable models) (Gelfond and Lifschitz 1991), encode solutions. However, due to nonmonotonicity, programs may be incoherent, i.e., lack an answer set due to cyclic dependencies of an atom from its default negation. Nonetheless, there are many cases when this is not intended and one might want to draw conclusions also from an incoherent program, e.g., for debugging purposes, or in order to keep a system (partially) responsive in exceptional situations. This is akin to the principle of paraconsistency, where non-trivial consequences shall be derivable from an inconsistent theory. As so-called extended logic programs also may be inconsistent in the classical sense, i.e., they may have the inconsistent answer set as their unique answer set, we use the term *paracoherent reasoning* to distinguish between paraconsistent reasoning and reasoning from incoherent programs.

Both types of reasoning from answer set programs have been studied in the course of the development of the answer set semantics; for approaches on paraconsistent ASP cf., e.g., Sakama and Inoue (1995), Alcântara, Damásio, and Pereira (2004), Odintsov and Pearce (2005)). Numerous semantics for logic programs with nonmonotonic negation can be considered as a paracoherent semantics for ASP. Ideally, such a semantics satisfies the following properties:

(1) Every (consistent) answer set of a program corresponds to a model (*answer set coverage*).

(2) If a (consistent) answer set exists for a program, then all models correspond to an answer set (*congruence*).

(3) If a program has a classical model, then it has a model (*classical coherence*).

Widely-known semantics, such as 3-valued stable models (Przymusinski 1991), L-stable models (Eiter, Leone, and Saccà 1997), revised stable models (Pereira and Pinto 2005), regular models (You and Yuan 1994), and pstable models (Osorio, Ramírez, and Carballido 2008), satisfy only part of these requirements (see the Discussion section for further semantics and more details). Semi-stable models (Sakama and Inoue 1995) however, satisfy all three properties and thus are the prevailing paracoherent semantics.

Despite the model-theoretic nature of ASP, semi-stable models have been defined by means of a program transformation, called epistemic transformation. A semantic characterization in the style of equilibrium models for answer sets (Pearce and Valverde 2008) is still missing. We address this problem and make the following main contributions.

- We characterize semi-stable models by pairs of 2-valued interpretations of the original program, similar to so-called here-and-there (HT) models. In that, we point out some anomalies of the semi-stable semantics with respect to basic rationality properties in modal logics (**K** and **N**), that essentially prohibit a 1-to-1 characterization in terms of HT-models.

- This leads us to propose an alternative paracoherent semantics, called semi-equilibrium semantics, which satisfies the aforementioned properties and is characterized using

*This work was partially supported by the Vienna Science and Technology Fund (WWTF) grant ICT 08-020 and the Austrian Science Fund (FWF) grant P20841.

HT-models. Moreover, semi-equilibrium models can be obtained by selecting answer sets of an extension of the epistemic transformation (applying the same criteria as for semi-stable models).

• We study major reasoning tasks under both semantics and provide precise characterizations of their computational complexity for normal programs as well as for disjunctive programs. Besides brave and cautious reasoning, deciding whether a program has a model, respectively recognizing models, is considered under the given semantics.

Our results contribute to a more logical foundation of paracoherent answer set programming, which gains increasing importance in inconsistency management.

Preliminaries

We consider a propositional setting; extensions to non-ground logic programs are straightforward (more details are given in the Discussion section below, though).

A (*disjunctive*) rule r is of the form

$$a_1 \vee \cdots \vee a_l \leftarrow b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n, \qquad (1)$$

where $a_1, \ldots, a_l, b_1, \ldots, b_n$ are atoms of a propositional signature $\mathcal{L}$, such that $l \geq 0$, $n \geq m \geq 0$, and $l + n > 0$. We refer to "$\neg$" as *default negation*. The *head* of r is the set $H(r) = \{a_1, \ldots, a_l\}$, and the *body* of r is denoted by $B(r) = \{b_1, \ldots, b_m, \neg b_{m+1}, \ldots, \neg b_n\}$. Furthermore, we define the sets $B^+(r) = \{b_1, \ldots, b_m\}$, and $B^-(r) = \{b_{m+1}, \ldots, b_n\}$. A rule r of form (1) is called (i) a *fact*, if $m = 0$ and $l = 1$ (here, $\leftarrow$ is usually omitted), (ii) a *constraint*, if $n = 0$, (iii) *normal*, if $l \leq 1$, and (iv) *positive*, if $m = n$. A *program* P (over $\mathcal{L}$) is a set of rules (over $\mathcal{L}$).

An interpretation I, i.e., a set of atoms over $\mathcal{L}$, satisfies a rule r, denoted $I \models r$, iff $I \cap H(r) \neq \emptyset$ if $B^+(r) \subseteq I$ and $B^-(r) \cap I = \emptyset$. The *Gelfond-Lifschitz reduct* (Gelfond and Lifschitz 1991) of a program P with respect to an interpretation I, denoted P^I, is given by the set of rules

$$a_1 \vee \cdots \vee a_l \ \leftarrow \ b_1, \ldots, b_m,$$

obtained from rules in P, such that $B^-(r) \cap I = \emptyset$.

An interpretation I is an *answer set* of P, iff $I \models P^I$ and it is subset minimal among the interpretations of $\mathcal{L}$ with this property; $\mathcal{AS}(P)$ denotes the set of all answer sets of P.

The *logic of here-and-there* (HT) (Pearce and Valverde 2008) serves as a valuable basis for characterizing semantic properties of ASP. It is an intermediate logic between intuitionistic and classical logic, considering formulas built over a propositional signature $\mathcal{L}$ and the connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\bot$. We restrict our attention to formulas of the form

$$b_1 \wedge \ldots \wedge b_m \wedge \neg b_{m+1} \wedge \ldots \wedge \neg b_n \rightarrow a_1 \vee \cdots \vee a_l, \qquad (2)$$

corresponding to rules of the form (1). Like intuitionistic logic HT can be semantically characterized by Kripke models, in particular using just two worlds, namely "*here*" and "*there*" (assuming that the *here* world is ordered before the *there* world). Accordingly, interpretations (HT-interpretations) are pairs (X, Y) of sets of atoms from $\mathcal{L}$, such that $X \subseteq Y$. An HT-interpretation is *total* if $X = Y$. Intuitively, atoms in X (the *here* part) are considered to be true, atoms not in Y (the *there* part) to be false, while the remaining atoms (from $Y \setminus X$) are undefined.

We denote classical satisfaction of a formula ϕ by an interpretation X, i.e., a set of atoms, as $X \models \phi$, whereas satisfaction in the logic of here-and-there (an HT-model), denoted $(X, Y) \models \phi$, is defined recursively:

1. $(X, Y) \models a$ if $a \in X$, for any atom a,
2. $(X, Y) \not\models \bot$,
3. $(X, Y) \models \neg\phi$ if $Y \not\models \phi$[1],
4. $(X, Y) \models \phi \wedge \psi$ if $(X, Y) \models \phi$ and $(X, Y) \models \psi$,
5. $(X, Y) \models \phi \vee \psi$ if $(X, Y) \models \phi$ or $(X, Y) \models \psi$,
6. $(X, Y) \models \phi \rightarrow \psi$ if (i) $(X, Y) \not\models \phi$ or $(X, Y) \models \psi$, and (ii) $Y \models \phi \rightarrow \psi$.

An HT-interpretation (X, Y) satisfies a theory Γ, iff it satisfies all formulas $\phi \in \Gamma$.

A total HT-interpretation (Y, Y) is called an *equilibrium model* of a theory Γ, iff $(Y, Y) \models \Gamma$ and for all HT-interpretations (X, Y), such that $X \subset Y$, it holds that $(X, Y) \not\models \Gamma$. An interpretation Y is an *answer set* of Γ iff (Y, Y) is an equilibrium model of Γ. For further details see, e.g., (Pearce and Valverde 2008).

Semi-Stable Models

Sakama and Inoue (1995) introduced *semi-stable models* as an extension of paraconsistent answer set semantics (called PAS semantics, respectively p-stable models by them) for extended disjunctive logic programs. Their aim was to provide a framework which is paraconsistent for incoherence, i.e., in situations where stability fails due to cyclic dependencies of a literal from its default negation.

Since we are primarily interested in paracoherence, in the following summary and study of semi-stable semantics, we disregard aspects devoted to paraconsistency, more specifically, we exclude strong negation. Note also that (Sakama and Inoue 1995) allowed for programs with variables, while we focus on the propositional case. These restrictions help to put their technique to handle incoherence in perspective. Moreover, our results easily carry over to the original setting considering PAS semantics and allowing for non-ground programs. This will be considered in more detail in the Discussion section below.

We consider an extended propositional language $\mathcal{L}^\kappa = \mathcal{L} \cup \{Ka \mid a \in \mathcal{L}\}$. Intuitively, Ka can be read as a is believed to hold. Semantically, we resort to subsets of $\mathcal{L}^\kappa$ as interpretations I^κ and the truth values false $\bot$[2], believed true $\mathbf{bt}$, and true $\mathbf{t}$, where $\bot \preceq \mathbf{bt} \preceq \mathbf{t}$. The truth value assigned by I^κ to a propositional variable a is defined by

$$I^\kappa(a) = \begin{cases} \mathbf{t} & \text{if } a \in I^\kappa, \\ \mathbf{bt} & \text{if } Ka \in I^\kappa \text{ and } a \notin I^\kappa, \\ \bot & \text{otherwise.} \end{cases}$$

[1] That is, Y satisfies $\neg\phi$ classically.

[2] In (Sakama and Inoue 1995) $\bot$ is called 'undefined', as it should be if strong negation is considered as well.

