Abduction, Argumentation and Bi-disjunctive Logic Programs *

Kewen Wang
Department of Computer Science and Technology
Changsha Institute of Technology, Changsha 410073, P. R. China
email: wkw@nudt.edu.cn

Abstract

Many efforts have been devoted to extending normal logic programming to perform abduction and argumentation. However, the problem of performing argumentation-besed abduction in *disjunctive* logic programming (DLP) has been rarely investigated. In this paper, we presents an argumentation-theoretic framework for DLP, which is an interesting and non-trivial generalization of Dung and Torres' work. The novelty of this framework is that the set of *disjuncts of negative literals* is taken as the set of possible assumptions rather than only negative literals. As a by-product, our framework can also be defined on the so-called bi-disjunctive programs (a more wide class of logic programs than disjunctive logic programs by allowing non-classical disjunction in the bodies of programs). This simple generalization of DLP is called bi-disjunctive logic programming (BDLP) and the corresponding abductive semantics is denoted as BDAS. In BDAS, each bi-disjunctive program is naturally considered as an argument framework and an acceptable hypothesis is defined as a set of disjuncts of negative literals that satisfies some kind of stability condition. We identifies three kinds of acceptible hypotheses PDH, CDH and WFDH to represent credulous, moderate and skeptical reasoning in AI, respectively. BDAS can not only handle the problems of commonsense reasoning properly, but also characterizes and extends many key semantics, such as the minimal models, EGCWA, well-founded model, and stable models.. The most interesting and novel result shown in this paper is that WFDH extends both the well-found semantics for normal logic programs[6)] and the extended generalized closed world assumption (EGCWA) to the class of general disjunctive programs in a natural way (and thus provides a unifying characterization for these two different semantics through abduction); also, PDH extends the disjunctive stable models13) and is complete. Thus, BDAS provides a simple, intuitive and flexible semantic framework for disjunctive logic programming.

Keywords: Disjunctive Logic Programs, Abduction, Argumentation , Acceptable Disjunctive Hypotheses, Semantics

1 Introduction

Abduction is usually defined as inferring the best or most reasonable explanation (or hypothesis) for a given set of facts. Moreover, it is a form of nonmonotonic reasoning, since explanations which are consistent in a given context may become inconsistent when new information is obtained. In fact, abduction plays an important role in much of human inference. It is relevant in our everyday commonsense reasoning as well as in many expert problem-solving tasks. Therefore, the computer-implementation of abduction is a key topic in artificial intelligence (AI). Several efforts have been recently devoted to extending normal logic programming

---

*The main contents of this paper is a part of the author's PhD. thesis (1996.4).

to perform abductive reasoning[5]. Eshghi and Kowalski[4] have first recognized the relationship between negation as failure (NAF) and abduction. They have proposed an abductive semantics for normal logic programs, which is equivalent to the stable semantics[3]. Their analysis is based on abductive frameworks of the form $< T, A, I >$, where T is a set of definite clauses, A a set of abducible predicates, and I a set of integrity constraints. I plays a crucial role in capturing the meaning of NAF. But this abductive semantics does not give meaning to programs that have no stable models and thus inherits the weakness of stable semantics: the existence of abductive solutions for normal programs is not guaranteed.

Kakas and Mancarella6) use a similar abductive framework as Ref.4) to define a generalization of stable models, called generalized stable models, and their approach is somewhat simpler. However, the existence of generalized stable models is not guaranteed and thus do not solve the problem of non-existence of stable models for normal logic programs. On the other hands, a flexible and unifying abductive framework is needed for various application problems.

To provide a suitable semantic framework for normal logic programming, Dung [2]), 3) and Torres18), 19) have proposed two new argumentation-theoretic semantics. Both of these equivalent approaches transform a normal logic program P into an argument framework $< P, \sim B_P, \leadsto_P >$, where $\sim B_P$ is the set of all (ground) negative literals in P and any subset Δ of $\sim B_P$ is considered as a hypothesis of P, $\leadsto_P$ is an attack relation among hypotheses of P. An admissible hypothesis Δ is defined by a kind of stability conditions through the attack relation.

It is known that the paradigm of disjunctive logic programming is significantly more expressive than normal logic programming, and the problem of how to extend some key approaches for normal logic programs to the class of disjunctive logic programs is very important but difficult. Currently, the relationship between disjunctive logic programming (DLP) and abduction is studied by many authors, such as Ref.1),10),14). However, the problem of defining an argumentation-based semantics for DLP is still open. In this paper, we first extend the syntax of disjunctive programs to bi-disjunctive logic programs by allowing non-classical disjunction in the bodies of the clauses of disjunctive logic programs and then present an argumentation-theoretic semantic framework for bi-disjunctive logic programs, called the bi-disjunctive abductive semantics (abbreviated as BDAS), which can be considered as a generalization of Dung's preferred scenarios[2),3] and Torres' non-deterministic well-founded semantics[18),19]. In fact, this paper is heavily influenced by their work. Our work also proves that this generalization is not trivial. In BDAS we transform a given disjunctive program P into an argument framework $F_P =< P, DB_P^-, \leadsto_P >$, where DB_P^- is the set of all *disjuncts of (ground) negative literals* in P, a subset Δ of DB_P^- is called a *disjunctive hypothesis* (or simply, hypothesis) of P, and $\leadsto_P$ is an attack relation among the hypotheses of P. One of the main novelty of this framework is that we take a set of disjuncts of negative liteals as hypothesis of F_P rather than only a set of negative literals. An admissible hypothesis Δ is one that can attack every hypothesis which attacks it. Based on this idea, we introduce mainly three subclasses of admissible hypotheses: preferred disjunctive hypothesis (PDH); complete disjunctive hypothesis (CDH); well-founded disjunctive hypothesis (WFDH). Each of these subclasses defines a disjunctive semantics and they are all complete for disjunctive programs, that is, every disjunctive programs have the corresponding hypotheses. We will show that BDAS can not only handle the problems of commonsense reasoning properly, but also characterizes and extends many key semantics. The most interesting and novel result is that WFDH extends both the well-found semantics for normal logic programs[6] and the extended generalized closed world assumption (EGCWA)[20] to the class of general disjunctive programs in a natural way (and thus provides a unifying characterization for these two different semantics through abduction); also, PDH extends the disjunctive stable models13) and is complete. As noted in Ref.3), the skepticism and credulism are two major semantic intuitions for knowledge representation. A skeptical reasoner does not infer any conclusion in uncertainty conditions, but a credulous reasoner tries to give conclusions as much as possible. BDAS integrates these two opposite semantic intuitions and, in particular, PDH and WFDH characterize credulism and skepticism, respectively.

Another characteristic of BDAS is its richer expressive power. It is known that, for the disjunctive program $P_2 = \{c \leftarrow\sim a; c \leftarrow\sim b\ a|b \leftarrow\}$, the atom c can be derived from P_2 under some semantics, such as the perfect model and disjunctive stable models[13]; but c can not derived from P_2 under other semantics, such as Ross's EWFS. Abstractly, we can not determine whether c should be derived from P_2. And more, one can enumerate a lot of application problems, in some of which c should be derived from P_2 but in others of which c should not be derived from P_2. BDAS provides such a semantic framework that the above two kinds of commonsense reasoning can be dealt with uniformly.

The rest of this paper is arranged as follows: Section 2 is devoted to establish our semantic framework BDAS; In Section 3 some applications of BDAS in commonsense reasoning are demonstrated; Section 4 shows some fundamental properties of BDAS; Section 5 studies the BDAS for positive disjunctive programs, in particular, a quite new characterization for EGCWA by abduction is provided; Section 6 considers the relationship between BDAS and the disjunctive stable semantics; Section 7 is our conclusion.

2 Bi-disjunctive Logic Programs and BDAS

In this section, we first introduce our definitions and notions, then establish the *bi-disjunctive abductive semantics (BDAS)*. As usual, without loss of generality, we consider only propositional logic programs, this means that a logic program is often understood as its ground instantiation.

2.1 Basic notions and definitions

A *Horn logic program* is a set of Horn clauses of the form

$$a \leftarrow a_1, \ldots, a_m,$$

where a and a_i $(i = 1, \ldots, m)$ are atoms and $m \geq 0$.

A *normal logic program* is a set of normal clauses of the form

$$a \leftarrow a_1, \ldots, a_s, \sim a_{s+1}, \ldots, \sim a_t,$$

where a and a_i $(i = 1, \ldots, t)$ are atoms and $t \geq s \geq 0$. The symbol $\sim$ denotes negation by default, rather than classical negation.

A *positive disjunctive logic program* is a set of positive disjunctive clauses of the form

$$a_1|\cdots|a_r \leftarrow a_{r+1}, \ldots, a_s,$$

where a_i $(i = 1, \ldots, s)$ are atoms and $s \geq r > 0$.

A *disjunctive logic program* is a set of disjunctive clauses of the form

$$a_1|\cdots|a_r \leftarrow a_{r+1}, \ldots, a_s, \sim a_{s+1}, \ldots, \sim a_t,$$

where a_i $(i = 1, \ldots, t)$ are atoms and $t \geq s \geq r > 0$.

In this subsection, we will extend the class of disjunctive logic programs to a more general class.

Definition 2.1

A *bi-disjunctive clause* C is a rule of the form

$$a_1|\cdots|a_r \leftarrow a_{r+1},\ldots,a_s,\beta_{s+1},\ldots,\beta_t,$$

where a_i $(i=1,\ldots,s)$ are atoms, β_j $(j=s+1,\ldots,t)$ are disjuncts of negative literals, and $t \geq s \geq r > 0$.

A *bi-disjunctive logic program* P is defined as a set of bi-disjunctive clauses. □

It is obvious that the following inclusions hold:

$$Bi-Disjunctive\ Programs \supset Disjunctive\ Programs \supset Normal\ Programs.$$

As usual, B_P denotes the *Herbrand base* of bi-disjunctive logic program P, that is, the set of all (ground) atoms in P. The set DB_P^+ of all disjuncts of the atoms in P is called the *disjunctive Herbrand base* of P; the set of all disjuncts of the negative literals in P is called the *negative disjunctive Herbrand base* of P. $\perp$ denotes the empty disjuncts.

If S is an expression, then $atoms(S)$ is the set of all atoms appearing in S.

For $\alpha,\beta \in DB_P^+$, if $atoms(\alpha) \subseteq atoms(\beta)$ then we say α implies β, denoted as $\alpha \Rightarrow \beta$. If $\alpha \in DB_P^+$, then $sfac(\alpha)$ is the *smallest factor* of α, that is the disjunct of atoms that is obtained from α by deleting all repeated occurrence of atoms in α. For instance, the smallest factor of $a|b|a$ is $a|b$. For $S \subseteq DB_P^+$, $sfac(S) = \{sfac(\alpha) : \alpha \in S\}$. The *expansion* of α is defined as $\parallel \alpha \parallel = \{\beta \in DB_P^+ : \alpha \Rightarrow \beta\}$; the expansion of S is $\parallel S \parallel = \{\beta \in DB_P^+ :$ there exists $\alpha \in S$ such that $\alpha \Rightarrow \beta\}$.

