

Comparing Disjunctive Well-founded Semantics^{*}

Matthias Knorr¹ and Pascal Hitzler²

¹ CENTRIA, Universidade Nova de Lisboa, Portugal

² AIFB, Universität Karlsruhe, Germany

Abstract. While the stable model semantics, in the form of Answer Set Programming, has become a successful semantics for disjunctive logic programs, a corresponding satisfactory extension of the well-founded semantics to disjunctive programs remains to be found. The many current proposals for such an extension are so diverse, that even a systematic comparison between them is a challenging task. In order to aid the quest for suitable disjunctive well-founded semantics, we present a systematic approach to a comparison based on level mappings, a recently introduced framework for characterizing logic programming semantics, which was quite successfully used for comparing the major semantics for normal logic programs. We extend this framework to disjunctive logic programs, which will allow us to gain comparative insights into their different handling of negation. Additionally, we show some of the problems occurring when trying to handle minimal models (and thus disjunctive stable models) within the framework.

1 Introduction

Two semantics are nowadays considered to be the most important ones for normal logic programs. Stable model semantics [1] is the main two-valued approach whereas the major three-valued semantics is the well-founded semantics [2]. These two semantics are well-known to be closely related. However, enriching normal logic programs with indefinite information by allowing disjunctions in the head³ of the clauses separates these two approaches. While disjunctive stable models [5] are a straightforward extension of the stable model semantics, the issue of disjunctive well-founded semantics remains unresolved, although several proposals exist.

Even a comparison of existing proposals is difficult due to the large variety of completely different constructions on which these semantics are based. In [6], Ross introduced the strong well-founded semantics (SWFS) based on a top-down procedure using derivation trees. The generalized disjunctive well-founded

^{*} This research has been partly funded by the European Commission within the 6th Framework Programme projects REVERSE number 506779 (cf. <http://reverse.net/>) and NeOn (IST-2006-027595, cf. <http://www.neon-project.org/>), and by the Deutsche Forschungsgemeinschaft under project ReaSem.

³ For an overview of semantics for disjunctive logic programs we refer to [3] and [4].

semantics (GDWFS) was defined by Baral, Lobo, and Minker in [7], built on several bottom-up operators and the extended generalized closed world assumption [8]. Brass and Dix proposed the disjunctive well-founded semantics (D-WFS) in [9] based on two operators iterating over conditional facts, respectively some general program transformations.

In order to allow for easier comparison of different semantics, a methodology has recently been proposed for uniformly characterizing semantics by means of level mappings, which allow for describing syntactic and semantic dependencies in logic programs [10]. This results in characterizations providing easy comparisons of the corresponding semantics.

In this paper, we attempt to utilize this approach and present level mapping characterizations for the three previously mentioned semantics, namely SWFS, GDWFS and D-WFS. The obtained uniform characterizations will allow us to compare the semantics in a new and more structured way. It turns out, however, that even under the uniform level-mapping characterizations the three semantics differ widely, such that there is simply not enough resemblance between the approaches to obtain a coherent picture. We can thus, basically, only confirm in a more formal way what has been known beforehand, namely that the issue of a good definition of well-founded semantics for disjunctive logic programs remains widely open. We still believe that our approach delivers structural insights which can help to guide the quest.

The paper is structured as follows. In Section 2, basic notions are presented and we recall shortly the well-founded semantics. Then we devote one section to each of the three semantics recalling the approach itself and presenting the level mapping characterization. We start with SWFS in Section 3, continue with GDWFS in Section 4 and end with D-WFS in Section 5. After that, in Section 6 we compare the characterizations looking for common conditions which might be properties for an appropriate well-founded semantics for disjunctive programs, and consider some of the difficulties occurring when applying the framework to minimal models. We conclude with Section 7.

The formal proofs required for the level-mapping characterizations of the semantics reported in this paper are very involved and technical. Due to space limitations, it was obviously not possible to include them. They can be found in the publicly available Technical Report [11].

2 General Notions and Preliminaries

A *disjunctive logic program* Π consists of finitely many universally quantified *clauses* of the form $H_1 \vee \dots \vee H_l \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ where H_k , A_i , and B_j , for $k = 1, \dots, l$, $i = 1, \dots, n$, and $j = 1, \dots, m$, are atoms of a given first order language, consisting of predicate symbols, function symbols, constants and variables. The symbol \neg is representing default negation. A clause c can be divided into the *head* $H_1 \vee \dots \vee H_l$ and the *body* $A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$. If the body is empty then c is called a *fact*. We also abbreviate c by $H \leftarrow A, \neg B$, where H , A and B are sets of pairwise distinct atoms and, likewise, we sometimes

handle disjunctions D and conjunctions C as sets. A *normal (definite)* clause contains exactly one atom in H (no atom in B) and we call a program consisting only of normal (definite) clauses a *normal (definite)* logic program. We denote normal programs by P to distinguish from disjunctive ones represented by Π . Any expression is called *ground* if it contains no variables. The *Herbrand base* B_Π is the set of all ground atoms that can be formed by using the given language from Π . A *literal* is either a *positive literal*, respectively an atom, or a *negative literal*, a negated atom, and usually we denote by A, B, \dots atoms and by L, M, \dots literals. Moreover, a *disjunction literal* is a disjunction or a negated disjunction. The *extended Herbrand base* EB_Π (*conjunctive Herbrand base* CB_Π) is the set of all disjunctions (conjunctions) that can be formed using pairwise distinct atoms from B_Π . Finally, $\mathbf{ground}(\Pi)$ is the set of all ground instances of clauses in Π with respect to B_Π .