The semi-stable models of a program P are defined via its *epistemic transformation* P^κ.

Definition 1 (**P^κ (Sakama and Inoue 1995)**) *Let P be a disjunctive program. Then its epistemic transformation is defined as the positive disjunctive program P^κ obtained from P by replacing each rule r of the form (1) in P, such that $B^-(r) \neq \emptyset$, with:*

$$\lambda_{r,1} \vee \ldots \vee \lambda_{r,l} \vee Kb_{m+1} \vee \ldots \vee Kb_n \leftarrow b_1, \ldots, b_m, \quad (3)$$
$$a_i \leftarrow \lambda_{r,i}, \quad (4)$$
$$\leftarrow \lambda_{r,i}, b_j, \quad (5)$$
$$\lambda_{r,i} \leftarrow a_i, \lambda_{r,k}, \quad (6)$$

for $1 \leq i \leq l$, $m+1 \leq j \leq n$, and $1 \leq k \leq l$.

Note that for any program P, its epistemic transformation P^κ is positive. Models of P^κ are defined in terms of a fixpoint operator in (Sakama and Inoue 1995), with the property that for positive programs, according to Theorem 2.9, minimal fixpoints coincide with minimal models of the program. Therefore, for any program P, minimal fixpoints of P^κ coincide with answer sets of P^κ.

Semi-stable models are then defined as *maximal canonical* interpretations among the minimal fixpoints (answer sets) of P^κ: Given an interpretation I^κ over $\mathcal{L}' \supseteq \mathcal{L}^\kappa$, let $gap(I^\kappa) = \{Ka \mid Ka \in I^\kappa \text{ and } a \notin I^\kappa\}$. Given a set S of interpretations over $\mathcal{L}'$, an interpretation $I^\kappa \in S$ is maximal canonical in S iff there is no interpretation $J^\kappa \in S$ such that $gap(I^\kappa) \subset gap(J^\kappa)$. Let $mc(S)$ denote maximal canonical interpretations in S and let $\mathcal{SST}(P)$ be the semi-stable models of a program P, then we can equivalently paraphrase the definition of semi-stable models in (Sakama and Inoue 1995) as follows.

Definition 2 *Let P be a program over $\mathcal{L}$. Then, $\mathcal{SST}(P) = \{I^\kappa \cap \mathcal{L}^\kappa \mid I^\kappa \in mc(\mathcal{AS}(P^\kappa))\}$.*

Semantic Characterization

As opposed to its transformational definition, in this work we aim at a model-theoretic characterization of semi-stable models in the line of model-theoretic characterizations of the answer set semantics by means of HT.

Example 1 *Let $P = \{a \leftarrow \neg a\}$. The program is incoherent, with $\{Ka\}$ as its unique semi-stable model. Its HT-models are $(\emptyset, \{a\})$ and $(\{a\}, \{a\})$. One might aim characterizing the semi-stable model by $(\emptyset, \{a\})$.*

However, resorting to HT-interpretations will not uniquely characterize semi-stable models as illustrated next.

Example 2 *Consider the program*

$$P = \{a;\ b;\ c;\ d \leftarrow \neg a, \neg b;\ d \leftarrow \neg b, \neg c\}.$$

It is coherent, with a single answer set $\{a, b, c\}$, while $\mathcal{SST}(P) = \{\{a, b, c, Kb\}, \{a, b, c, Ka, Kc\}\}$. Note that neither $(\{a, b, c\}, \{b\})$ nor $(\{a, b, c\}, \{a, c\})$ is a HT-interpretation.

Hence, for a 1-to-1 characterization we have to resort to different structures. Sticking to the requirement that, given a program P over $\mathcal{L}$, pairs of two-valued interpretations over $\mathcal{L}$ should serve as the underlying semantic structures, we say that a bi-interpetation of a program P over $\mathcal{L}$ is any pair (I, J) of interpretations over $\mathcal{L}$, and define:

Definition 3 *Let ϕ be a formula over $\mathcal{L}$, and let (I, J) be a bi-interpretation over $\mathcal{L}$. Then, (I, J) is a* bi-model *of ϕ, denoted $(I, J) \models_\beta \phi$ iff*

1. *$(I, J) \models_\beta a$ if $a \in I$, for any atom a,*
2. *$(I, J) \not\models_\beta \bot$,*
3. *$(I, J) \models_\beta \neg\phi$ if $J \not\models \phi$,*
4. *$(I, J) \models_\beta \phi \wedge \psi$ if $(I, J) \models_\beta \phi$ and $(I, J) \models_\beta \psi$,*
5. *$(I, J) \models_\beta \phi \vee \psi$ if $(I, J) \models_\beta \phi$ or $(I, J) \models_\beta \psi$,*
6. *$(I, J) \models_\beta \phi \rightarrow \psi$ if (i) $(I, J) \not\models_\beta \phi$, or (ii) $(I, J) \models_\beta \psi$ and $I \models \phi$.*

Moreover, (I, J) is a bi-model of a program P, iff $(I, J) \models_\beta \phi$, for all ϕ of the form (2) corresponding to a rule $r \in P$.

In case of programs, its bi-models can alternatively be characterized by the following condition on its rules.

Proposition 1 *Let r be a rule over $\mathcal{L}$, and let (I, J) be a bi-interpretation over $\mathcal{L}$. Then, $(I, J) \models_\beta r$ if and only if*

(a) *$B^+(r) \subseteq I$ and $J \cap B^-(r) = \emptyset$ implies $I \cap H(r) \neq \emptyset$ and $I \cap B^-(r) = \emptyset$.*

To every bi-model of a program P, we associate a corresponding interpretation $(I, J)^\kappa$ over $\mathcal{L}^\kappa$ by $(I, J)^\kappa = I \cup \{Ka \mid a \in J\}$. Conversely, given an interpretation I^κ over $\mathcal{L}^\kappa$ its associated bi-interpretation $\beta(I^\kappa)$ is given by $(I^\kappa \cap \mathcal{L}, \{a \mid Ka \in I^\kappa\})$.

In order to relate these constructions to models of the epistemic transformation, which builds on additional atoms of the form $\lambda_{r,i}$, we construct an interpretation $(I, J)^{\kappa,P}$ of P^κ from a given bi-interpretation (I, J) of P as:

$$(I, J)^{\kappa,P} = (I, J)^\kappa \cup \{\lambda_{r,i} \mid r \in P, B^-(r) \neq \emptyset, a_i \in I, I \models B(r), J \models B^-(r)\},$$

where r is of the form (1).

Proposition 2 *Let P be a program over $\mathcal{L}$. Then,*

1. *if (I, J) is a bi-model of P, then $(I, J)^{\kappa,P} \models P^\kappa$;*
2. *if $M \models P^\kappa$ then $\beta(M \cap \mathcal{L}^\kappa)$ is a bi-model of P.*

Based on bi-models, a 1-1 characterization of semi-stable models succeeds imposing suitable minimality criteria.

Theorem 1 *Let P be a program over $\mathcal{L}$. Then,*

(1) *if (I, J) is a bi-model of P such that*
  (i) *$(I', J) \not\models_\beta P$, for all $I' \subset I$,*
  (ii) *$(I, J') \not\models_\beta P$, for all $J' \subset J$, and*
  (iii) *there is no bi-model (I', J') of P that satisfies (i) and $J' \setminus I' \subset J \setminus I$,*
  then $(I, J)^\kappa \in \mathcal{SST}(P)$;

(2) *if $I^\kappa \in \mathcal{SST}(P)$, then $\beta(I^\kappa)$ is a bi-model of P that satisfies (i)-(iii).*

Intuitively, Conditions (i) and (ii) filter bi-models that uniquely correspond to (some but not all) answer sets of P^κ: due to minimality every answer set satisfies (i); there may be answer sets of P^κ that do not satisfy (ii), but they are certainly not maximal canonical. Eventually, Condition (iii) ensures that maximal cononical answer sets are selected. More formally, the proof of this theorem builds on the following relationship between bi-models of P and answer sets of P^κ.

Corollary 1 *Let P be a program over $\mathcal{L}$. If $M \in \mathcal{AS}(P^\kappa)$, then $\beta(M \cap \mathcal{L}^\kappa)$ satisfies (i). If (I, J) is a bi-model of P that satisfies (i) and (ii), then there exists $M \in \mathcal{AS}(P^\kappa)$, such that $\beta(M \cap \mathcal{L}^\kappa) = (I, J)$.*

For illustration consider the following example.

Example 3 *Let $P = \{a \leftarrow b;\ b \leftarrow \neg b\}$. Its bi-models are all pairs (I, J), where $I \in \{\emptyset, \{a\}, \{a, b\}\}$ and $J \in \{\{b\}, \{a, b\}\}$. Condition (i) of Theorem 1 holds for bi-models such that $I = \emptyset$, and Condition (ii) holds only-if $J = \{b\}$. Thus, $\{Kb\}$ is the unique semi-stable model of P.*

The examples given so far also exhibit some anomalies of the semi-stable semantics with respect to basic rationality properties considered in epistemic logics. In particular, *knowledge generalization* (or *necessitation*, resp. modal axiom **N**) is a basic principle in respective modal logics. For a semi-stable model I^κ, it would require that

Property N: $a \in I^\kappa$ implies $Ka \in I^\kappa$, for all $a \in \mathcal{L}$.

This property does not hold as witnessed by Example 2.