The *canonical form* of S is defined as the following:

$$can(S) = \{\alpha \in sfac(S) : \text{there exists no } \alpha' \in sfac(S) \text{such that } \alpha' \Rightarrow \alpha \text{ and } \alpha' \neq \alpha\}.$$

For $\alpha \in DB_P^-$ and $S \in DB_P^-$, $sfac(\alpha)$, $sfac(S)$, $\parallel \alpha \parallel$ and $\parallel S \parallel$ can be defined similarly.

A subset of DB_P^+ is called a *state* of the bi-disjunctive logic program P; a *state pair* of P is defined as $S = < S^+; S^- >$, where $S^+ \subseteq DB_P^+$ and $S^- \subseteq DB_P^-$.

The minimal models and the least model state are two important declarative semantics for positive disjunctive programs, both of which extend the least model theory of Horn logic programs. The minimal model semantics captures the disjunctive consequences from a positive disjunctive program as a set of models. The least model state captures the disjunctive consequences as a set of disjuncts of atoms and leads to a unique 'model' characterization.

Let P be a positive disjunctive program, then the *least model state* of P is defined as

$$ms(P) = \{\alpha \in DB_P^+ : P \vdash \alpha\},$$

where $\vdash$ is the inference of the first-order logic and P is considered as the corresponding first-order formulas.

The least model state $ms(P)$ of a positive disjunctive P can be characterized by the operator $T_P^S : 2^{DB_P^+} \to 2^{DB_P^+}$: for any $J \subseteq DB_P^+$,

$$T_P^S(J) = \{\alpha \in DB_P^+ : \text{ there exists a disjuncitve clause } \alpha' \leftarrow a_1,\ldots,a_n \text{ in } P \text{ and } a_i|\alpha_i \in J, i=1,$$

$$\ldots,n, \text{ such that } \alpha'' = \alpha'|\alpha_1|\cdots|\alpha_n, \text{ where } \alpha_1,\ldots,\alpha_n \in DB_P^+ \cup \{\perp\}, \text{ and } \alpha = sfac(\alpha'')\}.$$

Minker and Rajasekar, see Ref. 11), have shown that T_P^S has the least fixpoint $lfp(T_P^S) = T_P^S \uparrow \omega$, and the following result:

Theorem2.1

Let P be a positive disjunctive program, then $ms(P) = \| T_P^S \uparrow \omega \|$, and $ms(P)$ and P have the same set of minimal models.

2.2 The basic argumentation-theoretic framework

This subsection will devoted to establish our abdutive semantic framework for bi-disjunctive programs.

Given a bi-disjunctive program P, an element of DB_P^- is called a *disjunctive assumption (or assumption)of P and a subset Δ of DB_P^- is a disjunctive hypothesis (or simply, hypothesis) of P if $\| \Delta \| = \Delta$. In this paper, we will consider a bi-disjunctive program P as an argumentation-theoretic framework $\mathbf{F}_P =< P, H(P), \leadsto_P >$, where $H(P)$ is the set of all hypotheses of P, and $\leadsto_P$ is a binary relation on $H(P)$, called the attack relation of P. To define the attack relation, similar to the definition of GL-transformation[5)], we can define a generalized GL-transformation for the class of bi-disjunctive programs, by which a positve disjunctive progrm P_Δ^+ is obtained from any given bi-disjunctive program P and a (disjunctive) hypothesis Δ of P.*

Definition 2.2

Let Δ be a hypothesis, then

(1) For each bi-disjunctive clause C in P, delete all the disjuncts of negative literals in the body of C that belong to Δ. The resulting bi-disjunctive program is denoted as P_Δ;

(2) The positive disjunctive program consisting of all the positive disjunctive clauses of P_Δ is denoted as P_Δ^+, and is called the generalized GL-transformation of P. $\Box$

Example 2.1

Let P be the bi-disjunctive program: $\{a|b \leftarrow;\ e|c \leftarrow d, \sim a| \sim b;\ d \leftarrow \sim e\}$.

If $\Delta_1 = \| \sim a| \sim b \|$, then $P_{\Delta_1} = \{a|b \leftarrow;\ e|c \leftarrow d;\ d \leftarrow \sim e\}$, and $P_{\Delta_1}^+ = \{a|b \leftarrow;\ e|c \leftarrow d\}$. If $\Delta_2 = \| \sim a| \sim b, \sim e \|$, then $P_{\Delta_2}^+ = P_{\Delta_2} = \{a|b \leftarrow;\ e|c \leftarrow d;\ d \leftarrow\}$. $\Box$

As the basic inference relation to define the attack relation between hypotheses, we first introduce the inference $\vdash_P$, which is just a special kind of first-order inference.

Definition 2.3

Let Δ be a disjunctive hypothesis of bi-disjunctive program P, $\alpha \in DB_P^+$ and $\sim b_1, \ldots, \sim b_m \in \Delta$ such that the following two conditions are satisfied:

(1) $\beta = \alpha | b_1 | \cdots | b_m$; and

(2) $\beta \in can(ms(P_\Delta^+))$.

Then we call Δ is a supporting hypothesis for α, denoted as $\Delta \vdash_P \alpha$.

The set of all disjuncts of positive literals that are supported by Δ is denoted as $V_P(\Delta)$. That is,

$$V_P(\Delta) = \{\alpha \in DB_P^+ :\ \Delta \vdash_P \alpha\}. \Box$$

In example 2.1, $V_P(\Delta_1) = \{a|b\}$, $V_P(\Delta_2) = \{a|b, c, d\}$.

Definition 2.4

Let Δ be a hypothesis of bi-disjunctive program P, then $S_\Delta =< \| V_P(\Delta) \|; \Delta >$ is called a supported state pair of P. $\Box$

Though each hypothesis Δ corresponds to a state pair of P, not every state pair represent the intended meaning of P. For example $P = \{a \leftarrow \sim a\}$. If $\Delta = \| \sim a \|$, then $V_P(\Delta) = \{a\}$ and thus $S_\Delta =< \| a \|; \| \sim a \| >$. It is obvious that S_Δ does not represent the correct meaning of P. For the purpose of choosing correct hypothesis, we need the following attack relation among the hypotheses of P.

Definition 2.5

Let Δ and Δ' be two hypotheses of bi-disjunctive program P. If at least one of the following conditions holds:

(1) There exists $\beta = \sim b_1 | \cdots | \sim b_m \in \Delta'$ $m > 0$, such that $\Delta \vdash_P b_i, i = 1, \ldots, m$; or

(2) There exist $\sim b_1, \ldots, \sim b_m \in \Delta', m > 0$, such that $\Delta \vdash_P b_1 | \cdots | b_m$.

Then we say Δ attacks Δ', and denoted as $\Delta \rightsquigarrow_P \Delta'$. $\Box$

Example 2.2

Let P be the bi-disjunctive program of Example 2.1. Take $\Delta = \| \sim a | \sim b, \sim e \|$, $\Delta' = \| \sim c | \sim d \|$. Since $V_P(\Delta) = \{a|b, c, d\}$, that is, $\Delta \vdash_P c, d$ thus $\Delta \rightsquigarrow_P \Delta'$, but not $\Delta' \rightsquigarrow_P \Delta$.

For $P = \{a \leftarrow \sim a\}$ and $\Delta = \| \sim a \|$, it is not hard to see $\Delta \rightsquigarrow_P \Delta$, this means that Δ attacks itself. $\Box$

Intuitively, a plausible hypothesis should not attack itself.

Definition 2.6

A hypothesis Δ of bi-disjunctive program P is self-consistent if $\Delta \not\rightsquigarrow_P \Delta$. $\Box$

The empty hypothesis $\emptyset$ is always self-consistent, called trivial hypothesis. As noted above, there exist non-trivial hypotheses that are not self-consistent.

Corollary 2.1

A hypothesis Δ of bi-disjunctive program P is not self-consistent if and only if there exists $\sim b_1 | \cdots | \sim b_n \in DB_P^-$ such that $\Delta \vdash_P b_i$, $i = 1, \ldots, n$. $\Box$

Proof

The condition is obviously sufficient. For the necessity, suppose that $\Delta \rightsquigarrow_P \Delta$, According to Definition 2.5, there are possible two cases. If the case (1) holds, then it is just the required conclusion; If the case (2) holds, then there exist $\sim b_1, \ldots, \sim b_m \in \Delta$ $(m > 0)$ such that $\Delta \vdash_P b_1 | \cdots | b_m$. It follows from Definition 2.3 that $\Delta \vdash_P b_1$ and $\sim b_1 \in \Delta$. Thus, the required conclusion also holds. $\Box$

Definition 2.7

For any self-consistent hypothesis Δ of bi-disjunctive program P, the corresponding state pair S_Δ is called a self-consistent state pair of P. $\Box$

Definition 2.8

Let $S =< S_+; S_- >$ be a state pair of bi-disjunctive program P. If neither of the following two conditions hold, then we say S is simple-consistency:

(1) there exist $a_1, \ldots, a_r \in S_+$, such that $\sim a_1 | \cdots | \sim a_r \in S_-$; or

(2) there exists $a_1 | \cdots | a_r \in S_+$, such that $\sim a_1, \ldots, \sim a_r \in S_-$. $\Box$

Theorem 2.2

The self-consistent state pairs of a bi-disjunctive program P are simple-consistent. $\Box$

Proof

For any self-consistent state pair S of P, there ia a self-consistent hypothesis Δ of P such that $S =<\| V_P(\Delta) \|; \Delta >$. To the contrary, suppose that S is not simple-consistent, from Definition 2.5 and Definition 2.8, then $\Delta \rightsquigarrow_P \Delta$, a contradiction. $\Box$

Definition 2.9

A state pair $S =< S^+; S^- >$ is consistent if the set of the corresponding first-order formulas of $S^+ \cup S^-$ is consistent. $\Box$

A self-consistent state pair is not necessarily consistent.