We continue by recalling three-valued semantics based on the truth values true (**t**), undefined (**u**), and false (**f**). A (*partial*) *three-valued interpretation* I of a normal program P is a set $A \cup \neg B$, for $A, B \subseteq B_P$ and $A \cap B = \emptyset$, where elements in A, B respectively, are **t**, **f**, and the remaining wrt. B_P are **u**. The set of three-valued interpretations is denoted by $I_{P,3}$. Given a three-valued interpretation I , the body of a ground clause $H \leftarrow L_1, \dots, L_n$ is true in I if and only if $L_i \in I$, $1 \leq i \leq n$, or false in I if and only if $L_i \notin I$ for some i , $1 \leq i \leq n$. Otherwise the body is undefined. The ground clause $H \leftarrow \mathbf{body}$ is true in I if and only if the head H is true in I or \mathbf{body} is false in I or \mathbf{body} is undefined and H is not false in I . Moreover, I is a *three-valued model* for P if and only if all clauses in $\mathbf{ground}(P)$ are true in I . The *knowledge ordering* is recalled which, given two three-valued interpretations I_1 and I_2 , is defined as $I_1 \leq_k I_2$ if and only if $I_1 \subseteq I_2$. For a program P and a three-valued interpretation $I \in I_{P,3}$ an *I-partial level mapping* for P is a partial mapping $l : B_P \rightarrow \alpha$ with domain $\mathbf{dom}(l) = \{A \mid A \in I \text{ or } \neg A \in I\}$, where α is some (countable) ordinal. Every such mapping is extended to literals by setting $l(\neg A) = l(A)$ for all $A \in \mathbf{dom}(l)$. Any ordinal α is identified with the set of ordinals β such that $\alpha > \beta$. Thus, any mapping $f : X \rightarrow \{\beta \mid \beta < \alpha\}$ is represented by $f : X \rightarrow \alpha$. Given two ordinals α, β , the lexicographic order ($\alpha \times \beta$) is also an ordinal with $(a, b) \geq (a', b')$ if and only if $a > a'$ or $a = a'$ and $b \geq b'$ for all $(a, b), (a', b') \in \alpha \times \beta$. This order can be split into its components, namely $(a, b) >_1 (a', b')$ if and only if $a > a'$ for all $(a, b), (a', b') \in \alpha \times \beta$ and $(a, b) \geq_2 (a', b')$ if and only if $a = a'$ and $b \geq b'$ for all $(a, b), (a', b') \in \alpha \times \beta$. Additionally we allow the order \succ which given an ordinal $(\alpha \times \beta)$ is defined as $(a, b) \succ (a', b')$ if and only if $b > b'$ for all $(a, b), (a', b') \in (\alpha \times \beta)$.

We shortly recall the level mapping characterization of the well-founded semantics and refer for the original bottom-up operator to [2].

Definition 2.1. ([10]) *Let P be a normal logic program, let I be a model for P , and let l be an I -partial level mapping for P . We say that P satisfies (WF) with respect to I and l if each $A \in \mathbf{dom}(l)$ satisfies one of the following conditions.*

(WF*i*) $A \in I$ and there is a clause $A \leftarrow L_1, \dots, L_n$ in $\mathbf{ground}(P)$ such that $L_i \in I$ and $l(A) > l(L_i)$ for all i .

(WFii) $\neg A \in I$ and for each clause $A \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(P)$ one (at least) of the following conditions holds:

(WFia) There exists i with $\neg A_i \in I$ and $l(A) \geq l(A_i)$.

(WFib) There exists j with $B_j \in I$ and $l(A) > l(B_j)$.

If $A \in \text{dom}(l)$ satisfies (WFi), then we say that A satisfies (WFi) with respect to I and l , and similarly if $A \in \text{dom}(l)$ satisfies (WFii).

Theorem 2.1. ([10]) *Let P be a normal logic program with well-founded model M . Then, in the knowledge ordering, M is the greatest model amongst all models I for which there exists an I -partial level mapping l for P such that P satisfies (WF) with respect to I and l .*

Example 2.1. Consider the program $P = \{p \leftarrow \neg q; q \leftarrow q; r \leftarrow \neg p\}$. We obtain the well-founded model $M = \{p, \neg q, \neg r\}$ with $l(p) = 1$, $l(q) = 0$ and $l(r) = 2$. Note that, for $I = \emptyset$ and arbitrary l , P satisfies (WF) wrt. I and l as well but I is not the greatest such model wrt. \leq_k and thus not the well-founded model.

We continue extending some of the previous notions to the disjunctive case. Let I be a set of disjunction literals. The *closure* of I , $\text{cl}(I)$, is the least set I' with $I \subseteq I'$ satisfying the following conditions: if $D \in I'$ then $D' \in I'$ for all D' with $D \subseteq D'$, and for all disjunctions D_1 and D_2 , $\neg D_1 \in I'$ and $\neg D_2 \in I'$ if and only if $\neg(D_1 \vee D_2) \in I'$. Then, I is *consistent* if there is no $D \in \text{cl}(I)$ with $\neg D \in \text{cl}(I)$ as well⁴. A *disjunctive three-valued interpretation* I of a disjunctive program Π is a consistent set $A \cup \neg B$, $A, B \subseteq EB_\Pi$, where elements in A are **t**, elements in B are **f**, and the remaining wrt. EB_Π are **u**. The body of a ground clause $H \leftarrow A, \neg B$ is true in I if and only if all literals in the body are true in I , or false in I if and only if there is a D such that either $D \subseteq A$ with $\neg D \in I$ or $D \subseteq B$ with $D \in I$ ⁵. Otherwise the body is undefined. The truth of a ground clause $H \leftarrow \text{body}$ is identical to normal programs and I is a *disjunctive three-valued model* of Π if every clause in $\text{ground}(\Pi)$ is true in I . The *disjunctive knowledge ordering* \preceq_k is defined as $I_1 \preceq_k I_2$ if and only if $I_1 \subseteq I_2$ and the corresponding level mapping is extended as follows.

Definition 2.2. *For a disjunctive program Π and a disjunctive interpretation I a disjunctive I -partial level mapping for Π is a partial mapping $l : EB_\Pi \rightarrow \alpha$ with domain $\text{dom}(l) = \{D \mid D \in I \text{ or } \neg D \in I\}$, where α is some (countable) ordinal. Every such mapping is extended to negated disjunctions by setting $l(\neg D) = l(D)$ for all $D \in EB_\Pi$.*

Another way of representing disjunctive information are *state-pairs* $A \cup \neg B$, where A is a subset of EB_Π such that for all D' if $D \in A$ and $D \subseteq D'$ then $D' \in A$, and B is a subset of CB_Π such that for all C' if $C \in B$ and $C \subseteq C'$

⁴ Here, a consistent set is not automatically closed, in contrast with the assumption made in [6].

⁵ The extension is necessary since we might e.g. know the truth of some disjunction without knowing which particular disjunct is true.

then $C' \in B$. Disjunctions in A are **t**, conjunctions in B are **f**, and all remaining are **u**. A state-pair is consistent if whenever $D \in A$ then there is at least one disjunct $D' \in D$ such that $D' \notin B$ and whenever $C \in B$ then there is at least one conjunct $C' \in C$ such that $C' \notin A$. The notions of models and the disjunctive knowledge ordering can easily be adopted. Note that a state-pair is not necessarily consistent and that it contains indefinite positive and negative information in opposite to disjunctive interpretations where negative information will be precise. Level mappings are adjusted to state-pairs in the following and now we do not extend the mapping to identify $l(D) = l(\neg D)$ since in a state-pair D is a disjunction and $\neg D$ a negated conjunction.