Another basic requirement is the *distribution axiom* (modal axiom **K**). Assuming that we belief the rules of a given program (which might also be seen as the consequence of adopting knowledge generalization) the distribution property can be paraphrased for a rule of the form (1) as follows:

Property K: If $I^\kappa \models Kb_1 \wedge \ldots \wedge Kb_m$ and $I^\kappa \not\models Kb_{m+1} \vee \ldots \vee Kb_n$, then $I^\kappa \models Ka_1 \vee \ldots \vee Ka_l$.

Note that this does not hold for rule $a \leftarrow b$ in Example 3.

An Alternative Paracoherent Semantics

In this section we define and characterize an alternative paracoherent semantics which we call semi-equilibrium semantics (for reasons which will become clear immediately). The aim for semi-equilibrium models is to enforce Properties **N** and **K** on them. Let us start considering bi-models of a program P, that satisfy these properties. It turns out that such structures are exactly given by HT-models.

Proposition 3 *Let P be a program over $\mathcal{L}$. Then,*

(1) if (I, J) is a bi-model of P, such that $(I, J)^\kappa$ satisfies Property **N** *and Property* **K***, for all $r \in P$, then (I, J) is an HT-model of P;*

(2) if (H, T) is an HT-model of P, then $(H, T)^\kappa$ satisfies Property **N** *and Property* **K***, for all $r \in P$.*

In order to define semi-equilibrium models, we follow the basic idea of the semi-stable semantics and select subset minimal models that are maximal canonical. Let us define $HT^\kappa(P) = \{(H, T)^\kappa \mid (H, T) \models P\}$ and denote by $\mathcal{MM}(HT^\kappa(P))$ its minimal elements with respect to subset inclusion.

Definition 4 *Let P be a program over $\mathcal{L}$. An interpretation I^κ over $\mathcal{L}^\kappa$ is a semi-equilibrium model of P iff $I^\kappa \in mc(\mathcal{MM}(HT^\kappa(P)))$. The set of semi-equilibrium models of P is denoted by $\mathcal{SEQ}(P)$.*

A model-theoretic characterization for this semantics is obtained as before, replacing bi-models by HT-models and dropping Condition (ii). Intuitively, Condition (ii) is not needed as it is subsumed by Condition (iii) (i.e., Condition (ii′) below) if Property **N** and Condition (i) hold.

Theorem 2 *Let P be a program over $\mathcal{L}$. Then,*

(1) If (H, T) is an HT-model of P such that
  (i′) $(H', T) \not\models P$, for all $H' \subset H$, and
  (ii′) there is no HT-model (H', T') of P that satisfies (i′) and $T' \setminus H' \subset T \setminus H$,
  then $(H, T)^\kappa \in \mathcal{SEQ}(P)$;

(2) if $I^\kappa \in \mathcal{SEQ}(P)$, then $\beta(I^\kappa)$ is an HT-model of P that satisfies (i′) and (ii′).

Alternatively, semi-equilibrium models may be computed as maximal canonical answer sets, i.e., equilibrium models, of an extension of the epistemic program transformation.

Definition 5 (P^{HT}) *Let P be a program over $\mathcal{L}$. Then its epistemic HT-transformation P^{HT} is defined as the union of P^κ with the set of rules:*

$$Ka \leftarrow a,$$
$$Ka_1 \vee \ldots \vee Ka_l \vee Kb_{m+1} \vee \ldots \vee Kb_n \leftarrow Kb_1, \ldots, Kb_m,$$

for $a \in \mathcal{L}$, respectively for every rule $r \in P$ of the form (1).

The extensions of the transformation naturally ensure Properties **N** and **K** on its models and its maximal canonical answer sets coincide with semi-equilibrium models.

Theorem 3 *Let P be a program over $\mathcal{L}$, and let I^κ be an interpretation over $\mathcal{L}^\kappa$. Then, $I^\kappa \in \mathcal{SEQ}(P)$ if and only-if $I^\kappa \in \{M \cap \mathcal{L}^\kappa \mid M \in mc(\mathcal{AS}(P^{HT}))\}$.*

The resulting semantics is classically coherent.

Proposition 4 *Let P be a program over $\mathcal{L}$. If P has a model, then it has a semi-equilibrium model.*

Another simple property is a 1-to-1 correspondence between answer sets and semi-equilibrium models.

Proposition 5 *Let P be a coherent program over $\mathcal{L}$. Then,*

(1) if $Y \in \mathcal{AS}(P)$, then $(Y, Y)^\kappa$ is a semi-equilibrium model of P;

(2) if I^κ is a semi-equilibrium model of P then $\beta(I^\kappa)$ is an equilibrium model of P, i.e., $\beta(I^\kappa)$ is of the form (Y, Y) and $Y \in \mathcal{AS}(P)$.

For an illustration of the 1-to-1 relationship between answer sets and semi-equilibrium models, let us reconsider Example 2. Note that this example also gave evidence that semi-stable models do not satisfy Property **N**, which is the case for semi-equilibrium models, however.

Example 4 *Consider the coherent program of Example 2. Its unique semi-equilibrium model is $\{a, b, c, Ka, Kb, Kc\}$, corresponding to the single answer set $\{a, b, c\}$.*

Problem	normal P	disjunctive P
$\mathcal{SST}(P) \neq \emptyset$?	NP	NP
$I^\kappa \in \mathcal{SST}(P)$?	coNP	Π_2^P
$P \models_b^v a$	Σ_2^P	Σ_3^P
$P \models_c^v a$	Π_2^P	Π_3^P

Table 1: Complexity of semi-stable models (completeness results). The same results hold for semi-equilibrium models.

As a consequence of Property **K**, semi-equilibrium semantics differs from semi-stable semantics not only with respect to believed consequences.

Example 5 *Consider the following extension of the program in Example 3:* $P = \{a \leftarrow b;\ b \leftarrow \neg b;\ c \leftarrow \neg a\}$, *and compare* $\mathcal{SST}(P) = \{\{c, Kb\}\}$ *with* $\mathcal{SEQ}(P) = \{\{Ka, Kb\}\}$ *concerning the knowledge with respect to* c.

Computational Complexity

We now consider the computational complexity of major reasoning tasks for programs under semi-stable semantics:

a) deciding whether a program P has a semi-stable model,

b) recognizing semi-stable models (given M and P), and

c) brave and cautious reasoning from the semi-stable models of a program P, where an atom a is a brave (resp., cautious) consequence of P with value $v \in \{\bot, \mathbf{bt}, \mathbf{t}\}$, denoted $P \models_b^v a$ (resp., $P \models_c^v a$) if $I^\kappa(a) = v$ for some (resp., every) semi-stable model of P.

We also consider these problems for semi-equilibrium models. For brevity, we compactly summarize the results in Table 1. In the following, we explain how the results are derived. The following is easy to see.

Lemma 1 *Given a bi-interpretation* (I, J) *of a program* P, *deciding* $(I, J) \models_\beta P$ *is polynomial.*

Problem a)

Since P has some semi-stable model iff P has some classical model, the complexity of Problem a) is immediate.

Theorem 4 *Given a disjunctive program* P, *deciding whether* $\mathcal{SST}(P) \neq \emptyset$ *is* NP*-complete. The* NP*-hardness holds already for normal programs* P.

However, the problem is trivial, if the program P has no constraints, and is polynomial e.g. if P consists of Horn clauses.

Problem b)

Recognizing semi-stable models, howwever, is more complext than recognizing classical models.

Theorem 5 *Given an interpretation* I^κ *over* $\mathcal{L}^\kappa$ *and a program* P, *deciding if* $I^\kappa \in \mathcal{SST}(P)$ *is*

(i) coNP*-complete for normal* P, *and*

(ii) Π_2^P*-complete for disjunctive* P.

This result can be derived as follows. Given an interpretation I^κ over $\mathcal{L}^\kappa$, we can verify whether it is a semi-stable model of P by checking the conditions (i)–(iii) in Theorem 1 for $\beta(I^\kappa) = (I, J)$. By the Lemma 1, both (i) and (ii) are feasible in coNP, and testing (iii) is then in Π_2^P; as Π_2^P is closed under conjunction, this shows that Problem b) is in Π_2^P for general programs.

In case of normal programs, both (i) and (ii) are checkable in polynomial time. Indeed, condition (a) in Def. 3 equals

(a′) $B^+(r) \subseteq I$ and $J \cap B^-(r) = \emptyset$ implies $I \cap H(r) \neq \emptyset$,

(b) $B^+(r) \subseteq I$ and $J \cap B^-(r) = \emptyset$ implies $I \cap B^-(r) = \emptyset$.

As for (i), some $I' \subset I$ such that $(I', J) \models_\beta P$ (if one exists) can be found as the least model of a rephrasing of (a') and (b) into Horn clauses for all rules r, omitting those r where $J \not\models B^-(r)$. Similarly, in (ii) some $J' \subset J$ such that $(I, J') \models_\beta P$ (if one exists) can be found as any minimal model of the positive clauses $\bigvee B^-(r)$ for all r such that $I \models B^+(r)$ and either $I \not\models H(r)$ or $I \not\models B^-(r)$. In both cases, computing I' resp. J' is polynomial.

The Π_2^P-hardness is shown via a reduction from deciding, given a disjunctive program P and an atom p, whether p occurs in no answer set of P; wlog, P is a positive program P_0 that has a minimal model $\{q\}$. We then let $P_1 = P_0 \cup \{\leftarrow \neg p\}$. It is easily seen that $\{q, Kp\}$ is a semi-stable model of P_1 iff P_0 has no stable model in which p occurs.