Example 2.3

Let $P = \{a|b \leftarrow; b|c \leftarrow; c|a \leftarrow\}$. Take $\Delta = \| \sim a | \sim b, \sim b | \sim c, \sim c | \sim a \|$, then Δ is a self-consistent hypothesis. However, $V_P(\Delta) = \{a|b, b|c, c|a\}$ and $\| V_P(\Delta) \| \cup \Delta$ being considered as a set of first-order formulas is not consistent, thus the state pair $S_\Delta =<\| V_P(\Delta) \|; \Delta >$ is not consistent. $\Box$

Definition 2.10

Let P be a bi-disjunctive program, $\beta = \sim b_1 | \cdots | \sim b_m \in DB_P^-$, $\Delta \subseteq DB_P^-$. β is admissible with respect to Δ if $\Delta \rightsquigarrow_P \Delta'$ holds for any $\Delta' \subseteq DB_P^-$ such that $\Delta' \rightsquigarrow_P b_i, (i = 1, \ldots, m)$. Write

$\mathbf{A}_P(\Delta) = \{\beta \in DB_P^- : \beta$ *is admissible wrt.* $\Delta\}$. $\Box$

Corollary 2.2

If Δ *and* Δ' *are two hypotheses of bi-disjunctive program* P, *then*

(1) $\| \mathbf{A}_P(\Delta) \| = \mathbf{A}_P(\Delta)$, *that is,* $\mathbf{A}_P(\Delta)$ *is a hypothesis of* P;

(2) If $\Delta \subseteq \Delta'$, *then* $\mathbf{A}_P(\Delta) \subseteq \mathbf{A}_P(\Delta')$. *This means that* $\mathbf{A}_P$ *is a monotonic operator.* $\Box$

Proof

It is the direct result of Definition 2.10. $\Box$

2.3 Some important bi-disjunctive semantics

To characterize the intuitions of commonsense reasoning, such as credulism and skepticism, we introduce the following three subsets of admissible hypotheses.

Definition 2.11

Let Δ *be a hypothesis of bi-disjunctive program* P:

(1) *If* Δ *is self-consistent and* $\Delta \subseteq \mathbf{A}_P(\Delta)$, *then* Δ *is called an admissible disjunctive hypothesis (abbreviated as ADH) of* P;

(2) *A preferred disjunctive hypothesis (PDH)* Δ *of* P *is defined as a maximal ADH of* P *with respect to set inclusion;*

(3) *If* Δ *is self-consistent and* $\Delta = \mathbf{A}_P(\Delta)$, *then* Δ *is called a complete disjunctive hypothesis (CDH) of* P;

(4) *If the hypothesis* $\mathbf{A}_P \uparrow \omega$ *is self-consistent, then it is called the well-founded disjunctive hypothesis of* P, *denoted as* $WFDH(P)$.

$\Box$

If Δ *is an ADH (res. PDH, CDH, WFDH), then the corresponding supported state pair* S_Δ *is called an ADS (res. PDS, CDS, WFDS) of* P.

It follows easily from the above definition that a CDH must be an ADH; In Section 4 we will show that a PDH is a CDH. But the converses do not hold.

Example 2.4

Let P *consists of only one program clause:* $a|b \leftarrow$.

Take $\Delta_0 = \emptyset$, *then* $\mathbf{A}_P(\Delta_0) = \| \sim a| \sim b \|$. *Hence* Δ_0 *is an ADH of* P *but not a CDH. If* $\Delta_1 = \| \sim a| \sim b \|$, *then* $\mathbf{A}_P(\Delta_1) = \Delta_1$ *and thus* Δ_1 *is a CDH of* P *but not a PDH, since* $\Delta_2 = \| \sim a \|$ *is an ADH of* P *and* $\Delta_1 \subset \Delta_2$. $\Box$

Theorem 2.3

For any bi-disjunctive program P, *its semantics ADH, CDH and PDH are all complete, that is, every bi-disjunctive program possesses at least one ADH (resp, CDH and PDH).* $\Box$

However, we do not know whether WFDH is complete at present. It will be shown in Section 4 that WFDH is complete for the class of disjunctive programs.

As a special case, we consider the BDAS of normal logic programs. In the rest of this section, P *will be a normal program. Let* Δ *be a disjunctive hypothesis of* P, *that is,* $\Delta \subseteq DB_P^-$, *and* $L(\Delta)$ *denotes the set of all negative literals in* Δ.

Definition 2.12

A hypothesis Δ *of* P *is a non-disjunctive hypothesis of* P *if* $L(\Delta) = can(\Delta)$. $\Box$

It is known from Definition 2.3 that, for any normal program P *and* $a \in B_P$,

$$\Delta \vdash_P a \text{ iff } a \in Min(P_\Delta^+) \text{ iff } a \in Min(P_{L(\Delta)}^+).$$

Corollary 2.3

If Δ is a CDH of normal program P, then $L(\Delta) = can(\Delta)$, that is, the CDHs of a normal program are non-disjunctive. $\Box$

Proof

Without loss of generality, it is enough to show that: $\sim a| \sim b \in \Delta = \mathbf{A}_P(\Delta)$ implies either $\sim a \in \Delta$ or $\sim b \in \Delta$: On the contrary, suppose that $\sim a, \sim b \notin \Delta = \mathbf{A}_P(\Delta)$, then there exist non-disjunctive hypotheses Δ_a and Δ_b of P such that $\Delta_a \vdash_P a$, $\Delta_b \vdash_P b$ but $\Delta \not\leadsto_P \Delta_a$, $\Delta \not\leadsto_P \Delta_a$. Therefore, $\Delta \not\leadsto_P \Delta_a \cup \Delta_b$. It is obvious that $\Delta_a \cup \Delta_b \leadsto_P \|\sim a| \sim b\|$, which contradicts $\sim a| \sim b \in \mathbf{A}_P(\Delta)$. $\Box$

It follows from Corollary 2.3 and the result in Ref.[9] that, for any normal program P, we will get the equivalent definition of Definition 2.5 if the basic inference $\Delta \vdash_P a$ is replaced by $P \cup \Delta \vdash a$. This means that our CDH and Dung's complete extension are equivalent concepts for the class of normal programs.

Theorem 2.4

If Δ is a non-disjunctive hypothesis of normal program P, then the following two statements are equivalent:

(1) *Δ is a CDH of P;*

(2) *$P \cup L(\Delta)$ is a complete extension.*

$\Box$

This theorem shows that BDAS generalizes the frameworks of Dung[2)] and Torres[18)].

3 Commonsense Reasoning in BDAS

In this section we will demonstrate some applications of BDAS and its suitability through some typical examples.

Example 3.1

Consider the program P: $a|b \leftarrow$.

This program has four ADHs: $\Delta_0 = \emptyset$, $\Delta_1 = \|\sim a\|$, $\Delta_2 = \|\sim b\|$, $\Delta_3 = \|\sim a| \sim b\|$, among which $\Delta_1, \Delta_2, \Delta_3$ are all CDHs of P. P has only two PDHs: Δ_1, Δ_2 and just correspond to two stable (i.e. perfect) models of P, respectively. The least CDH $\Delta_3 = \|\sim a| \sim b\|$ is exactly the EGCWA(P). $\Box$

Example 3.2

Let P be the program: $a \leftarrow \sim a$; $b \leftarrow$. The possible disjunctive hypotheses of P are: $\Delta_0 = \emptyset$, $\Delta_1 = \|\sim a\|$, $\Delta_2 = \|\sim b\|$, $\Delta_3 = \|\sim a| \sim b\|$, among which Δ_1, Δ_2 are not self-consistent. Since $\Delta_1 \leadsto_P \Delta_3$ and $\Delta_3 \not\leadsto_P \Delta_1$, Δ_3 is not a ADH of P, thus P has only one ADH $\Delta_0 = \emptyset$ and the corresponding state pair $S_{\Delta_0} = <\| b \|; \emptyset>$. This conclusion coincides with our intuition on P, that is, P provides no information about a for us and thus, from P, we can infer neither a nor $\sim a$, but can infer b. This example shows that BDAS can handle the inconsistency of disjunctive programs properly. Note that the Clark completion of P is not consistent and P has no stable model. $\Box$

Example 3.3

Let $P = \{a|b \leftarrow;\ c \leftarrow \sim a;\ c \leftarrow \sim b\}$. We need to consider only the following seven assumptions of P:

$$\sim a, \sim b, \sim c, \sim a| \sim b, \sim b| \sim c, \sim c| \sim a, , \sim a| \sim b| \sim c.$$

The possible hypotheses of P has 19:

$$\begin{array}{ll}
\Delta_0=\emptyset, & \Delta_1=\|\sim a\|, \\
\Delta_2=\|\sim b\|, & \Delta_3=\|\sim c\|, \\
\Delta_4=\|\sim a|\sim b\|, & \Delta_5=\|\sim b|\sim c\|, \\
\Delta_6=\|\sim a|\sim c\|, & \Delta_7=\|\sim a|\sim b|\sim c\|, \\
\Delta_8=\|\sim a,\sim b\|, & \Delta_9=\|\sim a,\sim c\|, \\
\Delta_{10}=\|\sim b,\sim c\|, & \Delta_{11}=\|\sim a,\sim b|\sim c\|, \\
\Delta_{12}=\|\sim a|\sim c,\sim b\|, & \Delta_{13}=\|\sim a|\sim b,\sim c\|, \\
\Delta_{14}=\|\sim a|\sim b,\sim c|\sim a\|, & \Delta_{15}=\|\sim a|\sim b,\sim b|\sim c\|, \\
\Delta_{16}=\|\sim b|\sim c,\sim c|\sim a\|, & \Delta_{17}=\|\sim a,\sim b,\sim c\|,
\end{array}$$

$$\Delta_{18}=\|\sim a|\sim b,\sim b|\sim c,\sim c|\sim a\|,$$

where $\Delta_0,\Delta_1,\Delta_2,\Delta_4,\Delta_{17}$ *are all the ADHs of P;* $\Delta_1,\Delta_2,\Delta_4$ *are CDHs; the PDHs* Δ_1,Δ_2 *correspond to the stable models* $\{b,c\}$ *and* $\{a,c\}$*. WFDH of P is* Δ_4 *and the state pair* $S_{\Delta_4}=<\| a|b\|;\|\sim a|\sim b\|>$*.*

Note that the least stationary model[14] *and the static model*[15] *of P coincide and equal to* $\bar{S}=<\| a|b,c\|,\|\sim a|\sim b\|>$*. It is obvious that* $S^-_{\Delta_4}=\bar{S}^-$ *and* $c\in\bar{S}^-$*, but* $c\notin S^-_{\Delta_4}$*. That is, BDAS infers the same negative information from P as the above disjunctive semantics but can not infer c. The main reason is that ordinary disjunctive semantics interpret* $P_1=\{c\leftarrow\sim a|\sim b\}$ *as the equivalent program* $P_2=\{c\leftarrow\sim a;\ c\leftarrow\sim b\}$*. To argumenting about whether c should be derived from P may be no sense, because one can enumerate a lot of examples of commonsense reasoning, in some of which inferring a from P is reasonable but others of which do not allow one inferring a from P. However, no disjunctive semantics can handles these two kinds of reasoning at the same time. Because it assigns different meaning to* P_1 *and* P_2*, BDAS is not only intuitive but also can handle these problems of commonsense reasoning more conveniently.* □

4 Some Fundamental Properties of BDAS

As the basis for further investigation, this section is devoted to studies the properties of BDAS. First, we give an equivalent characterization of admissible hypotheses.

Theorem 4.1

Let P be a self-consistent hypothesis of bi-disjunctive program P. Then Δ *is an ADH of P if and only if* $\Delta\leadsto_P\Delta'$ *for any hypothesis* Δ' *of P satisfying* $\Delta'\leadsto_P\Delta$*.* □

This theorem shows that an ADH is such a hypothesis that can attack any hypothesis that attacks it.

