Clausal Deductive Databases and a General Framework for Semantics in Disjunctive Deductive Databases

Dietmar Seipel

University of Würzburg, Dept. of Computer Science
Am Hubland, D – 97074 Würzburg, Germany
seipel@informatik.uni-wuerzburg.de
Tel. +49 931 888 5026, Fax. +49 931 888 4600

Abstract. In this paper we will investigate the novel concept of *clausal deductive databases* (cd–databases), which are special normal deductive databases – i.e. deductive databases which may contain default negation in rule bodies – over a meta–language $\mathcal{L}^{\mathsf{cd}}$ with a fixed set of predicate symbols, namely *dis*, *con*, and some *built–in* predicate symbols. The arguments of the literals in $\mathcal{L}^{\mathsf{cd}}$ are given by disjunctive and conjunctive clauses of a basic first–order language $\mathcal{L}$ (which are considered as terms in $\mathcal{L}^{\mathsf{cd}}$). On the other hand, *disjunctive deductive databases* (dd–databases) extend normal deductive databases by allowing for disjunctions (rather than just single atoms or literals) in rule heads – next to default negation in rule bodies.

We will present an *embedding* of dd–databases into cd–databases: a dd–database $\mathcal{D}$ is transformed into a cd–database $\mathcal{D}^{\mathsf{cd}}$, which talks about the clauses of $\mathcal{D}$ – rather than just the literals. Thus, cd–databases provide a *flexible framework* for declaratively specifying the semantics of dd–databases. We can fix a standard *control strategy*, e.g. stable model or well–founded semantics, and vary the *logical description* $\mathcal{D}^{\mathsf{cd}}$ for specifying different semantics. The *transformed database* $\mathcal{D}^{\mathsf{cd}}$ usually consists of a part $\mathcal{D}^{\otimes}$ which naturally expresses the rules of $\mathcal{D}$, and two generic parts which are independent of $\mathcal{D}$: $\mathcal{D}^{logic}$ specifies *logical inference rules* like resolution and subsumption, and $\mathcal{D}^{cwa}$ specifies non–monotonic inference rules like *closed–world–assumptions*.

In particular we will show that the *hyperresolution* consequence operator $\mathcal{T}_{\mathcal{D}}^{s}$ for dd–databases without default negation can be expressed as a standard consequence operator $\mathcal{T}_{\mathcal{D}^{\mathsf{cd}}}$, for a suitable transformed database $\mathcal{D}^{\mathsf{cd}}$, where $\mathcal{D}^{logic} = \mathcal{D}^{cwa} = \emptyset$. For dd–databases with default negation we can show that the semantics of *stable models* can be characterized by adding suitable sets $\mathcal{D}^{logic}$ and $\mathcal{D}^{cwa}$. Moreover, we will define a new semantics for dd–databases which we will call *stable state semantics*; it is based on Herbrand states rather than Herbrand interpretations.

Keywords

disjunctive deductive databases, clausal logic programming, program transformations, non–monotonic reasoning, stable models, closed–world–assumptions

1 Introduction

In the last twenty years the concept of relational databases has been extended in various ways, e.g. to deductive databases, object–oriented databases, and active databases [17, 25]. *Deductive databases* [2] provide a very powerful formalism for representing knowledge in the form of facts – which correspond to the tuples in a relational database – and different forms of inference rules. The *semantics* of normal deductive databases with rules that may contain default negation in rule bodies, is well–understood, and it is defined using special concepts of *non–monotonic reasoning*. The most prominent approaches are the *stable model semantics* [8] and the *well–founded semantics* [24]. *Disjunctive deductive databases* [11] allow for disjunctions – rather than just single atoms or literals – in rule heads, next to default negation in rule bodies; thus it becomes possible to specify knowledge with uncertain conclusions. For disjunctive deductive databases, the situation is much more complicated; there exist various generalizations of the stable model semantics [16, 5] and the well–founded semantics [11, 1], including some semantics dealing with three or more truth values[1].

In this paper we will investigate the novel concept of *clausal deductive databases*, which are special normal deductive databases over a meta–language $\mathcal{L}^{\mathsf{cd}}$ with a small set of predicate symbols which are talking about the clauses of a basic first–order language $\mathcal{L}$. E.g., the following inference rule is an allowed rule:

$$r = dis(X_d) \leftarrow complement(X_d, X_c) \wedge not\, con(X_c).$$

The arguments of the so–called cd–atoms $dis(X_d)$ and $con(X_c)$ can be arbitrary disjunctive or conjunctive formulas $X_d, X_c \in \mathcal{L}$, respectively. In addition, there are so–called bi–atoms, e.g. $complement(X_d, X_c)$, $subsumes(X_d, X'_d)$ or $resolvent(X_1, X_2, X_d)$, with special *built–in* predicate symbols with a fixed, natural interpretation known from predicate logic. The rule r above represents a *closed–world–assumption* for inferring knowledge (the disjunction X_d) in the absence of the complementary knowledge (the conjunction X_c)[2].

CD–databases provide a flexible *framework for specifying semantics* of dd–databases in a *declarative* way. It is based on the idea of transforming a dd–database $\mathcal{D}$ into a cd–database $\mathcal{D}^{\mathsf{cd}}$, the cd*–transformation* of $\mathcal{D}$, and then defining the (disjunctive) semantics of $\mathcal{D}$ based on a well–understood (normal) semantics for $\mathcal{D}^{\mathsf{cd}}$. We will show that the (disjunctive) stable model semantics of $\mathcal{D}$ can be characterized by simply using the (normal) stable model semantics of $\mathcal{D}^{\mathsf{cd}}$. In a more refined approach we will use a special *fixpoint condition* based on $\mathcal{D}^{\mathsf{cd}}$ for defining a new semantics, called *stable state semantics*, which is able to give plausible interpretations for dd–databases for which standard semantics (e.g. stable or partial stable models) are *inconsistent*, i.e. do not provide any interpretations.