The coNP-hardness for normal programs is shown by a reduction from UNSAT. Let $E = C_1 \wedge \cdots \wedge C_m$ be a CNF over atoms $x_1, \ldots, x_n$. Define a program P_2 with the rules

1. $full \leftarrow \neg sat$
2. $\leftarrow full, \neg a_i,$
   $\leftarrow full, \neg \bar{a}_i;$
3. $x_i \leftarrow \neg a_i,$
   $\bar{x}_i \leftarrow \neg \bar{a}_i;$
4. $full \leftarrow x_i, \bar{x}_i;$
5. $c_j \leftarrow l^*_{j_k};$
6. $sat \leftarrow c_1, \ldots, c_m;$

where $i = 1, \ldots, n$, $j = 1, \ldots m$, and $k = 1, \ldots, s_j$ given that $C_j = l_{j_1} \vee \ldots \vee l_{j_{s_j}}$, and $l^*_{j_k} = l_{j_k}$, if l_{j_k} is positive and $l^*_{j_k} = \bar{l}_{j_k}$ otherwise.

Then,

$$I_0 = \{full, Ka_1, K\bar{a}_1, \ldots, Ka_n, K\bar{a}_n\}$$

is such that

$$\beta(I_o) = (\{full\}, \{a_1, \bar{a}_1, \ldots, a_n, \bar{a}_n\})$$

is a bi-model of P that satisfies conditions (i) and (ii) of Theorem 1. Morever, $\beta(I_o)$ satisfies condition (iii) (and thus I_0 is semi-stable) iff E is unsatisfiable. Briefly, any (I', J') as in (iii) must be such that $full \notin I'$, which means $sat \in J'$ by the first rule; as $J' \setminus I' \subseteq J \setminus I$, this means that also $sat \in I'$. By condition (i) for (I', J'), this means that I' encodes a set of literals (chosen by the rules 3.) that satisfies each clause in E; the rule 4. ensures consistency.

We note that in this construction, the number of atoms under negation is, different from the case of disjunctive programs, not bounded by a constant. In the latter case, for normal programs Problem b) is polynomial. This is because only constantly many J matter, and for each J the unique minimal I' such that $(I', J) \models P$ (if $(I, J) \models P$ for some I) is computable in polynomial time.

Problem c)

Brave resp. cautious reasoning from the semi-stable models of programs is one level higher up in the Polynomial Hierarchy than respective reasoning from the stable models. Intuitively, this is because maximal canonicality is an orthogonal selection on top of the stable models.

Theorem 6 *Given a program P, an atom a, and a value $v \in \{\bot, \mathbf{bt}, \mathbf{t}\}$, deciding whether*

(i) $P \models_b^v a$ is Σ_2^P-complete for normal P and Σ_3^P-complete for disjunctive P;

(ii) $P \models_c^v a$ is Π_2^P-complete for normal P and Π_3^P-complete for disjunctive P.

The membership of brave reasoning in Σ_3^P for disjunctive programs (resp. in Σ_2^P for normal programs) is a simple consequence of the result for Problem b), and similarly the membership of cautious reasoning in Π_3^P (resp. Π_2^P).

The Σ_3^P/Π_3^P-hardness for disjunctive programs can be shown by lifting the reduction for model recognition. The program P_0 used for P_1 evaluates wlog a $\forall\exists$-QBF Φ, cf. (Eiter and Gottlob 1995) such that p is in no minimal model of P iff Φ is true. The construction of P_1 from P_0 works if $\Phi = \Phi(Z)$ contains free variables $Z = z_1, \ldots, z_l$ that are set by facts z_i (make z_i true) and $\bar{z}_i$ (make z_i false). A choice from $z_i, \bar{z}_i$ (emulating such facts) can be encoded with rules

$$z_i \vee \bar{z}_i \leftarrow, \quad \leftarrow z_i, \neg b_i, \quad \leftarrow \bar{z}_i, \neg \bar{b}_i.$$

which ensure that every bi-model (I, J) of P that satisfies conditions (i) and (ii) of Theorem 1 is such that either $z_i \in I, b_i \in J$ or $\bar{z}_i \in I, \bar{b}_i \in J$. The resulting program P_3 has some semi-stable model containing q (recall that P_0 used for P_1 has a minimal model $\{q\}$) iff $\exists Z \Phi(Z)$ is true. Furthermore, p belongs to all semi-stable models of P_3 iff $\exists Z \Phi(Z)$ is false.

The Σ_2^P/Π_2^P-hardness for normal programs is similarly shown by lifting the reduction from UNSAT (i.e., a $\forall$-QBF $\Phi = \forall X E$) in Problem b). However, we can't simply add disjunctive clauses $z_i \vee \bar{z}_i$ to set additional free variables $Z = z_1, \ldots, z_l$ in $\Phi = \Phi(Z)$. To this end, we use rules

1. $z_i \leftarrow \neg b_i$,
   $\bar{z}_i \leftarrow \neg \bar{b}_i$;
2. $\leftarrow z_i, \bar{z}_i$,
3. $ok_i \leftarrow z_i$,
   $ok_i \leftarrow \bar{z}_i$;
4. $ok \leftarrow ok_1, \ldots, ok_l$;

those in 1. and 2. effect that every bi-model (I, J) of P that satisfies conditions (i) and (ii) of Theorem 1 is such that either (a) $z_i \in I, \bar{b}_i \in J$, (b) $\bar{z}_i \in I, b_i \in J$, or (c) $b_i \in I, \bar{b}_i \in J$; (a) and (b) correspond to a proper choice, which is noticed with ok_i; if all choices are proper, then ok is true. In the program for model recognition P_2, we add ok in each rule body; then only complete choices trigger the respective rules. The resulting program P_4 has then some semi-stable model containing $full$ iff the formula $\exists Z \forall X E(X, Z)$ is true. Furthermore, if we add a rule $p \leftarrow \neg full$, then p belongs to all semi-stable models of the resulting program P_5 iff $\exists Z \forall X E(X, Z)$ is false.

Semi-Equilibrium Models

For disjunctive programs, Problems a)-c) have for semi-equilibrium models the same complexity as for semi-stable models; we omit stating here the formal results.

The memberships proofs are similar but use Theorem 2 instead of Theorem 1; note that deciding $(H, T) \models P$ is polynomial, and thus checking condition (i$'$) in Theorem 2 is in coNP. Furthermore, similar hardness proofs work (with slight adaptions).

Also for normal programs P, we get analog complexity results, since testing condition (i$'$) for such P is polynomial: similarly as for condition (i) in Theorem 1, we can compute some $H' \subset H$ such that $(H', T) \models P$ (if one exists) as the least model of a set of Horn clauses (more precisely of P^T).

In all cases, we also have matching hardness results. This follows easily from well-known results for answer sets semantics of positive disjunctive programs, cf. (Eiter and Gottlob 1995), which contain wolog no facts; we can easily emulate such programs under semi-equilibrium semantics, by just shifting disjunctions to the rule bodies, and creating constraints in this way.

Discussion

In this section, we first review some general principles for logic programs with negation, and we then consider the relationship of semi-stable and semi-equilibrium semantics to other semantics of logic programs with negation. Finally, we address some possible extensions of our work.

General Principles

In the context of logic programs with negation, several principles have been identified which a semantics desirably should satisfy. Among them are:

- the *principle of minimal undefinedness* (You and Yuan 1994), which says that a smallest set of atoms should be undefined (i.e., neither true nor false);
- the *principle of justifiability (or foundedness)* (You and Yuan 1994): every atom which is true must be derived from the rules of the program, possibly using negative literals as additional axioms.
- the *principle of the closed world assumption (CWA)*, for models of disjunctive logic programs (Eiter et al. 1997): "If every rule with an atom a in the head has a false body, or its head contains a true atom distinct from A w.r.t. an acceptable model, then a must be false in that model."

It can be shown that both the semi-stable and the semi-equilibrium semantics satisfy the first two principles (properly rephrased and viewing $\mathbf{bt}$ as undefined), but not the CWA principle; this is shown by the simple program $P = \{a \leftarrow \neg a\}$ and the acceptable model $\{Ka\}$. However, this is due to the particular feature of making, as in this example, assumptions about the truth of atoms; if the CWA condition is restricted to atoms a that are not believed by assumption, i.e., $I^\kappa(a) \neq \mathbf{bt}$ in a semi-stable resp. semi-equilibrium model I^κ, then the CWA property holds.

We eventually remark that Property **N** can be enforced on semi-stable models by adding constraints $\leftarrow a, \neg a$ for all atoms a to the (original) program. However, enforcing Property **K** on semi-stable models is more involved and efficient encodings seem to require an extended signature.

Related Semantics

P-stable (partial stable) models, which coincide with the 3-valued stable models of (Przymusinski 1991), are one of the best known approximation of answer sets. Recently, the P-stable models have been semantically characterized by Cabalar et al. (2007) in the logic HT^2 in terms of partial equilibrium models.

The L-stable models semantics by Eiter, Leone, and Saccà (1997), which selects those P-stable models where a smallest set of atoms is undefined, is closest in spirit to semi-stable and semi-equilibrium semantics; furthermore, it satisfies the three principles from above, as well as answer set coverage and congruence (cf. Introduction). The main difference is that L-stable—like P-stable—semantics takes a neutral position on undefinedness, which in combination with negation may lead to weaker conclusions.

For example, the program in Example 5 has a single P-stable (and L-stable) model in which all atoms are undefined, while c is true under semi-equlibrium semantics. Similarly, the program

$$P = \{a \leftarrow \neg b;\ b \leftarrow \neg c;\ c \leftarrow \neg a\}$$

has a single P-stable (and thus L-stable) model in which all atoms are undefined, while it has multiple semi-stable models, viz. $\{a, Ka, Kc\}$, $\{b, Kb, Ka\}$, and $\{c, Kc, Kb\}$, which coincide with the semi-equilibrium models. If we add the rules $d \leftarrow a$, $d \leftarrow b$, and $d \leftarrow c$ to P, the new program cautiously entails under both semi-stable and semi-equilibrium model semantics that d is true, but not under L-stable semantics.