Proof

$\Leftarrow)$: *To show that* Δ *is admissible in the sense of Definition 2.10. Because* Δ *is self-consistent, we need only to prove that* $\Delta\subseteq\mathbf{A}_P(\Delta)$*. For any* $\beta=\sim b_1|\cdots|\sim b_m\in\Delta$*, if* Δ' *is a hypothesis of P such that* $\Delta'\vdash_P b_i, i=1,\ldots,m$*, then* $\Delta'\leadsto_P\Delta$ *and thus* $\Delta\leadsto_P\Delta'$ *from the assumption of the theorem. This means that* $\beta\in\mathbf{A}_P(\Delta)$*, that is,* $\Delta\subseteq\mathbf{A}_P(\Delta)$*. Hence* Δ *is an ADH.*

$\Rightarrow)$: *Suppose that* Δ *is an ADH of P. If* $\Delta'\leadsto_P\Delta$ *for a hypothesis* Δ' *of P, then there are two possible cases:*

Case 1. There exists a disjunct of negative literals $\sim b_1|\cdots|\sim b_m\in\Delta$ *such that* $\Delta'\vdash_P b_i, i=1,\ldots,m$*; or*

Case 2. There exist negative literals $\sim b_1,\ldots,\sim b_m\in\Delta$ *such that* $\Delta'\vdash_P b_1|\cdots|b_m$*.*

If Case 1 holds, since $\sim b_1|\cdots|\sim b_m\in\Delta\subseteq\mathbf{A}_P(\Delta)$*, it is obvious that* $\Delta\leadsto_P\Delta'$*; For Case 2, then* $\|\Delta'\cup\{\sim b_2,\ldots,\sim b_m\}\|\vdash_P b_1$ *and* $\sim b_1\in\Delta$ *is admissible with respect to* Δ*. Thus* $\Delta\leadsto_P\Delta''$*, where* $\Delta''=\|\Delta'\cup\{\sim b_2,\ldots,\sim b_m\}\|;$*. This case can be again divided into two subcases:*

Subcase 1. There exists $\sim c_1|\cdots|\sim c_n \in \Delta$ such that $\Delta\vdash_P c_1,\ldots,c_n$: Suppose that there exists $i(1\leq i\leq n)$ such that $\sim c_i\in\{\sim b_2,\ldots,\sim b_m\}$ and $m\geq 2$, then $\sim c_i\in\Delta$. On the other hand, $\Delta\vdash_P c_i$, we have $\Delta\leadsto_P\Delta$, this is impossible. Therefore, $\{\sim c_1,\ldots,\sim c_n\}\cap\{\sim b_2,\ldots,b_m\}=\emptyset$. It is the case that $\sim c_1|\cdots|\sim c_n\in\Delta'$, and thus $\Delta\leadsto_P\Delta'$.

Subcase 2. There exist $\sim c_1,\ldots,\sim c_n\in\Delta''$ such that $\Delta\vdash_P c_1|\cdots|c_n$: Let $\{\sim c_1,\ldots,\sim c_t\}\cap\{\sim b_2,\ldots,\sim b_m\}=\emptyset$, and $\{\sim c_{t+1},\ldots,\sim c_n\}\subseteq\{\sim b_2,\ldots,b_m\}\subseteq\Delta, 0\leq t\leq n$. We can assert that $t\neq 0$: otherwise, if $t=0$, that is, $\{\sim c_1,\ldots,\sim c_n\}\subseteq\{\sim b_2,\ldots,\sim b_m\}\subseteq\Delta$, this contradicts the self-consistency of Δ. Hence, $1\leq t\leq n$. We have $\sim c_1,\ldots,\sim c_t\in\Delta'$ and $\Delta'\vdash_P c_1|\cdots|c_t$, this also means $\Delta\leadsto_P\Delta'$.

In any case, we have $\Delta\leadsto_P\Delta'$ whenever $\Delta'\leadsto_P\Delta$. □

In the following we will characterize ADHs in another way.

Definition 4.1

Let Δ and Δ' be ADHs of bi-disjunctive program P. If $\Delta\subseteq\Delta'$, then Δ' is called an admissible extension of Δ. In particular, Δ' is called a non-trivial admissible extension of Δ if $\Delta\neq\Delta'$. □

Definition 4.2

Let Δ be an ADH of bi-disjunctive program P. If Δ' satisfies the following two conditions:

(1) $\Delta\cup\Delta'$ is self-consistent;

(2) $\Delta'\subseteq\mathbf{A}_P(\Delta\cup\Delta')$.

then Δ' is called a plausible hypothesis with respect to Δ. □

The following three corollaries can be easily derived from Definition 4.1 and Definition 4.2.

Corollary 4.1

If Δ' is a plausible hypothesis wrt an ADH Δ, then $\Delta\cup\Delta'$ is an ADH. □

Corollary 4.2

Δ' is an admissible extension of Δ if and only if $\Delta\subseteq\Delta'$ and $\Delta'\setminus\Delta$ is plausible with respect to Δ. □

Corollary 4.3

For any bi-disjunctive program P, the following statements are equivalent:

(1) Δ is an ADH of P;

(2) Δ is an admissible extension of the empty hypothesis $\emptyset$;

(3) Δ is plausible with respect to $\emptyset$. □

Definition 4.3

A sequence of admissible extensions of bi-disjunctive program P is a sequence $\Delta_1,\Delta_2,\ldots,\Delta_n,\ldots$ of ADHs of P such that $\Delta_n\subseteq\Delta_{n+1}$ for any $n>0$. □

The following theorem states that the sequences of bi-disjunctive program P posses the property of completeness.

Theorem 4.2

If $\Delta_1,\Delta_2,\ldots,\Delta_n,\ldots$ is a sequence of admissible extensions of bi-disjunctive program P, then the hypothesis $\Delta=\cup_{n=1}^{\infty}\Delta_n$ is an ADH of P. □

Proof

If $\alpha\in\Delta$, then there exists $n\geq 1$ such that $\alpha\in\Delta_n$. Because Δ_n is admissible wrt. Δ, then $\alpha\in\Delta_n\subseteq\mathbf{A}_P(\Delta_n)\subseteq\mathbf{A}_P(\Delta)$ and thus $\Delta\subseteq\mathbf{A}_P(\Delta)$. It remains to prove that Δ is self-consistent: To the contrary, suppose that $\Delta\leadsto\Delta$. From Corollary 2.1, there exists $\alpha=\sim a_1|\cdots|\sim a_r\in\Delta$ such that $\Delta\vdash_P a_j, j=1,\ldots,r$. Since each a_j is derived by the finite number of resolutions from P_Δ^+ and Δ, we can choose the number n is large enough such that $\Delta_n\vdash_P a_j, j=1,\ldots,r$ and $\alpha\in\Delta_n$. Therefore, $\Delta_n\leadsto_P\Delta_n$, a contradiction. Thus Δ is self-consistent. We have proved that Δ is an ADH of P.

In particular, we have the following result:

Corollary 4.4
Every ADH of a bi-disjunctive program P is contained in a PDH. $\Box$

Theorem 4.3 *For any ADH Δ of disjunctive program P, if $\alpha \in DB_P^-$ is admissible wrt. Δ, that is, $\alpha \in \mathbf{A}_P(\Delta)$, then $\Delta' = \| \Delta \cup \{\alpha\} \|$ is also an ADH of P.* $\Box$

This theorem guarantees that, for any ADH Δ of disjunctive program P, if α is admissible wrt. Δ and $\alpha \notin \Delta$ then we can obtained a non-trivial admissible extension of Δ by simply adding α to Δ. Before giving the proof of this theorem, we first prove two lemmas.

Lemma 4.1
If Δ is an ADH of disjunctive program P and $\alpha = \sim a_1| \cdots | \sim a_m \in \mathbf{A}_P(\Delta)$, $m > 1$, then $\Delta' = \| \Delta \cup \{\alpha\} \|$ is also an ADH of P. $\Box$

Proof
By Corollary 2.2, $\Delta' = \| \Delta \cup \{\alpha\} \| \subseteq \mathbf{A}_P(\Delta')$. It is enough to show Δ' is self-consistent. Suppose that $\Delta' \leadsto \Delta'$, by Corollary 2.1, there exists $\gamma = \sim c_1| \cdots | \sim c_n \in \Delta'$ such that $\Delta' \vdash_P c_j, j = 1, \ldots, n$. From $m > 1$, we have $P^+_{\Delta'} = P^+_{\Delta}$ and thus $\Delta \vdash_P c_j, j = 1, \ldots, n$. Since Δ is self-consistent, it has to be the case that $\gamma \in \| \alpha \|$. It follows that $\gamma \in \mathbf{A}_P(\Delta)$ from Corollary 2.2(1) and $\alpha \in \mathbf{A}_P(\Delta)$. Therefore, $\Delta \leadsto_P \Delta$, a contradiction. That is, Δ' is self-consistent. $\Box$

Remark
In this theorem, the condition $P^+_{\Delta'} = P^+_{\Delta}$ is guaranteed by the premise 'P is a disjunctive program'.

Lemma 4.2
If $\Delta' = \Delta \cup \|\sim a \|$ and $\sim a \in \mathbf{A}_P(\Delta) \setminus \Delta$, then $\Delta' \vdash_P a$ does not hold. $\Box$

Proof
To the contrary, suppose that $\Delta' \vdash_P a$. From $\sim a \in \mathbf{A}_P(\Delta)$, then it follows that $\Delta \leadsto_P \Delta'$. We distinct two cases:

Case 1. There exist negative literals $\sim b_1, \ldots, \sim b_m \in \Delta \cup \|\sim a \|$ such that $\Delta \vdash_P b_1| \cdots |b_m$: Since Δ is self-consistent, there is a number $i (1 \leq i \leq m)$ such that $\sim b_i = \sim a$. Without loss of generality, suppose that $\sim b_1, \ldots, b_{m-1} \in \Delta$ and $\sim b_m = \sim a$. Then, from $\Delta \vdash_P b_1| \cdots |b_{m-1}|a$, we have $\Delta \vdash_P a$ and $\sim a \in \mathbf{A}_P(\Delta)$. Hence $\Delta \leadsto_P \Delta$, a contradiction. This means that it can not be the case $\Delta' \vdash_P a$.