Definition 2.3. For a disjunctive program Π and a state-pair I a disjunctive I -partial level mapping for Π is a partial mapping $l : (EB_\Pi \cup \neg CB_\Pi) \rightarrow \alpha$ with domain $\text{dom}(l) = \{D \mid D \in I \text{ or } \neg D \in I\}$, where α is some (countable) ordinal.

3 Strong Well-founded Semantics

We start with SWFS which was introduced by Ross [6] and based on disjunctive interpretations. The derivation rules of the applied top-down procedure are the following. Given a set of disjunction literals I and a disjunctive program Π the *derivate* I' is *strongly derived* from I ($I \Leftarrow I'$) if I contains a disjunction D and $\text{ground}(\Pi)$ a clause $H \leftarrow A_1, \dots, A_n, \neg B$ such that either

- (S1) $H \subseteq D$ and $I' = (I \setminus \{D\}) \cup \{A_1 \vee D, \dots, A_n \vee D\} \cup \neg B$ or
- (S2) $H \not\subseteq D$, $H \cap D \neq \emptyset$, $C = H \setminus D$, and $I' = (I \setminus \{D\}) \cup A \cup \neg B \cup \neg C$.

Consider a ground disjunction D , let $I_0 = \{D\}$ and suppose that $I_0 \Leftarrow I_1 \Leftarrow I_2 \dots$, then $I_0, I_1, I_2 \dots$ is a (*strong*) *derivation sequence* for D . An *active* (*strong*) *derivation sequence* for D is a finite derivation sequence for D whose last element, also called a *basis* of D , is either empty or contains only negative literals. A basis $I = \{\neg l_1, \dots, \neg l_n\}$ is turned into a disjunction $\bar{I} = l_1 \vee \dots \vee l_n$ and if I is empty, denoting **t**, then \bar{I} denotes **f**. Thus, a *strong global tree* Γ_D^S for a given disjunction $D \in EB_\Pi$ contains the root D and its children are all disjunctions of the form \bar{I} , where I ranges over all bases for D . The *strong well-founded model* of a disjunctive program Π is called $M_{WF}^S(\Pi)$ and $D \in M_{WF}^S(\Pi)$, i.e. D is true, if some child of D is false and $\neg D \in M_{WF}^S(\Pi)$, i.e. D is false, if every child of D is true. Otherwise, D is undefined and neither D nor $\neg D$ occur in $M_{WF}^S(\Pi)$. In [6], it was shown that $M_{WF}^S(\Pi)$ is a consistent interpretation and that, for normal programs, SWFS coincides with the well-founded semantics⁶.

Example 3.1. The following program Π will be used to demonstrate the behavior of the three semantics.

$$\begin{array}{llll} p \vee q \leftarrow \neg q & b \vee l \leftarrow \neg r & c \leftarrow \neg l, \neg r & f \leftarrow \neg e \\ q \leftarrow \neg q & l \vee r \leftarrow & e \leftarrow \neg f, c & g \leftarrow e \end{array}$$

⁶ More precisely, the disjunctive model has to be restricted to (non-disjunctive) literals.

We obtain a sequence $\{l \vee r\} \Leftarrow \{\}$ and $l \vee r$ is true as expected. Furthermore, there is a finite sequence in Γ_e^S , namely $\{e\} \Leftarrow \{\neg f, e \vee c\} \Leftarrow \{\neg f, \neg l, \neg c\}$ with the only (true) child and e is false. Thus, we have that $M_{WF}^S(\Pi) = \{l \vee r, f, \neg b, \neg c, \neg e, \neg g\}$ while p and q remain undefined. Literally, this is only a small part of the model and we might close the model (e.g. $\neg(e \vee g) \in M_{WF}^S$) for this example, but the strong well-founded is not necessarily closed which does not allow us to add this implicit information in general.

The level mapping framework is based on bottom-up operators and SWFS is a top-down-procedure so we introduced a bottom-up operator on derivation trees defined on $\mathbf{\Gamma}_\Pi^S$ which is the power set of Γ_Π^S - the set of all strong global trees with respect to Π .

Definition 3.1. *Let Π be a disjunctive logic program, $M_{WF}^S(\Pi)$ the strong well-founded model, and $\Gamma \in \mathbf{\Gamma}_\Pi^S$. We define:*

- $T_\Pi^S(\Gamma) = \{\Gamma_D^S \in \Gamma_\Pi^S \mid \Gamma_D^S \text{ contains an active strong derivation sequence } \{D\}, I_1, \dots, I_r \text{ with child } C = \bar{I}_r \text{ and } I_1 = \{D_1, \dots, D_n, \neg D_{n+1}, \dots, \neg D_m\} \text{ where } \neg C \in M_{WF}^S(\Pi), \Gamma_C^S \in \Gamma \text{ if } C \neq \{\}, \Gamma_{D_i}^S \in \Gamma, D_i \in M_{WF}^S, \Gamma_{D_j}^S \in \Gamma, \neg D_j \in M_{WF}^S \text{ for all } i = 1, \dots, n \text{ and } j = n+1, \dots, m\}$
- $U_\Pi^S(\Gamma) = \{\Gamma_D^S \in \Gamma_\Pi^S \mid \text{for all active strong derivation sequences in } \Gamma_D^S \text{ the corresponding child } C \text{ is true in } M_{WF}^S(\Pi) \text{ and } \Gamma_C^S \in \Gamma\}$

The information is joined by $W_\Pi^S(\Gamma) = T_\Pi^S(\Gamma) \cup U_\Pi^S(\Gamma)$ and iterated: $W_\Pi^S \uparrow 0 = \emptyset$, $W_\Pi^S \uparrow n+1 = W_\Pi^S(W_\Pi^S \uparrow n)$, and $W_\Pi^S \uparrow \alpha = \bigcup_{\beta < \alpha} W_\Pi^S \uparrow \beta$ for limit ordinal α . It was shown in [11] that W_Π^S is monotonic, allowing to apply the Tarski fixed-point theorem which yields that the operator W_Π^S always has a least fixed point, and that this least fixed point coincides with M_{WF}^S . This was used to derive the following alternative characterization of SWFS.