Furthermore, disjunctive programs may lack L-stable models, e.g.

$$P' = P \cup \{\, a \vee b \vee c \leftarrow \};$$

the semi-stable resp. semi-equilibrium models of P' are those of P.

Opposite to the L-stable semantics is the least P-stable model semantics, which selects the P-stable model in which a largest set of atoms is undefined; for normal logic programs, a unique such model always exists, and this model coincides with the well-founded model of van Gelder, Ross, and Schlipf (1991); furthermore, it is characterized in the logic HT^2 in terms of the minimal partial equilibrium model (under a suitable ordering) (Cabalar et al. 2007). As in the examples for L-stable semantics above, the normal programs have a single P-stable model, the least P-stable and the L-stable semantics for these programs coincide, showing thus similar differences to the semi-stable and semi-equilibrium semantics.

Regular semantics (You and Yuan 1994) is another 3-valued approximation of answer set semantics that satisfies least undefinedness and foundedness, but not the CWA principle. However, it is classically coherent. For the program P above, the regular models coincide with the L-stable models; the program P' has the regular models $\{a\}$, $\{b\}$, and $\{c\}$. While regular models fulfill answer set coverage, they do not fulfill congruence. For more discussion of 3-valued stable and regular models as well as many other semantics coinciding with them, see (Eiter, Leone, and Saccà 1997).

Revised stable models (Pereira and Pinto 2005) are a 2-valued approximation of answer sets; negated literals are assumed to be maximally true, where assumptions are revised if they would lead to self-incoherence through odd loops or infinite proof chains. For example, the odd-loop program P above has three revised stable models, viz. $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$. The semantics is only defined for normal logic programs, and fullfills answer set coverage but not congruence, cf. (Pereira and Pinto 2005). Similarly, pstable models (Osorio, Ramírez, and Carballido 2008), which have a definition for disjunctive programs however, satisfy answer set coverage (but just for normal programs) and congruence fails. Moreover, every pstable model of a program is a minimal model of the program, but there are programs, e.g. P above, that have models but no pstable model, thus classical coherence does not hold.

Extensions

As already mentioned, semi-stable semantics has originally been developed as an extension to p-minimal model semantics (Sakama and Inoue 1995), a paraconsistent semantics for extended disjunctive logic programs, i.e., programs which besides default negation also allow for strong (classical) negation. A declarative characterization of p-minimal models by means of frames was given by Alcântara, Damásio, and Pereira (2004), who coined the term *Paraconsistent Answer-set Semantics* (PAS) for it. This characterization has been further simplified and underpinned with a logical axiomatization in (Odintsov and Pearce 2005) by using Routley models, i.e., a simpler possible worlds model.

Our characterizations for both, semi-stable models and semi-equilibrium models, can be easily extended to this setting if they are applied to semantic structures which are given by 4-tuples of interpretations rather than bi-interpretations, respectively to Routley here-and-there models rather than HT-models. Intuitively, this again amounts to considering two 'worlds', each of which consists of a pair of interpretations: one for positive literals (atoms), and one for negative literals (strongly negated atoms). The respective epistemic transformations are unaffected except for the fact that literals are considered rather than atoms. One can also show for both semantics that there is a simple 1-to-1 correspondence to the semi-stable (semi-equilibrium) models of a transformed logic program without strong negation: A given extended program P is translated into a program P' over $\mathcal{L} \cup \{a' \mid a \in \mathcal{L}\}$ without strong negation by replacing each negative literal of the form $-a$ by a'. If (I, J) is a semi-stable (semi-equilibrium) model of P', then

$$(I \cap \mathcal{L}, \{-a \mid a' \in I\}, J \cap \mathcal{L}, \{-a \mid a' \in J\})$$

is a semi-stable (semi-equilibrium) model of P. Note that semi-stable (semi-equilibrium) models of extended logic programs obtained in this way generalize the PAS semantics, which means that they are paraconsistent as well as paracoherent. Logically this amounts to distinguishing nine truth values rather than three, with the additional truth values *undefined*, *believed false*, *believed inconsistent*, *true with contradictory belief*, *false with contradictory belief*, and *inconsistent*. The computational complexity for extended programs is the same.

Compared to (Sakama and Inoue 1995), we further restrict here to propositional programs, as opposed to programs with variables (non-ground programs). However, respective semantics for non-ground programs via their grounding are straightforward. Alternatively, in case of semi-equilibrium models one can simply replace HT-models by Herbrand models of quantified equilibrium logic (Pearce and Valverde 2008). Similarly for the other semantics, replacing interpretations in the semantic structures by Herbrand interpretations over a given function-free first-order signature, yields a characterization of the respective models.

Conclusion

We have given a semantic characterization of semi-stable models in terms of bi-models, and of semi-equilibrium models, which eliminate some anomalies of semi-stable models, in terms of HT-models. Furthermore, we characterized the complexity of major reasoning tasks of these semantics.

Regarding implementation, we developed experimental prototypes for computing $\mathcal{SST}(P)$ and $\mathcal{SEQ}(P)$ based on these characterizations. They construct the bi-models (resp., HT-models) of P and filter them according to the conditions in Theorem 1 (resp., Theorem 2). Alternatively, $\mathcal{SST}(P)$ and $\mathcal{SEQ}(P)$ are obtainable by postprocessing the answer sets of the epistemic transformation P^κ resp. its extension P^{HT}, which are computed with any ASP solver.

Concerning future work, there are several issues. In this paper, we have considered paracoherence based on program transformation, as introduced by Sakama and Inoue (1995). Other notions, like forward chaining construction and strong compatibility (Wang, Zhang, and You 2009; Marek, Nerode, and Remmel 1999) might be other candidates to deal with paracoherent reasoning in logic programs; it remains to explore this.

Another subject is to extend paracoherence to language extensions, including aggregates, nested logic programs etc. Of particular interest are here modular logic programs (Janhunen et al. 2009; Dao-Tran et al. 2009), where module interaction may lead to incoherence. Related to the latter are the more general multi-context systems (Brewka and Eiter 2007), in which knowledge bases exchange beliefs via non-monotonic bridge rules; based on ideas and results of this paper, paracoherent semantics for certain classes of such multi-context systems may be devised.

Finally, another issue is to investigate the use of paracoherent semantics in AI applications such as diagnosis, where assumptions may be exploited to generate candidate diagnoses, in the vein of the generalised stable model semantics (Kakas and Mancarella 1990).

Appendix: Proofs

We provide selected proofs of the results, omitting some details.

Proof of Proposition 1. Let r be a rule over $\mathcal{L}$, and let (I, J) be a bi-interpretation over $\mathcal{L}$.

($\Leftarrow$) Suppose that (I, J) satisfies (a), i.e., $B^+(r) \subseteq I$ and $J \cap B^-(r) = \emptyset$ implies $I \cap H(r) \neq \emptyset$ and $I \cap B^-(r) = \emptyset$. We prove that $(I, J) \models_\beta r$, considering three cases:

Case 1: Assume that $B^+(r) \not\subseteq I$. Then $(I, J) \not\models_\beta a$, for some atom $a \in B^+(r)$, and thus $(I, J) \not\models_\beta B(r)$ which implies $(I, J) \models_\beta r$.

Case 2: Assume that $J \cap B^-(r) \neq \emptyset$, Then $(I, J) \not\models_\beta \neg a$, for some atom $a \in B^-(r)$, and thus $(I, J) \not\models_\beta B(r)$ which implies $(I, J) \models_\beta r$.

Case 3: Assume that $B^+(r) \subseteq I$ and $J \cap B^-(r) = \emptyset$. Then, since (I, J) satisfies (a), it also holds that $I \cap H(r) \neq \emptyset$ and $I \cap B^-(r) = \emptyset$. From $B^+(r) \subseteq I$ and $I \cap B^-(r) = \emptyset$, we conclude that $I \models B(r)$. Moreover, $I \cap H(r) \neq \emptyset$ implies $(I, J) \models_\beta H(r)$. Thus, $(I, J) \models_\beta r$.

By our assumption, one of these three cases applies for (I, J), proving the claim.

($\Rightarrow$) Suppose that $(I, J) \models_\beta r$. We prove that (I, J) satisfies (a), distinguishing two cases:

Case 1: Assume that $(I, J) \not\models_\beta B(r)$. Then either $(I, J) \not\models_\beta a$, for some atom $a \in B^+(r)$, or $(I, J) \not\models_\beta \neg a$, for some atom $a \in B^-(r)$. Hence, $B^+(r) \not\subseteq I$ or $J \cap B^-(r) \neq \emptyset$, which implies that (I, J) satisfies (a).

Case 2: Assume that $(I, J) \models_\beta H(r)$ and $I \models B(r)$. Then $I \cap H(r) \neq \emptyset$ and $I \cap B^-(r) = \emptyset$, and thus (I, J) satisfies (a).

By our assumption, one of the two cases applies for (I, J), which proves the claim. □

Proof of Proposition 2. Let P be a program over $\mathcal{L}$.

Part (1). First, let (I, J) be a bi-model of P. We prove that $(I, J)^{\kappa,P} \models P^\kappa$.