Case 2. It is not the Case 1: then there exists a disjunct $\beta = \sim b_1 \cdots | \sim b_m \in \Delta \cup \|\sim a \|$ such that $m > 1$ and $\Delta \vdash_P b_j, j = 1, \ldots, m$. Because $\beta \notin \| a \|$, $\beta \in \Delta$ and therefore $\Delta \leadsto_P \Delta$, also a contradiction. $\Box$

The proof of Theorem 4.3:

By Lemma 4.1, we consider only the case of $\alpha = \sim a$. Since $\|\sim a \| \subseteq \mathbf{A}_P(\Delta) \subseteq \mathbf{A}_P(\Delta')$, it is enough to prove that $\Delta' = \Delta \cup \|\sim a \|$ is self-consistent. To the contrary, suppose that $\Delta' \leadsto_P \Delta'$. Then there are two cases:

Case 1: There exists $\gamma = \sim c_1| \cdots | \sim c_n \in \Delta'$ such that $\Delta' \vdash_P c_i$, $i = 1, \ldots, n$:

Consider two subcases: (1) If $\gamma \in \Delta$, then $\Delta' \leadsto_P \Delta$. Since Δ is an ADH of P, it follows from Theorem 4.1 that $\Delta \leadsto_P \Delta'$, and hence $\Delta \vdash_P a$, a contradiction. (2) If $\gamma \notin \Delta$, that is, $\gamma \in \|\sim a \|$. Then there exists i $(1 \leq i \leq n)$ such that $\sim c_i = \sim a$. Thus $\Delta' \vdash_P a$, also a contradiction.

Case 2: There exist $\sim c_1, \ldots, \sim c_n \in \Delta'$ such that $\Delta' \vdash_P c_1| \cdots |c_n$: Because Δ is consistent, there exists $i (1 \leq i \leq n)$, such that $\sim c_i = \sim a$. Without loss of generality, suppose that $i = n$ and $\sim c_1, \ldots, \sim c_{n-1} \in \Delta$. From $\Delta' \vdash_P c_1| \cdots |c_{n-1}$ and $\sim c_1, \ldots, c_{n-1} \in \Delta$, we have $\Delta' \vdash_P a$, a contradiction. Thus Δ' is self-consistent. $\Box$

As a direct corollary of Theorem 4.1, it is not hard to see that an PDH of a disjunctive program must be a CDH.

Theorem 4.4

If P is a disjunctive program and Δ is an PDH, then Δ is also a CDH of P. $\square$

Proof

If Δ is an ADH of P, then it follows from Theorem 4.1 that $\mathbf{A}_P(\Delta)$ is also an ADH. Furthermore, from $\Delta \subseteq \mathbf{A}_P(\Delta)$ and the maximality of Δ, we have $\Delta = \mathbf{A}_P(\Delta)$. $\square$

In th rest of this section P will denote disjunctive if without state explicitly. $H(P)$ is the set of all disjunctive hypotheses of P and it can be easily verified that the partial order set $(H(P), \subseteq)$ is a complete lattice. From Definition 2.10, $\mathbf{A}_P$ can be considered as an operator on $H(P)$, called the admissible operator of P, and we will show that $\mathbf{A}_P$ is continuous.

Lemma 4.3

For any disjunctive program P, its admissible operator $\mathbf{A}_P : H(P) \to H(P)$ is continuous. That is, for any direct subset $\mathbf{D}$ of $H(P)$, the following holds:

$$\mathbf{A}_P(\cup\{\Delta : \Delta \in \mathbf{D}\}) = \cup\{\mathbf{A}_P(\Delta) : \ \Delta \in \mathbf{D}\}.$$

$\square$

Proof

For any $\Delta' \in \mathbf{D}$, then $\Delta' \subseteq \{\Delta : \Delta \in \mathbf{D}\}$. Furthermore, $\mathbf{A}_P(\Delta') \subseteq \mathbf{A}_P(\cup\{\Delta : \ \Delta \in \mathbf{D}\})$, and hence $\cup\{\mathbf{A}_P(\Delta) : \ \Delta \in \mathbf{D}\} \subseteq \mathbf{A}_P(\cup\{\Delta : \Delta \in \mathbf{D}\})$.

For the converse, suppose that $\alpha \in \mathbf{A}_P(\cup\{\Delta : \Delta \in \mathbf{D}\})$. If $\Delta' \leadsto \{\alpha\}$, write $\bar{\Delta} = \cup\{\Delta : \Delta \in \mathbf{D}\}$, then $\bar{\Delta} \leadsto_P \Delta'$. Similar to the proof of Theorem 4. 2, we can show that there exists $\Delta_0 \in \mathbf{D}$ such that $\Delta_0 \leadsto \Delta'$. Thus $\alpha \in \mathbf{A}_P(\Delta_0)$, that is, $\mathbf{A}_P(\cup\{\Delta : \Delta \in \mathbf{D}\}) \subseteq \cup\{\mathbf{A}_P(\Delta) : \ \Delta \in \mathbf{D}\}$.

It follows from Theorem 4.3 and Tarski's theorem [17] that $\mathbf{A}_P$ has the least fixpoint $lfp(\mathbf{A}_P)$ and $lfp(\mathbf{A}_P) = \mathbf{A}_P \uparrow \omega$, that is, the closure cardinal of $\mathbf{A}_P$ is ω. $\square$

Theorem 4.5

Every disjunctive program P possesses the unique well-founded disjunctive hypothesis (WFDH).

$\square$

S5 BDAS for Positive Disjunctive Programs

In this section we investigate the relationship between BDAS and some of semantics for positive disjunctive programs (without negation in the bodies of program clauses). In particular, we showed that the well-founded disjunctive hypotheses (WFDHs) provide a quite new characterization of EGCWA[20] by abduction, besides, WFDHs integrate and extend both the well-founded semantics for normal logic programs and EGCWA for positive disjunctive programs. As a result, EGCWA can be used to implement abductive reasoning.

If we do not explicitly stated, P will denote a positive disjunctive program in this section.

Theorem 5.1

If $ms(P)$ is the least model state of a positive disjunctive program P, then the state pair corresponding to the ADH $\emptyset$ is $S_\emptyset =< ms(P); \emptyset >$. $\square$

This theorem shows that the ADH $\emptyset$ characterizes the least model state.

Proof

It is a direct result of Definition 2.3. $\square$

Theorem 5.2

Let Δ be a hypothesis of positive disjunctive program P:

(1) If Δ is a PDH of P and Δ is consistent (i. e. the first-order formulas $V_P(\Delta) \cup \Delta$ is consistent), then $I_\Delta = B_P \setminus \{a \in B_P | \ \sim a \in \Delta\}$ is a minimal model of P;

(2) If I is a minimal model of P then $\Delta = \| \sim \bar{I} \ \|$ is a PDH of P, where $\bar{I} = B_P \setminus I$ and $\sim \bar{I} = \{\sim a | \ a \in \bar{I}\}$. $\square$

Before giving the proof of this theorem we first show the following lemmas.

Lemma 5.1
Let I be a model of P and $\Delta = \| \sim \bar{I} \|$. If $\Delta \vdash_P a$ then $a \in I$. $\Box$
Proof
If $\Delta \vdash_P a$, then there exists a disjunct of positive literals $a|a_1|\cdots|a_r \in ms(P), r \geq 0$, and $\sim a_1, \ldots, \sim a_r \in \Delta = \| \sim \bar{I} \|$, that is $a_1, \ldots, a_r \notin I$. Because $P \models ms(P)$, we have $I \models a|a_1|\cdots|a_r$ and thus $I \models a$. This means $a \in I$. $\Box$
Lemma 5.2
For any model I of P, $\Delta = \| \sim \bar{I} \|$ is an ADH of P. $\Box$
Proof
First, to show $\sim \bar{I} \subseteq \mathbf{A}_P(\Delta)$: Suppose that $\Delta' \vdash_P a$ and $\sim a \in \sim \bar{I}$. It follows from P being positive that $P^+_\Delta = P$, and thus there exists a disjunct $a|a_1|\cdots|a_r \in ms(P), r \geq 0$, such that $\sim a_1, \ldots, \sim a_r \in \Delta'$. Again, from $\sim a \in \Delta$, we have $\Delta \vdash_P a_1|\cdots|a_r$, thus $\Delta \leadsto_P \Delta'$. This means $\sim a \in \mathbf{A}_P(\Delta)$.

Next, to show that $\Delta = \| \sim \bar{I} \|$ is self-consistent: Suppose that Δ is not self-consistent, then there is $\sim b_1|\cdots|\sim b_m \in \Delta$ such that $\Delta \vdash_P b_1, \ldots, b_m$. It follows $b_1, \ldots, b_m \in I$ from Lemma 4.1. On the other hand, the fact that $\sim b_1|\cdots|\sim b_m \in \Delta$ implies $b_i \in \sim \bar{I}$ for some i $(1 \leq i \leq m)$, that is $b_i \notin I$, a contradiction. Hence $\Delta = \| \sim \bar{I} \|$ is self-consistent. $\Box$
Lemma 5.3
If Δ is an ADH of positive disjunctive program P and $\Delta \cup V_P(\Delta)$ is consistent, then there exists a subset A of B_P such that $\Delta = \| \sim A \|$, where $\sim A = \{\sim a : a \in A\}$. $\Box$
Proof
Because $V_P(\Delta) \cup \Delta$ is consistent, it has a minimal model M and hence M is also a minimal model of $P \cup \Delta$. For any $\sim c_1|\cdots|\sim c_n \in \Delta$, then $M \models \sim c_1|\cdots|\sim c_n$. That is, $M \models \sim c_i$ for some i $(1 \leq i \leq n)$, or $\sim c_1|\cdots|\sim c_n \in \| \sim \bar{M} \|$. Take $A = \bar{M}$, we get $\Delta = \| \sim A \|$ from the maximality of Δ. $\Box$

Proof of Theorem 5.2*:*

(1) From Lemma 5.3, it follows that $\Delta = \| \sim \bar{I}_\Delta \|$, where $I_\Delta = \{a \in B_P : \sim a \notin \Delta\}$. Because both P and $ms(P)$ has the same set of minimal models, it is enough to prove that I_Δ is a minimal model of $ms(P)$. First, to prove I_Δ is a model of $ms(P)$: Otherwise, there exists $\beta = b_1|\cdots|b_m \in ms(P)$ such that $I_\Delta \not\models b_1|\cdots|b_m$, that is, $\sim b_1, \ldots, \sim b_m \in \Delta$. From $\beta \in ms(P)$, we get $\Delta \vdash_P \beta$ and hence $\sim b_1 \in \Delta$. This implies $\Delta \leadsto_P \Delta$, a contradiction. Next, we prove I_Δ is minimal: If I' is a minimal model of $ms(P)$ such that $I' \subseteq I_\Delta$, it follows from Lemma 5.2 that $\Delta' = \| \sim \bar{I'} \|$ is an ADH of P. Obviously, $I' \subseteq I_\Delta$. By the maximality of Δ, we have $\Delta' = \Delta$, or $I' = I_\Delta$.