Definition 3.2. *Let Π be a disjunctive logic program, let I be a model for Π , and let l be a disjunctive partial level mapping for Π . We say that Π satisfies (SWF) with respect to I and l if each $D \in \text{dom}(l)$ satisfies one of the following conditions:*

(SWFi) $D \in I$ and Γ_D^S contains an active strong derivation sequence with child C , $\neg C \in I$ and $l(D) > l(C)$ if $C \neq \{\}$, and there is a clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Pi)$ which is used for the first derivation of that sequence such that $\neg B_j \in I$ and $l(D) > l(B_j)$, $1 \leq j \leq m$, and one of the following conditions holds:

(SWFia) $H \subseteq D$ such that there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$ and $l(D) > l(D_i \vee A_i)$, $1 \leq i \leq n$.

(SWFib) $H \not\subseteq D$, $H \cap D \neq \emptyset$, $\{C_1, \dots, C_l\} = H \setminus D$, $A_i \in I$ and $l(D) > l(A_i)$, $1 \leq i \leq n$, and $\neg C_k \in I$ and $l(D) > l(C_k)$, $1 \leq k \leq l$.

(SWFii) $\neg D \in I$ and for each active strong derivation sequence in Γ_D^S with child $C \in I$ there is a clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Pi)$ which is used for the first derivation of that sequence such that (at least) one of the following conditions holds:

(SWFiia') $H \subseteq D$ and there exists i , $1 \leq i \leq n$, with $\neg(A_i \vee D) \in I$, $l(D) \geq l(A_i \vee D)$.
 (SWFiia'') $H \not\subseteq D$, $H \cap D \neq \emptyset$, and there exists i with $\neg A_i \in I$, $l(D) \geq l(A_i)$, $1 \leq i \leq n$.
 (SWFiib') $H \subseteq D$ and there exists D' with $D' \subseteq B$, $D' \in I$ and $l(D) > l(D')$.
 (SWFiib'') $H \not\subseteq D$, $H \cap D \neq \emptyset$, $C = (H \setminus D)$, and there exists D' with $D' \subseteq (B \cup C)$, $D' \in I$ and $l(D) > l(D')$.
 (SWFiic) $l(D) > l(C)$.

Theorem 3.1. *Let Π be a disjunctive program with strong well-founded model M . Then, in the disjunctive knowledge ordering, M is the greatest model amongst all models I for which there exists a disjunctive I -partial level mapping l for Π such that Π satisfies (SWF) with respect to I and l .*

The characterization is obviously more involved than Definition 2.1. In fact, even though it appears that for every true disjunction there are a sequence and a clause satisfying (SWFia), we were unable to show that due to the missing closure property of the strong well-founded model. Thus, we have to keep the condition (SWFib). Moreover, it can be checked that all the cases for negated disjunctions yield that (SWFiic) holds as well. We therefore could have formulated (SWFii) just using (SWFiic), but for a better comparison to the characterization of well-founded semantics and the following semantics we separate the case. We continue with the example.

Example 3.2. (Example 3.1 continued) As shown in [11], we obtain $l(D) = \alpha$, where α is the least ordinal such that $\Gamma_D^S \in (W_\Pi^S \uparrow (\alpha + 1)) = W_\Pi^S(W_\Pi^S \uparrow \alpha)$. Thus, we have $l(l \vee r) = 0$ by (SWFia) and $l(e) = 1$ by (SWFiia') and therefore $l(f) = 2$ by (SWFia). Moreover, $l(b) = 1$ by (SWFiib') whereas $l(c) = 1$ by (SWFiib''). Note that in case of b , c , and e also (SWFiic) is satisfied.

It should be mentioned that the reference to derivation sequences in the conditions is also necessary because of the missing closure property of $M_{WF}^S(\Pi)$.

4 Generalized Disjunctive Well-founded Semantics

Baral, Lobo, and Minker introduced GDWFS [7] based on state-pairs. They applied various operators for calculating the semantics and we recall at first \mathcal{T}_S^D and \mathcal{F}_S^D for disjunctive programs.

Definition 4.1. *Let S be a state-pair and Π be a disjunctive program. Let $T \subseteq EB_\Pi$ and $F \subseteq CB_\Pi$.*

$\mathcal{T}_S^D(T) = \{D \in EB_\Pi \mid D \text{ undefined in } S, H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m \text{ in } \text{ground}(\Pi) \text{ such that for all } i, 1 \leq i \leq n, (A_i \vee D_i) \in S \text{ or } (A_i \vee D_i) \in T, D_i \text{ might be empty, } \neg B_j \in S \text{ for all } j, 1 \leq j \leq m, \text{ and } (H \cup \bigcup_i D_i) \subseteq D.\}$

$\mathcal{F}_S^D(F) = \{C \in CB_\Pi \mid C \text{ is undefined in } S, A \in C, \text{ and for all clauses } H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m \text{ in } \text{ground}(\Pi), \text{ with } A \in H, \text{ at least one of the following three cases holds: } (B_1 \vee \dots \vee B_m) \in S, \neg(A_1 \wedge \dots \wedge A_n) \in S, \text{ or } \neg(A_1 \wedge \dots \wedge A_n) \in F\}$

T_S^D is bottom-up and F_S^D is top-down: $T_S^D \uparrow 0 = \emptyset$, $T_S^D \uparrow (n+1) = T_S^D(T_S^D \uparrow n)$, $T_S^D = \bigcup_{n < \omega} T_S^D \uparrow n$, and $\mathcal{F}_S^D \downarrow 0 = CB_{\Pi}$, $\mathcal{F}_S^D \downarrow (n+1) = \mathcal{F}_S^D(\mathcal{F}_S^D \downarrow n)$, $\mathcal{F}_S^D = \bigcap_{n < \omega} \mathcal{F}_S^D \downarrow n$.

There are two more operators defined for definite programs which necessitates the following program transformations. Given a disjunctive program Π and a state-pair S , $DIS(\Pi)$ is obtained by transferring all negated atoms in the body of each clause of Π as atoms to its head. Then, $Dis(\Pi, S)$ results from $DIS(\Pi)$ by reducing the clauses in $DIS(\Pi)$ as follows: remove atoms from the body of a clause if they are true in S , remove a clause if its head is true in S , and remove atoms from the head of a clause if they are false in S . This is similar to the construction used for stable models and we recall $T_H^D(T)$, a simplification of $T_S^D(T)$ to definite programs. Given a definite (disjunctive) program Π and T , a subset of EB_{Π} , we have that $T_H^D(T) = \{D \in EB_{\Pi} \mid H \leftarrow A_1, \dots, A_n \text{ in } \text{ground}(\Pi) \text{ such that for all } i, 1 \leq i \leq n, (A_i \vee D_i) \in T, D_i \text{ might be empty, and } (H \cup \bigcup_i D_i) \subseteq D.\}$ We then iterate $T_H^D \uparrow 0 = \emptyset$, $T_H^D \uparrow (n+1) = T_H^D(T_H^D \uparrow n)$, and $T_H^D = \bigcup_{n < \omega} T_H^D \uparrow n$.