Towards a contradiction assume the contrary. Then there exists a rule r' in P^κ, such that $(I, J)^{\kappa,P} \not\models r'$. Suppose that r' is not transformed, i.e., $r' \in P$ and $B^-(r') = \emptyset$. Since $(I, J) \models_\beta r'$, by Proposition 1 we conclude that $B^+(r') \subseteq I$ implies $I \cap H(r') \neq \emptyset$ (recall that $B^-(r') = \emptyset$). By construction $(I, J)^{\kappa,P}$ restricted to $\mathcal{L}$ coincides with I. Therefore, $B^+(r') \subseteq (I, J)^{\kappa,P}$ implies $(I, J)^{\kappa,P} \cap H(r') \neq \emptyset$, i.e., $(I, J)^{\kappa,P} \models r'$, a contradiction.

Next, suppose that r' is obtained by the epistemic transformation of a corresponding rule $r \in P$ of the form (1), and consider the following cases:

– r' is of the form (3): then $\{b_1, \ldots, b_m\} \subseteq (I,J)^{\kappa,P}$, which implies $B^+(r) \subseteq I$. Moreover, $H(r') \cap (I,J)^{\kappa,P} = \emptyset$ by the assumption that $(I,J)^{\kappa,P} \not\models r'$. By construction of $(I,J)^{\kappa,P}$, this implies $J \cap B^-(r) = \emptyset$. Since $(I,J) \models_\beta r$, we also conclude that $I \cap H(r) \neq \emptyset$ and that $I \cap B^-(r) = \emptyset$. Consequently, $J \models B^-(r)$, $a_i \in I$ for some $a_i \in H(r)$, and $I \models B(r)$. Note also, that $B^-(r) \neq \emptyset$ by definition of the epistemic transformation. According to the construction of $(I,J)^{\kappa,P}$, it follows that $\lambda_{r,i} \in (I,J)^{\kappa,P}$, a contradiction to $H(r') \cap (I,J)^{\kappa,P} = \emptyset$.

– r' is of the form (4): in this case, $(I,J)^{\kappa,P} \not\models r'$ implies $\lambda_{r,i} \in (I,J)^{\kappa,P}$ and $a_i \notin (I,J)^{\kappa,P}$. However, by construction $\lambda_{r,i} \in (I,J)^{\kappa,P}$ implies $a_i \in I$; from the latter, again by construction, we conclude $a_i \in (I,J)^{\kappa,P}$, a contradiction.

– r' is of the form (5): in this case, $(I,J)^{\kappa,P} \not\models r'$ implies $\lambda_{r,i} \in (I,J)^{\kappa,P}$ and $b_j \in (I,J)^{\kappa,P}$. Note that $b_j \in (I,J)^{\kappa,P}$ iff $b_j \in I$. A consequence of the latter is that $I \not\models B(r)$, contradicting a requirement for $\lambda_{r,i} \in (I,J)^{\kappa,P}$ (cf. the construction of $(I,J)^{\kappa,P}$).

– r' is of the form (6): by the assumption that $(I,J)^{\kappa,P} \not\models r'$, it holds that $\lambda_{r,k} \in (I,J)^{\kappa,P}$ and $a_i \in (I,J)^{\kappa,P}$, but $\lambda_{r,i} \notin (I,J)^{\kappa,P}$. From the latter we conclude, by the construction of $(I,J)^{\kappa,P}$, that $a_i \notin I$, since all other requirements for the inclusion of $\lambda_{r,i}$ (i.e., $r \in P$, $B^-(r) \neq \emptyset$, $I \models B(r)$, and $J \models B^-(r)$) must be satisfied as witnessed by $\lambda_{r,k} \in (I,J)^{\kappa,P}$. However, if $a_i \notin I$, then $a_i \notin (I,J)^{\kappa,P}$ (again by construction), contradiction.

This concludes the proof of the fact that if (I,J) is a bi-model of P, then $(I,J)^{\kappa,P} \models P^\kappa$.

Part (2). Let M be a model of P^κ. We prove that $\beta(M \cap \mathcal{L}^\kappa) = (I,J)$ is a bi-model of P. Note that by construction $I = M \cap \mathcal{L}$ and $J = \{a \mid Ka \in M\}$. First, we consider any rule r in P such that $B^-(r) = \emptyset$. Then $r \in P^\kappa$, $J \cap B^-(r) = \emptyset$ and $I \cap B^-(r) = \emptyset$. Hence, by Proposition 1, we need to show that $B^+(r) \subseteq (M \cap \mathcal{L})$ implies $(M \cap \mathcal{L}) \cap H(r) \neq \emptyset$. Since $r \in P^\kappa$, this follows from the assumption, i.e., $M \models P^\kappa$ implies $M \models r$, and therefore if $B^+(r) \subseteq M$, then $M \cap H(r) \neq \emptyset$. Since r is over $\mathcal{L}$, this proves the claim for all $r \in P$ such that $B^-(r) = \emptyset$.

It remains to show that $(I,J) \models_\beta r$ for all $r \in P$ such that $B^-(r) \neq \emptyset$. Towards a contradiction assume that this is not the case, i.e., (i) $B^+(r) \subseteq (M \cap \mathcal{L})$, (ii) $J \cap B^-(r) = \emptyset$, and either (iii) $(M \cap \mathcal{L}) \cap H(r) = \emptyset$ or (iv) $(M \cap \mathcal{L}) \cap B^-(r) \neq \emptyset$ hold for some $r \in P$ of the form (1), such that $B^-(r) \neq \emptyset$. Conditions (i) and (ii), together with $M \models P^\kappa$, imply that $\lambda_{r,i}$ is in M, for some $1 \leq i \leq l$ (cf. the rule of the form (3) in the epistemic transformation of r). Consequently, a_i is in M (cf. the corresponding rule of the form (4) in the epistemic transformation of r), and hence $a_i \in (M \cap \mathcal{L})$. This rules out (iii), so (iv) must hold, i.e., $b_j \in (M \cap \mathcal{L})$, for some $m+1 \leq j \leq n$. But then, M satisfies the body of a constraint in P^κ (cf. the corresponding rule of the form (5) in the epistemic transformation of r), contradicting $M \models P^\kappa$. This proves that there exists no $r \in P$ such that $B^-(r) \neq \emptyset$ and $(I,J) \not\models_\beta r$, and thus concludes our proof of $(I,J) \models_\beta r$. Since $r \in P$ was arbitrary, it follows that $\beta(M \cap \mathcal{L}^\kappa)$ is a bi-model of P. □

Proof of Theorem 1. Let P be a program over $\mathcal{L}$. The proof uses the following lemmas.

Lemma 2 *If $M \in \mathcal{AS}(P^\kappa)$, then $\beta(M \cap \mathcal{L}^\kappa)$ satisfies (i).*

Lemma 3 *If (I,J) is a bi-model of P that satisfies (i) and (ii), then there exists some $M \in \mathcal{AS}(P^\kappa)$, such that $\beta(M \cap \mathcal{L}^\kappa) = (I,J)$.*

Part (1). Let (I,J) be a bi-model of P that satisfies (i)-(iii). We prove that $(I,J)^\kappa \in \mathcal{SST}(P)$. By Lemma 3, we conclude that there exists some $M \in \mathcal{AS}(P^\kappa)$ such that $\beta(M \cap \mathcal{L}^\kappa) = (I,J)$. It remains to show that M is maximal canonical. Towards a contradiction assume the contrary. Then, there exists $M' \in \mathcal{AS}(P^\kappa)$ such that $gap(M') \subset gap(M)$. Let $(I',J') = \beta(M' \cap \mathcal{L}^\kappa)$. By Lemma 2, (I',J') satisfies (i), and by construction since $gap(M') \subset gap(M)$, it holds that $J' \setminus I' \subset J \setminus I$. However, this contradicts the assumption that (I,J) satisfies (iii). Therefore, M is maximal canonical, and hence $(I,J)^\kappa \in \mathcal{SST}(P)$.

Part (2). Let $I^\kappa \in \mathcal{SST}(P)$. We show that $\beta(I^\kappa)$ is a bi-model of P that satisfies (i)-(iii). Let $(I,J) = \beta(I^\kappa)$ and let M be a maximal canonical answer set of P^κ corresponding to I^κ. Then, $\beta(M \cap \mathcal{L}^\kappa) = (I,J)$ by construction, and (I,J) satisfies (i) by Lemma 2.

Towards a contradiction first assume that (I,J) does not satisfy (iii). Then there exists a bi-model (I',J') of P such that (I',J') satisfies (i) and $J' \setminus I' \subset J \setminus I$. Let $M' = (I',J')^{\kappa,P}$ and note that if $M' \in \mathcal{AS}(P^\kappa)$, we arrive at a contradiction to $M \in mc(\mathcal{AS}(P^\kappa))$, since $gap(M') \subset gap(M)$. Thus, there exists $M'' \in \mathcal{AS}(P^\kappa)$, such that $M'' \subset M'$. Let $(I'',J'') = \beta(M'' \cap \mathcal{L}^\kappa)$. We show that (I'',J') is a bi-model of P, and thus by (i) it follows that $I'' = I'$. Towards a contradiction, suppose that (I'',J') is not a bi-model of P. Then, by Proposition 1, there exists $r \in P$, such that $B^+(r) \subseteq I''$, $J' \cap B^-(r) = \emptyset$, and either $I'' \cap H(r) = \emptyset$ or $I'' \cap B^-(r) \neq \emptyset$. Note that $B^+(r) \subseteq I''$ implies $B^+(r) \subseteq I'$, and since (I',J') is a bi-model of P, we conclude $I' \cap H(r) \neq \emptyset$ and $I' \cap B^-(r) = \emptyset$. The latter implies $I'' \cap B^-(r) = \emptyset$, hence $I'' \cap H(r) = \emptyset$ holds. If $B^-(r) = \emptyset$, then r is in P^κ and $M'' \not\models r$, contradiction. Thus, $B^-(r) \neq \emptyset$. However, in this case the epistemic transformation of r is in P^κ. Since $J' \cap B^-(r) = \emptyset$ and $J'' \subseteq J'$ together imply $J'' \cap B^-(r) = \emptyset$, we conclude that for the rule of the form (3) of the epistemic transformation of r, it holds that $\{b_1, \ldots, b_m\} \subseteq M''$ (due to $B^+(r) \subseteq I''$), and that $M'' \not\models Kb_{m+1} \vee \ldots \vee Kb_n$ (due to $J'' \cap B^-(r) = \emptyset$). Moreover $M'' \models P^\kappa$, hence $\lambda_{r,i}$ is in M'', for some $1 \leq i \leq l$. Considering the corresponding rule of the form (4) of the epistemic transformation of r, we also conclude that $a_i \in M''$, a contradiction to $I'' \cap H(r) = \emptyset$. This proves that (I'',J') is a bi-model of P. From the assumption that (I',J') satisfies (i), it follows that $I'' = I'$. Therefore $gap(M'') \subseteq gap(M')$ holds, which implies $gap(M'') \subset gap(M)$, a contradiction to $M \in mc(\mathcal{AS}(P^\kappa))$. This proves (I,J) satisfies (iii).