(2) It follows from Lemma 5.2 that Δ is an ADH of P. To the contrary, suppose that Δ is not a PDH, then there exists a PDH Δ' such that $\Delta' \subset \Delta$. By the above (1), $I_{\Delta'}$ is a minimal model of P and $I_{\Delta'} \subset I$, this contradicts the minimality of I. Hence Δ is a PDH. $\Box$

For any positive disjunctive program P, its WFDH does not only exist, but also can be obtained by one step iteration of $\mathbf{A}_P$ from $\emptyset$.
Theorem 5.3
Let P be a positive disjunctive program, then the closure ordinal of $\mathbf{A}_P$ is 1, that is, the (unique) WFDH of P is $\mathbf{A}_P(\emptyset)$. $\Box$

We first show a Lemma:
Lemma 5.4
For positive disjunctive program P, if $\mathbf{A}_P(\emptyset) \vdash_P b_1|\cdots|b_m$, $m \geq 1$, then $\emptyset \vdash_P b_1|\cdots|b_m$. $\Box$
Proof

By the assumption of this lemma, there is a disjunct $\beta = b_1|\cdots|b_m|b_{m+1}|\cdots|b_n \in can(ms(P))$ *such that* $n \geq m$ *and* $\sim b_{m+1}, \ldots, \sim b_n \in \mathbf{A}_P(\emptyset)$. *We can assert that* $n = m$*: Otherwise,* $n > m$*, set* $\Delta'' = \| \sim b_1, \ldots, \sim b_m, \sim b_{m+2}, \ldots, \sim b_n \|$*, then* $\Delta'' \vdash_P b_{m+1}$. *Again, from* $\beta \in ms(P)$*, we have* $b_1|\cdots|b_m|b_{m+2}|\cdots|b_n \notin can(ms(P))$. *Therefore,* $\emptyset \not\rightsquigarrow_P \Delta''$*, this contradicts* $\sim b_{m+1} \in \mathbf{A}_P(\emptyset)$.

The proof of Theorem 5.3*: It suffices to show* $\mathbf{A}_P(\mathbf{A}_P(\emptyset)) = \mathbf{A}_P(\emptyset)$. *From the monotonicity of* $\mathbf{A}_P$*, it follows easily that* $\mathbf{A}_P(\emptyset) \subseteq \mathbf{A}_P(\mathbf{A}_P(\emptyset))$. *For the converse, suppose that* $\alpha = \sim a_1|\cdots|\sim a_r \in \mathbf{A}_P(\mathbf{A}_P(\emptyset))$. *If* Δ' *is any hypothesis of* P *such that* $\Delta' \vdash_P a_i$, $(i = 1, \ldots, r)$*, then* $\mathbf{A}_P(\emptyset) \rightsquigarrow_P \Delta'$. *It is not hard to be seen that there exist* $\sim b_1, \ldots, \sim b_m \in \Delta'$ *such that* $\mathbf{A}_P(\emptyset) \rightsquigarrow_P b_1|\cdots|b_m$. *By Lemma 5.4,* $\emptyset \rightsquigarrow_P \Delta'$*, and hence* $\alpha \in \mathbf{A}_P(\emptyset)$. *That is,* $\mathbf{A}_P(\mathbf{A}_P(\emptyset)) \subseteq \mathbf{A}_P(\emptyset)$. □

To characterize EGCWA in BDAS, we first give the model-theoretic definition of EGCWA.

Definition 5.1[20]

$Let P be a positive disjunctive program, then EGCWA(P) = \{\beta \in DB_P^- : P \models_{min} \beta\}$

The following theorem shows that $EGCWA$ *coincides with* $WFDH$ *for the class of positive disjunctive programs.*

Theorem 5.4

For positive disjunctive program P, $EGCWA(P) = WFDH(P)$. □

Now we prepare to prove this theorem.

Lemma 5.5

If $can(ms(P))$ *is cosidered as a positive disjunctive program, then, for any* $\Delta \in H(P)$ *and* $\alpha \in DB_P^+$,

$$\Delta \vdash_P \alpha \text{ if and only if } \Delta \vdash_{can(ms(P))} \alpha.$$

□

Proof

It is easy to be seen from Definition 2.3. □

Lemma 5.6

Let $can(ms(P)) = \{\alpha_1, \alpha_2, \ldots, \alpha_n, \ldots\}$*, take* $a_1 \in atoms(\alpha_1)$ *and* $M_1 = \{a_1\}$. *Suppose that* M_n *is well-defined, set*

$$M_{n+1} = \begin{cases} M_n, & \text{if } M_n \models \alpha_{n+1}, \\ M_n \cup \{a_{n+1}\}, & \text{if } M_n \not\models \alpha_{n+1}, \text{ and take } a_{n+1} \in atoms(\alpha_{n+1}). \end{cases}$$

Then $M = \cup_{n=1}^{\infty} M_n$ *is a minimal model of* $can(ms(P))$. □

Proof

It follows easily from the definition of M. □

Proof of Theorem 5.4:

First, to prove $EGCWA(P) \subseteq WFDH(P)$*: To the contrary, suppose that there exists* $\beta = \sim b_1|\cdots|\sim b_m \notin WF(P) = \mathbf{A}_P(\emptyset)$ $(b_1, \ldots, b_m$ *are distinct from each other). Then, for some hypothesis* Δ' *of* P, $\alpha' \vdash_P b_i (i = 1, \ldots, m)$*, but* $\emptyset \rightsquigarrow_P \alpha'$. *Hence there exist* $\alpha_1, \ldots, \alpha_m \in can(ms(P))$ *such that*

$$\alpha_1 = b_1|b_{11}|\cdots|b_{1t_1},\ t_1 \geq 0,$$

$$\cdots\cdots$$

$$\alpha_m = b_m|b_{m1}|\cdots|b_{mt_m},\ t_m \geq 0,$$

and $\sim b_{ij} \in \Delta'$, $i = 1, \ldots, m; j = 1, \ldots, t_i$.

We assert that Δ' is self-consistent. Otherwise, i.e. $\Delta' \rightsquigarrow_P \Delta'$, then there exist $\sim a_1, \ldots, \sim a_r \in \Delta'$ and $\Delta' \vdash_P a_1|\cdots|a_r$. Hence there is a disjunct $a_1|\cdots|a_r|a_{r+1}|\cdots|a_s \in can(ms(P))$ such that $\sim a_{r+1}, \ldots, \sim a_s \in \Delta'$. This implies $\emptyset \vdash_P a_1|\ldots|a_s$. From $\sim a_1, \ldots, \sim a_r \in \Delta'$, it follows that $\emptyset \rightsquigarrow_P \Delta'$, a contradiction.

Since Δ' is self-consistent, no clause $b_{ij} \leftarrow$ is in P, that is, $b_{ij} \notin can(ms(P))$, and b_j does not appear in α_i for any $i \neq j$. Construct a minimal model of $can(ms(P))$ as follows:

Let $can(ms(P)) = \{\alpha_1, \ldots, \alpha_m, \alpha_{m+1}, \ldots\}$, and take $M_1 = \{b_1\}$. Then $M_1 \not\models \alpha_2$, we can set $M_2 = M_1 \cup \{b_2\}$. Similarly, $M_{m-1} = \{b_1, \ldots, b_{m-1}\}$ and $M_{m-1} \not\models \alpha_m$. Set $M_m = \{b_1, \ldots, b_m\}$. Following the method in Lemma 5.6, we get a sequence of models $\{M_n\}_{n=1}^{\infty}$. Then $M = \cup_{n=1}^{\infty} M_n$ is a minimal model of $can(ms(P))$ and therefore M is also a minimal model of P, but $b_1, \ldots, b_m \notin M$, this means $\sim b_1|\cdots|\sim b_m \notin EGCWA(P)$.

Next, to show $WFDH(P) \subseteq EGCWA(P)$: If $\sim b_1|\cdots|\sim b_m \notin EGCWA(P)$, then there is a model M of P such that $\{b_1, \ldots, b_m\} \subseteq M$. For any i $(1 \leq i \leq m)$, there must be a $\alpha_i \in can(ms(P))$ such that $\alpha_i = b_i|b_{i1}|\cdots|b_{it_i}$, $t_i \geq 0$, and we can choose α_i with the property that $atoms(\alpha_i) \cap M$ is minimal. Set $\Delta' = \| \cup_{i=1}^{m} \cup_{j=1}^{t_i} \{\sim b_{ij}\} \|$, then

(1) $\Delta' \vdash_P b_i$ $(i = 1, \ldots, m)$;

(2) Δ' is a self-consistent hypothesis of P;

(3) $\emptyset \rightsquigarrow_P \Delta'$.

It is easy to see (1) holds. We now show (2): Suppose that $\Delta' \rightsquigarrow_P \Delta'$. Since Δ' is generated only by negative literals, there must exists a $b'_1 \in \Delta'$ such that $\Delta' \vdash_P b'_1$, and hence there is a disjunct of atoms $b'_1|b'_2|\cdots|b'_s \in can(ms(P))$ satisfying $\sim b'_2, \ldots, \sim b'_s \in \Delta'$, $s \geq 1$. Because $M \models b'_1|b'_2|\cdots|b'_s$, there is some $b'_i \in M$, $1 \leq i \leq s$. Again, $\{\sim b'_1, \ldots, \sim b'_s\} \subseteq \Delta'$, then, for some j, k $(1 \leq j \leq m;\ 1 \leq k \leq t_j)$, $b'_i = b_{jk}$. This implies that α_j contains at least two atoms of M (b_j and b_{jk}). If $\alpha \in can(ms(P))$ and b_j appears in α, then α contains at least two atoms of M. Hence $M \setminus \{b_j\}$ is also a model of $can(ms(P))$, this contradicts the minimality of M. So Δ' is self-consistent.

(3) is an immediate result of (2).

From (1) and (3), $\beta = \sim b_1|\cdots|\sim b_m \notin \mathbf{A}_P(\emptyset) = WFDH(P)$. $\Box$

The following corollaries are directly followed from Theorem 5.6 and the results in Ref.14).

Corollary 5.1

For any positive disjunctive program P, its WFDH is consistent. $\Box$

The generalized closed world assumption (GCWA) can also be characterized by WFDH.

Corollary 5.2

$GCWA(P) = L(WFDH(P)) = \{\sim a : \sim a \in WFDH(P)\}$. $\Box$

Corollary 5.3

M is a minimal model of P if and only if M is a minimal model of $P \cup WFDH(P)$. $\Box$

6 BDAS for General Disjunctive Programs

In this section we will study the relationship between BDAS and the (disjunctive) stable semantics for general disjunctive programs. An extensive class of disjunctive programs, called the strongly stable programs, are introduced, for which we show that PDHs and stable models have a one-to-one correspondence. Hence the abductive semantics PDH is not only complete but also extends the disjunctive stable semantics.

6.1 The least fixpoint transformation

To simplify the proof of the results in this section, we first give a program transformation for bi-disjunctive programs, called the least fixpoint transformation, which can not only make *our proofs simpler but also provides a connocial form for bi-disjunctive programs with respect to BDAS and stable semantics.*

Let GDB_P be the set of all negative bi-disjunctive programs whose atoms are in B_P:

$$GDB_P = \{a_1|\cdots|a_r \leftarrow \beta_1, \ldots, \beta_s : \ a_i \in B_P, \beta_j \in DB_P^-, i = 1, \ldots, r; j = 1, \ldots, s\}$$

and $\leftarrow$ the empty clause.