For deriving indefinite false conjunctions the Extended Generalized Closed World Assumption (EGCWA) [8] is applied. It intuitively says that a conjunction can be inferred to be false from Π if and only if it is false in all minimal models of Π where a minimal model [12] is a two-valued model M of Π such that no subset of it is a model as well.

The previous two constructions yield the operators $T_S^{ED} = \{D \mid D \in T_{Dis(\Pi, S)}^D \text{ and } D \notin S\}$ and $F_S^{ED} = \{C \mid C \in \text{EGCWA}(Dis(\Pi, S) \cup S) \text{ and } C \notin S\}$. Now we can combine all the operators and obtain

$$S^{ED}(S) = S \cup T_S^D \cup \neg \mathcal{F}_S^D \cup T_S^{ED} \cup \neg F_S^{ED}.$$

The iteration is done via $M_0 = \emptyset$, $M_{\alpha+1} = S^{ED}(M_{\alpha})$, and $M_{\alpha} = \bigcup_{\beta < \alpha} M_{\beta}$ for limit ordinal α , and has a fixed point [7]. The fixed point corresponds to the generalized disjunctive well-founded model M_{Π}^{ED} which is consistent [7]. However, for normal logic programs in general, the well-founded semantics and GDWFS do not coincide.

Example 4.1. Recall the program from Example 3.1. We have $M_{\Pi}^{ED} = \{l \vee r, q, f, \neg p, \neg b, \neg c, \neg e, \neg g, \neg(l \wedge r)\}$. Note that M_{Π}^{ED} is closed in so far that any superset of a true disjunction (false conjunction) is true (false) as well. Moreover, the semantics concludes q to be true from $q \leftarrow \neg q$. This is exactly the kind of reasoning which is not applied for the well-founded semantics, thus being the cause for GDWFS and WFS not to coincide on normal programs.

We continue with the level mapping characterization of GDWFS.

Definition 4.2. *Let Π be a disjunctive logic program, let the state-pair I be a model for Π , and let l_1, l_2 be disjunctive I -partial level mappings for Π . We say that Π satisfies (GDWF) with respect to I, l_1 , and l_2 if each $D \in \text{dom}(l_1)$ and each $\neg C \in \text{dom}(l_1)$ satisfies one of the following conditions:*

(GDWFi) $D \in I$ and there is a clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Pi)$ with $H \subseteq D$ such that $\neg B_j \in I$ and $l_1(D) >_1 l_t(\neg B_j)$, $t \in \{1, 2\}$, for all $j = 1, \dots, m$, and, for all $i = 1, \dots, n$, there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$ where $l_1(D) >_1 l_1(D_i \vee A_i)$ or $l_1(D) >_1 l_2(D_i \vee A_i)$.

(GDWFii) $\neg C \in I$ with atom $A \in C$ and for each clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Pi)$ with $A \in H$ (at least) one of the following conditions holds:

(GDWFiia) $\neg(A_1 \wedge \dots \wedge A_n) \in I$ and $l_1(\neg C) \geq l_1(\neg(A_1 \wedge \dots \wedge A_n))$.

(GDWFiia') $\neg(A_1 \wedge \dots \wedge A_n) \in I$ and $l_1(\neg C) >_1 l_2(\neg(A_1 \wedge \dots \wedge A_n))$.

(GDWFiib) $(B_1 \vee \dots \vee B_m) \in I$ and $l_1(\neg C) >_1 l_t(B_1 \vee \dots \vee B_m)$ for $t \in \{1, 2\}$.

and each $D, \neg C \in \text{dom}(l_2)$ satisfies one of the following conditions:

(GDWFi') $D \in I$ and there is a clause $H_1 \vee \dots \vee H_l \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Pi)$ such that $\emptyset \neq ((H \cup B) \setminus D') \subseteq D$, $H_k \in D'$ for each $\neg H_k \in I$ with $l_2(D) >_1 l_t(\neg H_k)$, $t \in \{1, 2\}$, $B_j \in D'$ for each $\neg B_j \in I$ with $l_2(D) >_1 l_t(\neg B_j)$, $t \in \{1, 2\}$, for all $k = 1, \dots, l$ and all $j = 1, \dots, m$, and, for all $i = 1, \dots, n$, there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$ where $l_2(D) >_2 l_2(D_i \vee A_i)$ or $A_i \in I$ where $l_2(D) >_1 l_s(A_i)$, $s \in \{1, 2\}$.

(GDWFii') $\neg C \in I$ and $C \in \text{EGCWA}(\text{Dis}(\Pi, S) \cup S)$, $C \notin S$ and $l_2(\neg C) >_1 l_t(L)$, $t \in \{1, 2\}$, if and only if $L \in S$.

The reason for introducing two mappings is to extrapolate exactly the simultaneous iteration of the two operators dealing with positive, negative respectively, information. The very same argument necessitates the different orderings. From a more general perspective, e.g. (GDWFiia) and (GDWFiia') employ basically the same kind of dependency, just the proof of the following theorem stating the equivalence enforces the diverse conditions.

Theorem 4.1. *Let Π be a disjunctive program with generalized disjunctive well-founded model M . Then, in the disjunctive knowledge ordering, M is the greatest model amongst all models I for which there exist disjunctive I -partial level mappings l_1 and l_2 for Π such that Π satisfies (GDWF) w.r.t. I , l_1 , and l_2 .*

Example 4.2. (Example 4.1 continued) From [11] we know that if $D \in \mathcal{T}_{M_\alpha}^D$ then β is the least ordinal such that $D \in \mathcal{T}_{M_\alpha}^D \uparrow (\beta+1)$ and $l_1(D) = (\alpha, \beta)$, if $D \in \mathcal{T}_{M_\alpha}^{ED}$ then β is the least ordinal such that $D \in \mathcal{T}_{\text{Dis}(\Pi, M_\alpha)}^D \uparrow (\beta+1)$ and $l_2(D) = (\alpha, \beta)$. For negative conjunctions it holds that $l_1(\neg C) = (\alpha, 0)$ if $C \in \mathcal{F}_{M_\alpha}^D$ and $l_2(\neg C) = (\alpha, 0)$ if $C \in \mathcal{F}_{M_\alpha}^{ED}$. Thus, we obtain e.g. $l_1(l \vee r) = l_2(l \vee r) = (0, 0)$ by (GDWFi) and (GDWFi'), $l_1(f) = (2, 0)$ by (GDWFi), $l_2(\neg p) = (0, 0)$ by (GDWFii'), $l_1(\neg e) = (1, 0)$ by (GDWFiiia') and $l_1(\neg g) = (1, 0)$ by (GDWFiia).