Next assume that (I, J) does not satisfy (ii). Then, there exists a bi-model (I, J') of P, such that $J' \subset J$. We show that (I, J') satisfies (i). Otherwise, there exists a bi-model (I', J') of P, such that $I' \subset I$; but then also (I', J) is a bi-model of P. To see the latter, assume that there exists a rule $r \in P$, such that $B(r) \subseteq I'$, $J \cap B^-(r) = \emptyset$ and either $I' \cap H(r) = \emptyset$ or $I' \cap B^-(r) \neq \emptyset$. Since $J' \subset J$, it then also holds that $J' \cap B^-(r) = \emptyset$. This contradicts the assumption that (I', J') is a bi-model of P, hence $(I', J) \models_\beta P$. The latter is a contradiction to the assumption that (I, J) satisfies (i), proving that (I, J') satisfies (i). Since (I, J) satisfies (iii), we conclude that $J' \setminus I = J \setminus I$. Now let $S' = \{\lambda_{r,i} \mid \lambda_{r,i} \in (I, J')^{\kappa,P}\}$ and let $S = \{\lambda_{r,i} \mid \lambda_{r,i} \in M\}$. It holds that $S' \not\subseteq S$ (otherwise $(I, J')^{\kappa,P} \subset M$, a contradiction to $M \in \mathcal{AS}(P^\kappa)$), i.e., there exists $r \in P$ of the form (1) and $1 \leq i \leq l$, such that $\lambda_{r,i} \in S$ and $\lambda_{r,i} \notin S'$. From the former, since M is a minimal model of P^κ, we conclude that $I \models B^+(r)$, $a_i \in I$, and $J \cap B^-(r) = \emptyset$. Since $J' \subset J$, also $J' \cap B^-(r) = \emptyset$. This implies that $\lambda_{r,k} \in S'$, for some $1 \leq k \neq i \leq l$ (otherwise $(I, J')^{\kappa,P}$ does not satisfy the rule of form (3) corresponding to r in P^κ, a contradiction to $(I, J')^{\kappa,P} \models P^\kappa$). However, since $a_i \in I$, and thus $a_i \in (I, J')^{\kappa,P}$, and since $\lambda_{r,k} \in (I, J')^{\kappa,P}$, we conclude that $\lambda_{r,i} \in (I, J')^{\kappa,P}$ (cf. the respective rule of form (6) of the epistemic transformation of r). This contradicts $\lambda_{r,i} \notin S'$, and thus proves that (I, J) satisfies (ii). □

Proof of Proposition 3. Let P be a program over $\mathcal{L}$.

Part (1). Let (I, J) be a bi-model of P, such that $(I, J)^\kappa$ satisfies Property **N** and Property **K**, for all $r \in P$. We show that (I, J) is an HT-model of P. Since $(I, J)^\kappa$ satisfies Property **N**, it holds that $a \in I$ implies $a \in J$, therefore $I \subseteq J$, i.e., (I, J) is an HT-interpretation. For every rule $r \in P$, $(I, J) \models_\beta r$ implies $(I, J) \not\models_\beta B(r)$, or $(I, J) \models_\beta H(r)$ and $I \models B(r)$. First suppose that $(I, J) \not\models_\beta B(r)$. Then $(I, J) \not\models B(r)$ (note that for a conjunction of literals, such as $B(r)$, the satisfaction relations do not differ). Moreover, since $(I, J)^\kappa$ satisfies Property **K** for r, it holds that $J \models r$. To see the latter, let Kr denote the rule obtained from r by replacing every $a \in \mathcal{L}$ occurring in r by Ka, and let KJ denote the set $\{Ka \in (I, J)^\kappa \mid a \in \mathcal{L}\}$. Then, $(I, J)^\kappa$ satisfies Property **K** for r iff $KJ \models Kr$. Observing that $KJ = \{Ka \mid a \in J\}$, we conclude that $J \models r$. This proves $(I, J) \models r$, if $(I, J) \not\models_\beta B(r)$. Next assume that $(I, J) \models_\beta H(r)$ and $I \models B(r)$. We conclude that $(I, J) \models H(r)$ (the satisfaction relations also coincide for disjunctions of atoms). As $(I, J)^\kappa$ satisfies Property **K** for r, it follows $J \models r$. This proves $(I, J) \models r$, for every $r \in P$; in other words, (I, J) is an HT-model of P.

Part (2). Let (H, T) be an HT-model of P. We show that $(H, T)^\kappa$ satisfies Property **N** and Property **K**, for all $r \in P$. As a consequence of $H \subseteq T$, for every $a \in (H, T)^\kappa$ such that $a \in \mathcal{L}$, it also holds that $Ka \in (H, T)^\kappa$, i.e., $(H, T)^\kappa$ satisfies Property **N**. Moreover, $(H, T) \models P$ implies $T \models r$, for all $r \in P$. Let $KT = \{Ka \mid a \in T\}$ and let Kr as above; $T \models r$ implies $KT \models Kr$, for all $r \in P$. By construction of $(H, T)^\kappa$ and definition of Property **K** for r, we conclude that $(H, T)^\kappa$ satisfies Property **K** for all $r \in P$. □

Proof of Theorem 2. Let P be a program over $\mathcal{L}$.

Part (1). Let (H, T) be an HT-model of P that satisfies (i′) and (ii′). We show that $(H, T)^\kappa \in \mathcal{SEQ}(P)$. Towards a contradiction, first assume that $(H, T)^\kappa \notin \mathcal{MM}(HT^\kappa(P))$. Then, there exists an HT-model (H', T') of P, such that $H' \subseteq H$, $T' \subseteq T$, and at least one of the inclusions is strict. Suppose that $H' \subset H$. Then (H', T) is an HT-model of P (by a well-known property of HT), a contradiction to the assumption that (H, T) satisfies (i′). Hence, $H' = H$ and $T' \subset T$ must hold. Moreover, by the same argument (H', T') also satisfies (i′). But, since $T' \setminus H' \subset T \setminus H$, this is in contradiction to the assumption that (H, T) satisfies (ii′). Consequently, $(H, T)^\kappa \in \mathcal{MM}(HT^\kappa(P))$. We continue the indirect proof assuming that $(H, T)^\kappa \notin mc(\mathcal{MM}(HT^\kappa(P)))$, i.e., there exists an HT-model (H', T') of P, such that $T' \setminus H' \subset T \setminus H$ and $(H', T')^\kappa \in \mathcal{MM}(HT^\kappa(P))$. The latter obviously implies that (H', T') satisfies (i′). Again, this contradicts that (H, T) satisfies (ii′), which proves that $(H, T)^\kappa \in \mathcal{SEQ}(P)$.

Part (2). Let $I^\kappa \in \mathcal{SEQ}(P)$. We show that $\beta(I^\kappa)$ is an HT-model of P that satisfies (i′) and (ii′). Let $\beta(I^\kappa) = (H, T)$. Towards a contradiction first assume that (H, T) is not an HT-model of P. Then by the definition of $\mathcal{SEQ}(P)$, and the fact that I^κ uniquely corresponds to sets H and T, we conclude that $I^\kappa \notin mc(\mathcal{MM}(HT^\kappa(P)))$, i.e., $I^\kappa \notin \mathcal{SEQ}(P)$; contradiction. Next, suppose that (H, T) does not satisfy (i′). Then, $I^\kappa \notin \mathcal{MM}(HT^\kappa(P))$, as witnessed by $(H', T)^\kappa$ for an HT-model (H', T) such that $H' \subset H$, which exists if (H, T) does not satisfy (i′). Therefore, $I^\kappa \notin mc(\mathcal{MM}(HT^\kappa(P)))$, i.e., $I^\kappa \notin \mathcal{SEQ}(P)$; contradiction. Eventually assume that (H, T) does not satisfy (ii′). Then, $I^\kappa \notin mc(\mathcal{MM}(HT^\kappa(P)))$, as witnessed by $(H', T')^\kappa$ for an HT-model (H', T'), such that $T' \setminus H' \subset T \setminus H$ and (H', T') satisfies (i′)—note that (H', T') exists if (H, T) does not satisfy (ii′). To see that $(H', T')^\kappa$ is a witness for $I^\kappa \notin mc(\mathcal{MM}(HT^\kappa(P)))$, observe that either $(H', T')^\kappa \in \mathcal{MM}(HT^\kappa(P))$ or there exists an HT-model (H', T''), such that $(H', T'')^\kappa \in \mathcal{MM}(HT^\kappa(P))$ and $T'' \subset T'$ (which implies $T'' \setminus H' \subset T' \setminus H' \subset T \setminus H$). This proves that $I^\kappa \notin \mathcal{SEQ}(P)$, again a contradiction. This concludes the proof that $\beta(I^\kappa)$ is an HT-model of P that satisfies (i′) and (ii′). □

Proof of Theorem 3. Let P be a program over $\mathcal{L}$, and let I^κ be an interpretation over $\mathcal{L}^\kappa$. The proof uses the following lemmas.