Definition 6.1

For any bi-disjunctive program P, the generalized consequence operator $T_P^G : 2^{GDB_P} \rightarrow 2^{GDB_P}$ is defined as, for any $J \subseteq GDB_P$,

$$T_P^G(J) = \{C \in GDB_P : \text{ there exists a bi-disjunctive clause } \alpha' \leftarrow b_1, \ldots, b_m, \beta_1, \ldots, \beta_s \text{ and}$$

$$C_1, \ldots, C_m \in GDB_P \cup \{\leftarrow\} \text{ such that } (1)\ b_i|head(C_i) \leftarrow body(C_i) \text{ is in } J, i = 1, \ldots, m;$$

$$(2)\ A \text{ is the clause } can(\alpha'|head(C_1)|\cdots|head(C_m)) \leftarrow \beta_1, \ldots, \beta_s, body(C_1), \ldots, body(C_m)\}.$$

□

Example 6.1

Suppose that $P = \{a_1|a_2 \leftarrow a_3, \sim a_4;\ a_3|a_5 \leftarrow \sim a_6\}$ and $J = T_P^G(\emptyset)$. Then $T_P^G(\emptyset) = \{a_3|a_5 \leftarrow \sim a_6\}$; If $J' = T_P^G(T_P^G(\emptyset))$. Then $T_P^G(J') = T_P^G(T_P^G(\emptyset)) = \{a_3|a_5 \leftarrow \sim a_6;\ a_1|a_2|a_5 \leftarrow \sim a_4, \sim a_6\}$.
□

Theorem 6.1

For any bi-disjunctive program P, its generalized consequence operator T_P^G is continuous and hence possesses the least fixpoint $lfp(T_P^G) = T_P^G \uparrow \omega$. □

Proof

Similar to the corresponding result of T_P^S, see Ref.11). □

Definition 6.2

Denote $lfp(T_P^G)$ as $Lft(P)$, then $Lft:\ P \rightarrow Lft(P)$ defines a transformation from the set of bi-disjunctive programs to the set of negative bi-disjunctive programs, and we call $Lft(P)$ the least fixpoint transformation of P. □

Before showing that Lft preserves BDAS and stable semantics, we first state the following lemma.

Lemma 6.1

Let P be a bi-disjunctive program, and $\Delta \subseteq DB_P^-$, then $(Lft(P)_\Delta^+)$ has the same least model state as P_Δ^+:

$$ms(Lft(P)_\Delta^+) = ms(P_\Delta^+).$$

□

Proof

First, to show $ms(Lft(P)_\Delta^+) \subseteq ms(P_\Delta^+)$: If $\alpha = a_1|\cdots|a_r \in can(ms(Lft(P)_\Delta^+))$, since $Lft(P)$ is a negative bi-disjunctive program, then the trivial clause $a_1|\cdots|a_r \leftarrow$ is in $Lft(P)_\Delta^+$ and hence there exists a negative clause $C' :\ a_1|\cdots|a_r \leftarrow \beta_1, \ldots, \beta_s$ belonging to $Lft(P)$ such that $\beta_1, \ldots, \beta_s \in \Delta$. Because $Lft(P) = T_P^G \uparrow \omega$, using induction on k, we prove that $a_1|\cdots|a_r \in ms(Lft(P)_\Delta^+)$ holds for any C' in $T_P^G \uparrow k$: $a_1|\cdots|a_r \leftarrow \beta_1, \ldots, \beta_s$.

If $C' \in T_P^G \uparrow 1$, then the clause $a_1|\cdots|a_r \leftarrow$ is in P. This means $a_1|\cdots|a_r \leftarrow$ in P_Δ^+ and therefore $a_1|\cdots|a_r \in ms(P_\Delta^+)$.

Assume that $C' \in T_P^+ G \uparrow k$ implies $a_1|\cdots|a_r \in ms(P_\Delta^+)$. If $C' \in T_P^G \uparrow k+1 = T_P^G(T_P^G \uparrow k)$, then there is a clause $C :\ \alpha' \leftarrow b_1, \ldots, b_m, \beta_1', \ldots, \beta_n'$ in P and $C_1', \ldots C_m' \in GDB_P \cup \{\leftarrow\}$ satisfying the following two conditions:

(1) For any $i = 1, \ldots, m$, the clause $b_i|head(C_i') \leftarrow body(C_i')$ is in $T_P^G \uparrow k$;

(2) C is the clause $can(\alpha'|head(C_1')|\cdots|head(C_m')) \leftarrow \beta_1', \ldots, \beta_n', body(C_1'), \ldots, body(C_m')$.

Since $\{\beta_1' \ldots, \beta_n'\} \subseteq body(C') \subseteq \Delta$, corresponding to C, the positive clause $C_\Delta^+ : \alpha' \leftarrow b_1, \ldots, b_m$ is in P_Δ^+. By the induction assumption, we have $b_i|head(C_i') \in ms(P_\Delta^+)$ for $i = 1, \ldots, m$. Again, from (2), $a_1|\ldots|a_r = can(\alpha'|head(C_1')|\cdots|head(C_m')) \in T_P^S(ms(P_\Delta^+)) \subseteq ms(P_\Delta^+)$. Hence $can(ms(Lft(P)_\Delta^+)) \subseteq ms(P_\Delta^+)$, that is, $ms(Lft(P)_\Delta^+) \subseteq ms(P_\Delta^+)$.

Next, to prove $ms(P_\Delta^+) \subseteq ms(Lft(P)_\Delta^+)$: Since $ms(P_\Delta^+) = \| T_{P_\Delta^+}^S \uparrow \omega \|$, using induction on k, it suffices to prove that $T_{P_\Delta^+}^S \uparrow k \subseteq ms(Lft(P)_\Delta^+)$. $\qquad (\star)$

If $k = 0$, $T_{P_\Delta^+}^S \uparrow 0 = \emptyset$, $(\star)$ is trivial.

Assume that $(\star)$ holds for k. We need to show $T_{P_\Delta^+}^S \uparrow k+1 \subseteq ms(Lft(P)_\Delta^+)$. In fact, if $a_1|\cdots|a_r \in T_{P_\Delta^+}^S \uparrow k+1 = T_{P_\Delta^+}^S(T_{P_\Delta^+}^S \uparrow k)$, then there is a clause C_Δ^+ of P_Δ^+: $\alpha' \leftarrow b_1, \ldots, b_m$, and $\alpha_1, \ldots, \alpha_m \in DB_P^+ \cup \{\leftarrow\}$ such that the following two conditions are satisfied:

(1) $b_i|\alpha_i \in T_{P_\Delta^+}^S \uparrow k$, $i = 1, \ldots, m$; and

(2) $a_1|\cdots|a_r = can(\alpha'|\alpha_1|\cdots|\alpha_m)$.

By induction assumption, $b_i|\alpha_i \in ms(Lft(P)_\Delta^+)$. Corresponding to $C_\Delta^+ \in P_\Delta^+$, there is a clause $\alpha' \leftarrow b_1, \ldots, b_m, \beta_1, \ldots, \beta_s$ in P such that $\beta_1, \ldots, \beta_s \in \Delta$. Therefore the clauses $b_i|\alpha_i \leftarrow$ is in $Lft(P)_\Delta^+$, for $i = 1, \ldots, m$. This implies that there are clauses $b_i|\alpha_i \leftarrow \beta_{i1}, \ldots, \beta_{it_i}$ in $Lft(P)$ such that $\beta_{i1}, \ldots, \beta_{it_i} \in \Delta$. By the definition of T_P^G, we have $\alpha'|\alpha_1|\cdots|\alpha_m \leftarrow \beta_1, \ldots, \beta_s, \beta_{11}, \ldots, \beta_{1t_1}, \ldots, \beta_{m1}, \ldots, \beta_{mt_m}$ is in $T_P^G(Lft(P)) = Lft(P)$. It follows from $body(C') \subseteq \Delta$ that $a_1|\cdots|a_r \leftarrow$ belongs to $Lft(P)_\Delta^+$, or $a_1|\cdots|a_r \in ms(Lft(P)_\Delta^+)$. Therefore, $ms(P_\Delta^+) \subseteq ms(Lft(P)_\Delta^+)$. $\square$

By the Lemma 6.1, one of the main results in this subsection will be obtained easily.

Theorem 6.2

For any bi-disjunctive program P, P is equivalent to its least fixpoint transformation $Lft(P)$ with respect to BDAS. As a result, P has the same ADH (res. CDH, PDH) as $Lft(P)$. $\square$

Proof

By Lemma 6.1, it follows that, for any $\alpha \in DB_P^+$ and $\Delta \subseteq DB_P^-$,

$$\Delta \vdash_P \alpha \text{ if and only if } \Delta \vdash_{Lft(P)} \alpha.$$

Therefore the theorem is true. $\square$

The following theorem shows that the least fixpoint transformation also preserves the (disjunctive) stable models.

For any disjunctive program P, and $M \subseteq B_P$. Set

$$P/M = \{a_1|\cdots|a_r \leftarrow a_{r+1}, \ldots, a_s : \text{ there exists a clause of } P: \ a_1|\cdots|a_r \leftarrow a_{r+1}, \ldots, a_s,$$
$$\sim a_{s+1}, \ldots, \sim a_t\}.$$

If M is a minimal model of P/M, then it is called a (disjunctive) stable model of P. The stable semantics of P is defined as the set of its all stable models.

Theorem 6.3

For any disjunctive program P, P is equivalent to its least fixpoint transformation $Lft(P)$ with respect to the stable semantics, that is, P has the same set of the stable models as $Lft(P)$. $\square$

Proof

Let $M \subseteq B_P$ and $\Delta_M = \| \{\sim a : \ a \in B_P \setminus M\} \|$, then $P/M = P_{\Delta_M}^+$. By Lemma 6.1, P/M and $Lft(P/M)$ have the same least model state and hence have the same set of minimal models. Again, $Lft(P/M) = Lft(P)/M$. Therefore,

M is a stable model of P if and only if
M is a minimal model of P/M if and only if
M is a minimal model of $Lft(P)/M$ if and only if
M is a stable model of $Lft(P)$. $\square$

6.2 PDH for the strongly stable programs

Let P be a disjunctive program and $\Delta \subseteq DB_P^-$, P_Δ^* is defined as the disjunctive program obtained by the following transformations:

(1) For any clause C in P, if $a \in head(C)$ and $\sim a \in \Delta$, then delete a from the head of C; if $\sim b \in body(C)$ and $\sim b \in \Delta$, then delete $\sim b$ from the body of C;

(2) From the program obtained by (1), delete all the clauses that have empty heads.