The condition (GDWFii') directly refers to EGCWA due to problems with minimal models in the level mapping framework (see Section 6).

5 Disjunctive Well-founded Semantics

The third approach we study is the disjunctive well-founded semantics presented by Brass and Dix in [9]. We use again disjunctive interpretations for representing information even though in [9] the syntactically different pure disjunctions are applied. D-WFS is only defined for (disjunctive) DATALOG programs which are programs whose corresponding language does not have any function symbols apart from (nullary) constants. Thus they correspond to propositional programs and we use the notation Φ from [9] for DATALOG programs.

We recall the operators defining D-WFS. Both map sets of *conditional facts* which are disjunctive clauses without any positive atoms in the body and we start with T_Φ . Given Φ and a set of conditional facts Γ , we have that $T_\Phi(\Gamma) = \{(H \cup \bigcup_i (H_i \setminus \{A_i\})) \leftarrow \neg(B \cup \bigcup_i B_i) \mid \text{there is } H \leftarrow A_1, \dots, A_n, \neg B \text{ in } \text{ground}(\Phi) \text{ and conditional facts } H_i \leftarrow \neg B_i \in \Gamma \text{ with } A_i \in H_i \text{ for all } i = 1, \dots, n.\}$ The iteration of T_Φ is given as $T_\Phi \uparrow 0 = \emptyset$, $T_\Phi \uparrow (n+1) = T_\Phi(T_\Phi \uparrow n)$, and $T_\Phi = \bigcup_{n < \omega} T_\Phi \uparrow n$ and yields a fixed point.

The next operator is top-down starting with the previous fixed point also applying the notion of $\text{heads}(S)$ which is the set of all atoms occurring in some head of a clause contained in a given set of ground clauses S : given a set of conditional facts Γ we define $R(\Gamma) = \{H \leftarrow \neg(B \cap \text{heads}(\Gamma)) \mid H \leftarrow \neg B \in \Gamma, \text{ and there is no } H' \leftarrow \text{ in } \Gamma \text{ with } H' \subseteq B \text{ or there is no } H' \leftarrow \neg B' \text{ in } \Gamma \text{ with } H' \subseteq H \text{ and } B' \subseteq B \text{ where at least one } \subseteq \text{ is proper.}\}$ Note that the second condition forcing one \subseteq to be proper is necessary since otherwise we could remove each conditional fact by means of itself. The iteration of this operator is defined as $R \uparrow 0 = T_\Phi$, $R \uparrow (n+1) = R(R \uparrow n)$ and the fixed point of this operator is called the *residual program* of Φ .

Given the residual program $\text{res}(\Phi)$, the disjunctive well-founded model M_Φ is $M_\Phi = \{D \in EB_\Phi \mid \text{there is } H \leftarrow \text{ in } \text{res}(\Phi) \text{ with } H \subseteq D\} \cup \{\neg D \mid D \in EB_\Phi \text{ and } \forall D' \in D : D' \notin \text{heads}(\text{res}(\Phi))\}$. Though T_Φ is monotonic, R is not and we cannot generalize the following results to all disjunctive logic programs. We should note that in [13] the approach was extended to disjunctive logic programs by combining the transformation rules with constraint logic programming. But the operators are not extended as well and we remain with that restriction.

Example 5.1. Recall Π from Example 3.1. It is obvious that Π is also a DATALOG program and we obtain $M_\Pi = \{l \vee r, f, \neg p, \neg c, \neg e, \neg g\}$. Note that M_Π is closed by definition of the model.

In the following, we present the alternative characterization of D-WFS.

Definition 5.1. *Let Φ be a DATALOG program, let I be a model for Φ , and let l be a disjunctive I -partial level mapping for Φ . We say that Φ satisfies (DWF) with respect to I and l if each $D \in \text{dom}(l)$ satisfies one of the following conditions:*

(DWF i) $D \in I$ and there is a clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Phi)$ with $H \subseteq D$ such that there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$, $l(D) > l(D_i \vee A_i)$, and $l(D) \succ l(D_i \vee A_i)$ if $l(D) >_1 l(D_i \vee A_i)$, for all $i = 1, \dots, n$, and $\neg B_j \in I$ and $l(D) >_1 l(B_j)$ for all $j = 1, \dots, m$.

(DWFii) $\neg D \in I$ and for each clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(\Phi)$ with $A \in H$ and $A \in D$ (at least) one of the following conditions holds:

(DWFiia) $\neg A_i \in I$ and $l(D) \geq l(A_i)$.

(DWFiib) $D' \in I$ with $D' \subseteq B$ and $l(D) >_1 l(D')$.

(DWFii') $\neg D \in I$ and for each conditional fact $H \leftarrow \neg B$ in T_Φ with $A \in H$ and $A \in D$ (at least) one of the following conditions holds:

(DWFiia') there is $H' \leftarrow \neg B'$ in $R \uparrow \alpha$ with $H' \subset H$ and $B' \subseteq (B \setminus D')$ where $A \notin H'$, $B_j \in B$, $\neg B_j \in I$, and $l(D) >_1 (l(B_j) + 1)$ for all $B_j \in D'$, and $l(D) >_1 (\alpha, \beta)$ for some β .

(DWFiib') $D' \in I$ with $D' \subseteq B$ and $l(D) >_1 l(D')$.

It is evident that (DWFiib) and (DWFiib') apply the same kind of dependency only that the former does this wrt. to one clause while the latter may employ several, i.e. (DWFiib) can be considered a special case of (DWFiib') which appears basically for easier comparison.

Theorem 5.1. *Let Φ be a (disjunctive) DATALOG program with disjunctive well-founded model M . Then, in the disjunctive knowledge ordering, M is the greatest model amongst all models I for which there exists a disjunctive I -partial level mapping l for Φ such that Φ satisfies (DWF) with respect to I and l .*

Example 5.2. (Example 5.1 continued) From [11] we know that if $D \in M$ then $l(D) = (\alpha, \beta)$ where α is the least ordinal such that $H \leftarrow$ in $R \uparrow \alpha$ with $H \subseteq D$ and β is the least ordinal such that the corresponding conditional fact $H \leftarrow \neg B$ in $T_\Phi \uparrow (\beta + 1)$. Furthermore, if $\neg D \in M$ then $l(D) = (\alpha, 0)$ where α is the least ordinal such that for each $A \in D$ there is no conditional fact $H \leftarrow \neg B$ in $R \uparrow \alpha$ with $A \in H$. Thus, we obtain e.g. $l(f) = (2, 0)$ by (DWFi), $l(p) = (1, 0)$ by (DWFiia'), $l(c) = (1, 0)$ by (DWFiib) and $l(e) = (1, 0)$ by (DWFiia).