Lemma 4 *If $M \models P^{HT}$, then $\beta(M \cap \mathcal{L}^\kappa)$ is an HT-model of P.*

Lemma 5 *For every $M \in \mathcal{AS}(P^{HT})$, $\beta(M \cap \mathcal{L}^\kappa)$ satisfies* (i′) *in Theorem 2.*

Lemma 6 *For every HT-model (H, T) of P that satisfies* (i′) *of Theorem 2, there exists some $M \in \mathcal{AS}(P^{HT})$ such that $gap(M) \subseteq gap((H, T)^\kappa)$.*

The proof of the theorem is then as follows.

(⇐) Suppose that $I^\kappa \in \{M \cap \mathcal{L}^\kappa \mid M \in mc(\mathcal{AS}(P^{HT}))\}$. We prove $I^\kappa \in \mathcal{SEQ}(P)$ via Theorem 2. Let $M \in mc(\mathcal{AS}(P^{HT}))$, such that $I^\kappa = M \cap \mathcal{L}^\kappa$, and let $(I, J) =$

$\beta(M \cap \mathcal{L}^\kappa)$. Then, (I, J) is an HT-model of P by Lemma 4 and (I, J) satisfies (i′) in Theorem 2 by Lemma 5. We prove that (I, J) satisfies (ii′) in Theorem 2. Towards a contradiction, assume that this is not the case, then there exists an HT-model (H, T) of P, such that $T \setminus H \subset J \setminus I$ and (H, T) satisfies (i′). According to Lemma 6, there exists $M' \in \mathcal{AS}(P^{HT})$, such that $gap(M') \subseteq gap((H, T)^\kappa)$, which implies $gap(M') \subset gap(M)$ due to $T \setminus H \subset J \setminus I$. This contradicts the assumption that $M \in mc(\mathcal{AS}(P^{HT}))$, and thus proves that (I, J) satisfies (ii′) in Theorem 2. We conclude that $I^\kappa \in \mathcal{SEQ}(P)$.

($\Rightarrow$) Suppose that $I^\kappa \in \mathcal{SEQ}(P)$. We prove $I^\kappa \in \{M \cap \mathcal{L}^\kappa \mid M \in mc(\mathcal{AS}(P^{HT}))\}$. Let $(H, T) = \beta(I^\kappa)$. By Theorem 2, (H, T) is an HT-model of P that satisfies (i′) and (ii′). We show that there exists $M \in mc(\mathcal{AS}(P^{HT}))$ such that $\beta(M \cap \mathcal{L}^\kappa) = (H, T)$. Since $(H, T)^{\kappa,P} \models P^{HT}$, there exists $M \in \mathcal{AS}(P^{HT})$ such that $M \subseteq (H, T)^{\kappa,P}$. Since (H, T) satisfies (i′), it holds that $M \cap \mathcal{L} = H$. Moreover, $M \cap \mathcal{L}^\kappa \subset (H, T)^\kappa$ contradicts the fact that (H, T) satisfies (ii′), because then $\beta(M \cap \mathcal{L}^\kappa) = (H, T')$ is an HT-model of P, such that $T' \setminus H \subset T \setminus H$ and (H, T') satisfies (i′) due to Lemma 5. Hence, $\beta(M \cap \mathcal{L}^\kappa) = (H, T)$. It remains to show that $M \in mc(\mathcal{AS}(P^{HT}))$. If this is not the case, then some HT-model (H', T') of P exists such that $T' \setminus H' \subset T \setminus H$. Since $(H', T') = \beta(M' \cap \mathcal{L}^\kappa)$ for some $M' \in \mathcal{AS}(P^{HT})$, we conclude by Lemma 5 that (H', T') satisfies (i′), which again leads to a contradiction of the fact that (H, T) satisfies (ii′). This proves that $M \in mc(\mathcal{AS}(P^{HT}))$. As $M \cap \mathcal{L}^\kappa = I^\kappa$, we conclude that $I^\kappa \in \{M \cap \mathcal{L}^\kappa \mid M \in mc(\mathcal{AS}(P^{HT}))\}$. □

Proof of Proposition 4. Let P be a program over $\mathcal{L}$. If P has a model M, then (M, M) is an HT-model of P. Therefore $HT^\kappa(P) \neq \emptyset$, which implies $\mathcal{MM}(HT^\kappa(P)) \neq \emptyset$, and thus $mc(\mathcal{MM}(HT^\kappa(P))) \neq \emptyset$. We conclude that $\mathcal{SEQ}(P) \neq \emptyset$, i.e., P has a semi-equilibrium model. □

Proof of Proposition 5. Let P be a coherent program over $\mathcal{L}$, and let $Y \in \mathcal{AS}(P)$. Then (Y, Y) is an HT-model of P that satisfies (i′) in Theorem 2, since it is in equilibrium. Moreover, it trivially satisfies also (ii′) because $Y \setminus Y = \emptyset$. Hence, $(Y, Y)^\kappa \in \mathcal{SEQ}(P)$.

As P is coherent, there exists $(T, T) \in HT(P)$ that satisfies (i′) in Theorem 2 and (trivially) (ii′). Hence, $gap(I^\kappa) = \emptyset$ for all $I^\kappa \in \mathcal{SEQ}(P)$. Moreover, $\beta(I^\kappa)$ is of the form (Y, Y), and $Y \in \mathcal{AS}(P)$. □

References

Alcântara, J.; Damásio, C. V.; and Pereira, L. M. 2004. A declarative characterization of disjunctive paraconsistent answer sets. In de Mántaras, R. L., and Saitta, L., eds., *ECAI*, 951–952. IOS Press.

Brewka, G., and Eiter, T. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. In *Proceedings 22nd Conference on Artificial Intelligence (AAAI '07), July 22-26, 2007, Vancouver*, 385–390. AAAI Press.

Cabalar, P.; Odintsov, S. P.; Pearce, D.; and Valverde, A. 2007. Partial equilibrium logic. *Ann. Math. Artif. Intell.* 50(3-4):305–331.

Dao-Tran, M.; Eiter, T.; Fink, M.; and Krennwallner, T. 2009. Modular nonmonotonic logic programming revisited. In Hill, P., and Warren, D., eds., *Proceedings 25th International Conference on Logic Programming (ICLP 2009)*, number 5649 in Lecture Notes in Computer Science, 145–159. Springer.

Eiter, T., and Gottlob, G. 1995. On the Computational Cost of Disjunctive Logic Programming: Propositional Case. *Annals of Mathematics and Artificial Intelligence* 15(3/4):289–323.

Eiter, T.; Leone, N.; and Saccà, D. 1997. On the Partial Semantics for Disjunctive Deductive Databases. *Annals of Mathematics and Artificial Intelligence* 19(1/2):59–96.

Gelfond, M., and Lifschitz, V. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9:365–385.

Janhunen, T.; Oikarinen, E.; Tompits, H.; and Woltran, S. 2009. Modularity Aspects of Disjunctive Stable Models. *Journal of Artificial Intelligence Research* 35:813–857.

Kakas, A. C., and Mancarella, P. 1990. Generalized stable models: A semantics for abduction. In *ECAI*, 385–391.

Marek, V. W.; Nerode, A.; and Remmel, J. B. 1999. Logic programs, well-orderings, and forward chaining. *Annals of Pure and Applied Logic* 96(1-3):231–276.

Odintsov, S. P., and Pearce, D. 2005. Routley semantics for answer sets. In Baral, C.; Greco, G.; Leone, N.; and Terracina, G., eds., *LPNMR*, volume 3662 of *Lecture Notes in Computer Science*, 343–355. Springer.

Osorio, M.; Ramírez, J. R. A.; and Carballido, J. L. 2008. Logical weak completions of paraconsistent logics. *J. Log. Comput.* 18(6):913–940.

Pearce, D., and Valverde, A. 2008. Quantified equilibrium logic and foundations for answer set programs. In de la Banda, M. G., and Pontelli, E., eds., *ICLP*, volume 5366 of *Lecture Notes in Computer Science*, 546–560. Springer.

Pereira, L. M., and Pinto, A. M. 2005. Revised stable models - a semantics for logic programs. In Bento, C.; Cardoso, A.; and Dias, G., eds., *EPIA*, volume 3808 of *Lecture Notes in Computer Science*, 29–42. Springer.

Przymusinski, T. 1991. Stable Semantics for Disjunctive Programs. *New Generation Computing* 9:401–424.

Sakama, C., and Inoue, K. 1995. Paraconsistent stable semantics for extended disjunctive programs. *J. Log. Comput.* 5(3):265–285.

van Gelder, A.; Ross, K.; and Schlipf, J. 1991. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM* 38(3):620–650.

Wang, Y.; Zhang, M.; and You, J.-H. 2009. Logic programs, compatibility and forward chaining construction. *J. Comput. Sci. Technol.* 24(6):1125–1137.

You, J.-H., and Yuan, L. 1994. A Three-Valued Semantics for Deductive Databases and Logic Programs. *Journal of Computer and System Sciences* 49:334–361.