Definition 6.3

A disjunctive program P is strongly stable if, for any $\Delta \subseteq DB_P^-$, P_Δ^* possesses at least one stable model. $\square$

It is obvious that positive disjunctive programs are strongly stable. More generally, the class of (local) stratified disjunctive programs are strongly stable.

The main theorem of this section is as follows:

Theorem 6.4

Suppose that disjunctive program P is strongly stable and its all PDHs are consistent:

(1) If Δ is a PDH of P, then $I_\Delta = \{a \in B_P : \sim a \notin \Delta\}$ is a stable model of P.

(2) If M is a stable model of P, then $\Delta_M = \| \{\sim a :\ a \in B_P \setminus M\} \|$ is a PDH of P. $\square$

As pointed in Section 4, we believe the condition that "the PDHs of P are consistent" is unnecessary.

Proof

By Theorem 6.2 and Theorem 6.3, it suffices to prove this theorem only for the class of negative disjunctive programs.

(1) Suppose that M is a stable model of P_Δ^*, we assert $\Delta \subseteq \Delta_M$: Otherwise, if $\Delta \not\subseteq \Delta_M$, then there is $\beta =\sim b_1|\cdots|\sim b_m \in \Delta \setminus \Delta_M$, that is, $b_1, \ldots, b_m \in M$. Since M is a minimal model of P_Δ^*, there exist clauses C_i' of P_Δ^*: $b_i|b_{i1}|\cdots|b_{it_i} \leftarrow$ such that $\sim b_{i1}, \ldots, \sim b_{it_i} \in \Delta_M$, $t_i \geq 0$, $i = 1, \ldots, m$. This implies there are clauses of P_Δ^*: $C_i:\ b_i|b_{i1}|\cdots|b_{it_i} \leftarrow\sim a_{i1}, \ldots, \sim a_{is_i}$ such that $\sim a_{i1}, \ldots, \sim a_{is_i} \in \Delta_M$, and thus $\Delta_M \vdash_{P_\Delta^*} b_i$, $i = 1, \ldots, m$. From $\beta =\sim b_1|\cdots|\sim b_m \in \Delta$, it follows that $\Delta_M \leadsto_{P_\Delta^*} \Delta$, that is, $\Delta_M \cup \Delta \leadsto_P \Delta$. Hence $\Delta \leadsto_P \Delta_M \cup \Delta$. Δ being self-consistent implies $\Delta \leadsto_P \Delta_M$. Therefore, there are $\sim c_1, \ldots, \sim c_n \in \Delta_M$ such that $\Delta \vdash_P c_1|\cdots|c_n$. Again from Δ being self-consistent, there exists i, $1 \leq i \leq n$, such that $\sim c_i \notin \Delta$. Without loss of generality, we can suppose that $\sim c_1, \ldots, \sim c_r \in \Delta_M \setminus \Delta$, but $\sim c_{r+1}, \ldots, \sim c_n \in \Delta$, $r \geq 1$. From $\Delta \vdash_P c_1|\cdots|c_n$, it follows that there is a clause C of P: $c_1|\cdots|c_r|c_{r+1}|\cdots|c_n|c_{n+1}|\cdots|c_u \leftarrow\sim c_{u+1}, \ldots, \sim c_v$ such that $\sim c_{n+1}, \ldots, \sim c_u, \sim c_{u+1}, \ldots, \sim c_v \in \Delta$, $u \geq v \geq n \geq n \geq r \geq 1$. This means the clause $c_1|\cdots|c_r \leftarrow$ is in P_Δ^*. Since $M \models P_\Delta^*$, there exists j, $1 \leq j \leq r$ such that $c_j \in M$, i. e. $\sim c_j \notin \Delta_M$, a contradiction. Therefore $\Delta \subseteq \Delta_M$. By the maximality of Δ, $\Delta = \Delta_M$. That is, $I_\Delta = M$ is a stable model of P.

(2) To the contrary, suppose that Δ_M is not self-consistent, i. e. $\Delta_M \leadsto_P \Delta_M$. Since $can(\Delta_M)$ consists of only negative literals, there are $\sim a_1, \ldots, \sim a_r \in \Delta_m$ such that $a_1|\cdots|a_r \in ms(P_{\Delta_M}^+)$, this contradicts to $M \models P_{\Delta_M}^*$. Therefore, Δ_M is self-consistent. Next, to show that $\Delta_M \subseteq \mathbf{A}_P(\Delta_M)$: If $\sim a \in \Delta_M$, then $a \notin M$. For any hypothesis Δ' of P such that $\Delta' \vdash_P a$, there is a clause C of P: $a|a_1|\cdots|a_r \leftarrow\sim b_1, \ldots, \sim b_m$ satisfying $\sim a_i, \sim b_j \in \Delta'$, $i = 1, \ldots, r; j = 1, \ldots, m$. Because $a \notin M$ and $M \models C$, there are two possible cases: (1) $b_j \in M$ for some j, $1 \leq j \leq m$; (2) $a_i \in M$ for some i, $1 \leq i \leq r$. Thus, in any case, there is $c \in M$ but $\sim c \in \Delta'$. Since M is a minimal model of $P/M = P_{\Delta_M}^+$, there is a clause of

$P^+_{\Delta_M}$: $c|c_1\cdots|c_n \leftarrow\sim c_1\ldots,c_n$ in Δ_M. Hence $\Delta_M \vdash_P c$, this implies $\Delta_M \leadsto_P \Delta'$. That is, $\sim a \in \mathbf{A}_P(\Delta_M)$, or $can(\Delta_M) \subseteq \mathbf{A}_P(\Delta_M)$. Thus $\Delta_M \subseteq \mathbf{A}_P(\Delta_M)$.

Last, to show that Δ_M is maximal: Let Δ be a PDH of P such that $\Delta_M \subseteq \Delta$. Then, by (1) just proved above, I_Δ is a stable model of P and $I_\Delta \subseteq M$. Therefore $P/M \subseteq P/I_\Delta = P^+_\Delta$. Since I_Δ is a minimal model of P/I_Δ, it is also a model of P/M. Again, M is a minimal model of P/M, then $I_\Delta = M$, or $\Delta = \Delta_M$. Thus, we have got the conclusion that Δ_M is a PDH of P. $\square$

Corollary 6.1

Any (local) stratified disjunctive program P has the unique PDH. $\square$

Proof

Since a stratified disjunctive program P must be strongly stable, from Theorem 6.4, the PDH of P is determined by its unique stable model (or perfect model).

7 Conclusion

It is well-known that nondeterminism is a considerably desirable property for logic languages. Besides enhanced modeling capabilities, nondeterministic languages tend to have more expressive power. Existing declarative semantics for disjunctive logic programming (DLP) does not provide satisfactory support for nondeterminism. On the other hand, the goal of introducing DLP is to fully represent incomplete information in logic programming, but, in our opinion, most of the existing disjunctive semantics do not obtain this goal in that they fail to assign an intuitive meaning for the so-called disjunctive information. In this paper, we first generalized the syntax of DLP to bi-disjunctive logic programs and then proposed an argumentation-theoretic framework BDAS for bi-disjunctive programs. As a nondeterministic disjunctive semantics, BDAS is not only more expressive than ordinary disjunctive semantics, but also integrates many existing semantics into a unifying abductive framework. Another feature of BDAS is its completeness (or, robust), that is, BDAS is defined for all disjunctive programs.

References

[1] Brewka,G.,“An abductive framework for generalized logic programs,”in Proceedings of the 2nd Workshop on Logic Programming and Nonmonotonic Reasoning (Marek, W. and Subrahmanian,V. eds.), MIT Press, pp. 266-282, 1993.

[2] Dung,P., “Negation as hypothesis: an abductive foundation to logic programming,”in Proceedings of the 8th International Conference on Logic Programming, MIT Press, pp.3-17, 1991.

[3] Dung,P., “An argumentation-theoretic foundation for logic programming,”J. Logic Programming, 24, pp.151-177, 1995.

[4] Eshghi,K. and Kowalski,R., “Abduction compared with negation by failure,”in Proceedings of the 6th International Conference on Logic Programming, MIT Press, pp.234-255, 1989.

[5] Gelfond,M. and Lifschitz,J., “The stable model semantics for logic programming,”in Proceedings of the 5th Symposium on Logic Programming, MIT Press, pp.1070-1080,1988.

[6] van Gelder,A., Ross,K. and Schlipf,J., “Unfounded sets and well-founded semantics for general logic programs,”in Proceedings of the 7th ACM Symposium on Principles Of Database Systems, pp.221-230,1988. Full version in J. ACM, 38, pp. 620-650,1992.

[7] Kakas,A., Kowalski,R. and Toni,F., “Abductive logic programming,”J. Logic and Computation, 2, pp.719-770,1992.

[8] *Kakas,A. and Mancarella,P., "Generalized stable models: a semantics for abduction,"in Proceedings of the 9th European Conference Artificial Intelligence, pp.385-391, 1990.*

[9] *Kakas,A. and Mancarella,P., "Negation as stable hypotheses,"in Proceedings of the 1st Workshop on Logic Programming and Nonmonotonic Reasoning (Marek, W. and Subrahmanian,V. eds.), MIT Press, pp. 275-288, 1991.*

[10] *Lifschitz,V. and Turner,H., "From disjunctive programs to abduction,"in Proceedings of the Workshop on Nonmonotonic Extensions of Logic Programming, (Dix,J., Pereira,L. and Przymunski,T. eds.), pp. 111-125, 1994.*

[11] *Lobo,J., Minker,J. and Rajasekar,A., Foundations of Disjunctive Logic Programming, MIT Press, 1992.*

[12] *Minker,J., "Overview of disjunctive logic programming,"Ann. Math. AI., 12, pp.1-24, 1990.*

[13] *Przymunski,T., "Stable semantics for disjunctive programs,"New Generation Computing, 9, pp.401-424, 1991.*

[14] *Przymunski,T., "Stationary semantics for disjunctive logic programs and deductive databases,"in Proceedings of the North American Conference on Logic Programming (Debray,S. and Hemenegildo,M. eds.), MIT Press, pp. 42-59,1991.*

[15] *Przymunski,T., "Static semantics of logic programs,"Ann. Math. AI., 14, 323-357, 1995.*

[16] *Sakama,C. and Inoue,K., "On the equivalence between disjunctive and abductive logic programming,"Proceedings of the 11th International Conference on Logic Programming (Van Hentenryck ed.), MIT Press, pp.489-503, 1994.*

[17] *Tarski,A., "A lattice-theoretic fixpoint theorem and its applications,"Pacific J. Math., 5, pp.285-309, 1955.*

[18] *Torres,A., "Negation as failure to support,"in Proceedings of the 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning (Marek, W. and Subrahmanian,V. eds.), MIT Press, pp.223-243, 1993.*

[19] *Torres,A., "A nondeterministic semantics,"J. Math. AI., 14, pp.37-73, 1995.*

[20] *Yahya,A. and Henschen,L., "Deduction in non-Horn databases,"J. Automated Reasoning, 1, pp.141-160, 1985.*