Finally, we mention that \succ was introduced for technical reasons in the proof to match the precise behavior of the operators [11].

6 Discussions

6.1 Comparison of the Characterizations

It was already shown in [9] that D-WFS and GDWFS satisfy five program transformation principles while SWFS does not, and that GDWFS always derives more or equal knowledge than D-WFS [14]. However, there is no similar result for D-WFS and SWFS since they are incomparable with respect to the derived knowledge (cf. our main example: SWFS derives $\neg b$ while D-WFS concludes $\neg p$).

We will now further compare the semantics on the basis of our characterizations. We will in particular attempt to obtain some insights into good general criteria for a well-founded semantics for disjunctive programs.

Level-mapping characterizations separate positive and negative information. One key insight which can be drawn from our investigations is that any characterization basically states that a true disjunction D satisfies the following scheme with respect to the model I and the program Π .

$D \in I$ and there is a clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(II)$ with $H \subseteq D$ such that there is $D_i \subseteq D$ with $(D_i \vee A_i) \in I$, $l(D) > l(D_i \vee A_i)$, for all $i = 1, \dots, n$, and $\neg B_j \in I$ and $l(D) > l(B_j)$ for all $j = 1, \dots, m$.

We can see that this corresponds in general to (SWFia) from Definition 3.2, to (GDWFi) from Definition 4.2, and to (DWFi) from Definition 5.1. We only have to consider that the relation $>$ is technically not sufficient and that we sometimes apply a more precise order. Nevertheless, in all cases we obtain levels such that $l(D)$ is greater with respect to the specific ordering. There are further differing details. For (SWF), we have to abstract additionally from the notion of derivation sequences and their children, and there is also (SWFib) which arises from proof-theoretical treatments. In case of (GDWF) we have additionally a condition (GDWFi') but that is the part (corresponding to T_H^D) which derives more knowledge than the well-founded semantics and should thus not be an intended result for a well-founded semantics for disjunctive programs. We claim that the condition given above is the 'disjunctive' version of (WFi) from Definition 2.1 and we propose it to be a condition for any semantics aiming to extend the well-founded semantics to disjunctive programs.

If we look for adequate extensions of (WFi) to disjunctive programs then we see that the conditions for negative information differ more. However, we still obtain straightforward extensions of (WFi) for each of the semantics only abstracting a little from the technical details. We generalize to the following scheme:

$\neg D \in I$ and for each clause $H \leftarrow A_1, \dots, A_n, \neg B_1, \dots, \neg B_m$ in $\text{ground}(II)$ with $A \in H$ and $A \in D$ (at least) one of the following conditions holds:

- (iia) $\neg \alpha \in I$ and $l(D) \geq l(\alpha)$.
- (iib) $D' \in I$ with $D' \subseteq B$ and $l(D) >_1 l(D')$.

For SWFS, we have (SWFiia') with $\alpha = D \vee A_i$ and (SWFiia'') with $\alpha = A_i$ corresponding to (iia) depending on whether $H \subseteq D$ or $H \not\subseteq D$ but $H \cap D \neq \emptyset$ ⁷, and (SWFiib') as equivalent to (iib). In case of GDWFS, we have (GDWFiia) and (GDWFiia') both with $\alpha = \neg(A_1 \wedge \dots \wedge A_n)$ for (iia) and (GDWFiib) for (iib). We only have to take care that $l(D)$ has to be $l(\neg D)$ in this case since we are dealing with negated conjunctions and thus indefinite information. Finally, (DWFiia) with $\alpha = A_i$ and (DWFiib) correspond to (iia) and (iib), respectively. We claim as well that the scheme above should be part of any well-founded semantics for disjunctive programs ignoring some minor differences depending e.g. on whether we represent negative information by conjunctions or disjunctions.

Unfortunately, since positive information may be indefinite, it is also possible to obtain a correspondence to (iib) which results from several clauses (consider the program $II = \{p \vee q \leftarrow; r \leftarrow s, \neg p; s \leftarrow \neg q\}$ where $\neg r$ is derivable). This is covered by (SWFiic), (GDWFi'), and (DWFiib'). Still, this is not the whole characterization for any of the semantics. (SWFiib'') extends (SWFiib') to include particular atoms from the head. (GDWFi') is in fact much more powerful by

⁷ Note that in both cases there is an A common to H and D .

means of the EGCWA and allows for deriving more knowledge difficult to characterize in a clause-based approach. In case of D-WFS we also have (DWFii') which resolves the elimination of non-minimal clauses, a feature not contained in SWFS and also covered by (GDWFii') for GDWFS.

Summarising, it is obvious (and certainly expected) that it is in the derivation of negative information where the semantics differ wildly. All characterizations contain extensions of (WFii), but contain also additional non-trivial conditions some of which are difficult to capture within level mapping characterizations. The obtained uniform characterizations thus display in a very explicit manner the very different natures of the different well-founded semantics – there is simply not enough resemblance between the approaches to obtain a coherent picture. We can thus, basically, only confirm in a more formal way what has been known beforehand, namely that the issue of a good definition of well-founded semantics for disjunctive logic programs remains widely open. We believe, though, that our approach delivers structural insights which can guide the quest.

6.2 Minimal Models

As mentioned when dealing with the EGCWA appearing in GDWFS it is difficult within the level mapping framework to characterize minimal models which are the main evaluation principle for EGCWA. In the appendix of [11] it was concluded that the best possible characterization obtained for minimal models is the following:

Corollary 6.1. ([11]) *Let Π be a definite disjunctive program and M be a model of Π . If there exists a total level mapping $l : B_{\Pi} \rightarrow \alpha$ such that for each $A \in M$ exists a clause $A \vee H_1 \vee \dots \vee H_l \leftarrow A_1, \dots, A_n$ in $\text{ground}(\Pi)$ with $A_i \in M$, $H_k \notin M$ or $l(H_k) > l(A)$, and $l(A) > l(A_i)$, for all $i = 1, \dots, n$ and all $k = 1, \dots, l$, then M is a minimal model of Π .*

This is of course not a characterization but just saying that a model satisfying the given level mapping characterization is in fact minimal. Unfortunately, it is not possible to state this the other way around since there are minimal models which do not satisfy this condition⁸.

Example 6.1.

$$\begin{aligned} a \vee b &\leftarrow \\ a &\leftarrow b \\ b &\leftarrow a \end{aligned}$$

This program has only one minimal model $\{a, b\}$, so according to the condition above, the first clause cannot be used since both atoms in the head are true. With the remaining two clauses we cannot have a level mapping satisfying the given condition since we must have $l(a) > l(b)$ and $l(b) > l(a)$ which is not possible.

⁸ Note though that in [15] a similar result was obtained working in both directions restricted to head-cycle free programs.

Apparently, the condition imposed is too strong but all attempts (cf. [11]) to correct the problem end up with a condition too weak being satisfied also by models which are not minimal.

We can thus not apply a more precise condition for EGCWA. More generally, any semantics based on minimal models seems to fail being characterized in the framework (excluding cases like GDWFS where we simply do not treat the details of EGCWA). So surprisingly, even though disjunctive stable models are a straightforward extension of stable models, the corresponding characterization does not extend easily if at all.

It remains to be said that in opposite to that there exist characterizations for various semantic extensions of the well-founded semantics, though being rather complicated and diverse, which might allow the conclusion that (almost) any of the approaches has a better structural foundation than minimal models.

7 Conclusions

We have characterized three of the extensions of the well-founded semantics to disjunctive logic programs. It has been revealed that these characterizations are non-trivial and we have seen that they share a common derivability for true disjunctions. The conditions for deriving negative information however vary a lot. Some parts of the characterizations are common extensions of conditions used for the well-founded semantics while others cover specific deduction mechanisms occurring only in one semantics. We have obtained some structural insights into the differences and similarities of proposals for disjunctive well-founded semantics, but the main conclusion we have to draw is a negative one: Even under our formal approach which provides uniform characterizations of different semantics, the different proposals turn out to be too diverse for a meaningful comparison. The quest for disjunctive well-founded semantics thus remains widely open. Our uniform characterizations provide, however, arguments for approaching the quest in a more systematic way.

In this paper, we covered only those of the well-founded semantics which a priori appeared to be the most important and promising ones. Obviously, further insights could be obtained from considering also the remaining proposals reported e.g. in [16–20, 14].

References

1. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In Kowalski, R.A., Bowen, K.A., eds.: *Logic Programming. Proceedings of the 5th International Conference and Symposium on Logic Programming*, MIT Press (1988) 1070–1080
2. van Gelder, A., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. *Journal of the ACM* **38** (1991) 620–650
3. Lobo, J., Minker, J., Rajasekar, A.: *Foundations of Disjunctive Logic Programming*. MIT Press (1992)

4. Minker, J., Seipel, D.: Disjunctive logic programming: A survey and assessment. In: *Essays in Honour of Robert A. Kowalski, Part I*, LNAI 2407, Springer (2002)
5. Przymusiński, T.C.: Stable semantics for disjunctive programs. *New Generation Computing* **9** (1991) 401–424
6. Ross, K.A.: The well founded semantics for disjunctive logic programs. In Kim, W., Nicolas, J.M., Nishio, S., eds.: *First International Conference on Deductive and Object Oriented Databases*, Elsevier Science Publishers B.V. (North-Holland) (1990) 385–402
7. Baral, C., Lobo, J., Minker, J.: Generalized disjunctive well-founded semantics for logic programs. In Z. W. Ras, M.Z., Emrich, M., eds.: *Proceedings of the Fifth International Symposium on Methodologies for Intelligent Systems*. ACM SIGMOD-SIGACT-SIGART, North Holland (1989) 465–473
8. Yahya, A., Henschen, L.: Deduction in non-Horn databases. *Journal Automated Reasoning* **1** (1985) 141–160
9. Brass, S., Dix, J.: Disjunctive semantics based upon partial and bottom-up evaluation. In: *Proceedings of the 12th International Logic Programming Conference*, Tokyo, MIT Press (1995) 199–213
10. Hitzler, P., Wendt, M.: A uniform approach to logic programming semantics. *Theory and Practice of Logic Programming* **5** (2005) 123–159
11. Knorr, M., Hitzler, P.: A comparison of disjunctive well-founded semantics. Technical report, AIFB, University of Karlsruhe, Germany (2006) http://www.aifb.uni-karlsruhe.de/Publikationen/showPublikation?publ_id=1383.
12. Minker, J.: On indefinite databases and the closed world assumption. In: *Lecture Notes in Computer Science* 138, Springer, Berlin (1982) 292–308
13. Dix, J., Stolzenburg, F.: Computation of non-ground disjunctive well-founded semantics with constraint logic programming. In Dix, J., Pereira, L.M., Przymusiński, T.C., eds.: *Selected Papers of the Workshop on Non-Monotonic Extensions of Logic Programming in Conjunction with Joint International Conference and Symposium on Logic Programming 1996*, Springer, Berlin, Heidelberg, New York (1997) 202–224
14. Dix, J., Müller, M.: An axiomatic approach to semantics of disjunctive programs. In van Hentenryck, P., ed.: *Logic Programming, Proc. of the 11th International Conference*, MIT Press (1994) 303–320
15. Ben-Eliyahu, R., Dechter, R.: Propositional semantics for disjunctive logic programs. *Ann. Math. Artif. Intell.* **12** (1994) 53–87
16. Wang, K., Zhou, L.: Comparisons and computation of well-founded semantics for disjunctive logic programs. *ACM Trans. Comput. Logic* **6** (2005) 295–327
17. Alcântara, J., Damásio, C.V., Pereira, L.M.: A well-founded semantics with disjunction. In M. Gabbrielli, G.G., ed.: *Proceedings of the 21st Intl.Conf. on Logic Programming (ICLP '05)*. LNCS 3668, Springer (2005) 341–355
18. Przymusiński, T.: Stationary Semantics for Normal and Disjunctive Logic Programs. In Delobel, C., Kifer, M., Masunaga, Y., eds.: *DOOD '91, Proceedings of the 2nd International Conference*, Springer, LNCS 566 (1991)
19. Przymusiński, T.C.: Static semantics for normal and disjunctive logic programs. *Annals of Mathematics and Artificial Intelligence* **14** (1995) 323–357
20. Baral, C., Lobo, J., Minker, J.: Wf 3 : A semantics for negation in normal disjunctive logic programs. In Ras, Z.W., Zemankova, M., eds.: *Methodologies for Intelligent Systems*, LNAI 542, Springer (1991) 459–468