For some extensions of deductive databases without default negation we will show that it is possible to use the principle of reasoning known from the definite consequence operator $\mathcal{T}_{\mathcal{D}}$, as given by Lloyd [10]. We will define a *generic*

[1] *true, false, undefined, overdefined,* etc.

[2] possible instantiations could be $X_d = \neg p(a) \vee \neg q(b)$ and $X_c = p(a) \wedge q(b)$

consequence operator $\mathcal{T}_{\mathcal{D}}^{\odot,\geq} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ that is parameterized with a natural transformation, which maps $\mathcal{D}$ to a definite deductive database $\mathcal{D}^{\odot}$, and with a subsumption relation $\geq$. We will prove that $\mathcal{T}_{\mathcal{D}}^{\odot,\geq}$ is *monotonic* and *continuous* on the lattice $\langle 2^{\mathcal{X}}, \subseteq \rangle$. Then the disjunctive hyperresolution consequence operator $\mathcal{T}_{\mathcal{D}}^{s} : 2^{D_{HB\mathcal{D}}} \to 2^{D_{HB\mathcal{D}}}$ of Minker and Rajasekar [13] for dd–databases $\mathcal{D}$ without default negation – whose disjunctive Herbrand base is denoted by $D_{HB\mathcal{D}}$ – is formulated as a definite consequence operator $\mathcal{T}_{\mathcal{D}}^{\odot,\geq}$ for $\mathcal{X} = \{dis(\alpha) | \alpha \in D_{HB\mathcal{D}}\}$. The transformation $\mathcal{D}^{\odot}$ can be extended for the case of dd–databases with default negation by adding a definite representation of the *logical axioms* for binary resolution and subsumption and by adding suitable *closed–world–assumptions*, cf. Reiter [18].

The advantage of transforming a disjunctive deductive database $\mathcal{D}$ into a normal deductive database $\mathcal{D}^{\mathsf{cd}}$ is that it becomes possible to use *standard techniques* for defining a semantics and *standard tools* for evaluating a database under this semantics [4, 15, 19, 23, 26]. In a first, prototypical implementation we have used the system Smodels developed by Niemelä [15], which can compute well–founded and stable model semantics for normal deductive databases.

Also other semantics of disjunctive deductive databases have been characterized or computed based on *transformations*. E.g., Brass and Dix [1] have used *partial evaluation* for defining their *disjunctive well–founded semantics*; Seipel et al. [21] have used the tu–*transformation* for characterizing *partial stable models*; Fernández et al. [6] and Seipel [22] have used the *evidential transformation* for characterizing *total stable models*, and *evidential stable models*, respectively.

The rest of the paper is organized as follows: In Section 2 we present the syntax of dd–databases and the stable model semantics. In Section 3 we define the syntax of cd–databases, and we also define clausal Herbrand interpretations for dd–databases. In Section 4 we develop the generic $\mathcal{T}_{\mathcal{D}}$–operator, and we show how the disjunctive consequence operator $\mathcal{T}_{\mathcal{D}}^{s}$ can be formulated. Section 5 deals with a characterization of the stable models semantics using clausal logic programming, and a generalization that is called *stable state semantics*.

2 Basic Definitions and Notations

2.1 Syntax of Disjunctive Deductive Databases

Given a first order language $\mathcal{L}$, a *disjunctive deductive database (dd–database)* $\mathcal{D}$ consists of logical inference rules of the form

$$r = A_1 \vee \ldots \vee A_k \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, C_1 \wedge \ldots \wedge not\, C_n, \qquad (1)$$

where A_i, $i \in \langle 1, k \rangle$, B_i, $i \in \langle 1, m \rangle$, and C_i, $i \in \langle 1, n \rangle$, are atoms in the language $\mathcal{L}$, $k, m, n \in \mathbb{N}_0$, and *not* is the negation–by–default operator.[3] The set of all

[3] By $\mathbb{N}_+$ we denote the set $\{1, 2, 3, \ldots\}$ of positive natural numbers, whereas $\mathbb{N}_0$ denotes the set $\{0, 1, 2, \ldots\}$ of all natural numbers. $\langle n, m \rangle$ denotes the interval $\{n, n+1, \ldots, m\}$ of natural numbers.

ground instances of the rules in $\mathcal{D}$ is denoted by $gnd(\mathcal{D})$. A rule is called a *fact* if $m = n = 0$, and it is called a *denial rule* if $k = 0$. A rule (or database) is called *positive–disjunctive* if it does not contain default negation (i.e. $n = 0$). It is called *normal* if $k = 1$, and *definite* if $k = 1$ and $n = 0$. A rule r of the form (1) above is denoted for short as

$$r = \alpha \leftarrow \beta \wedge not \cdot \gamma,$$

where $\alpha = A_1 \vee \ldots \vee A_k$, $\beta = B_1 \wedge \ldots \wedge B_m$, and $\gamma = C_1 \vee \ldots \vee C_n$.[4] α is called the *head* of r, β is called the *positive body*, and $not \cdot \gamma$ is called the *negative body* of r.

The Herbrand base $HB_{\mathcal{D}}$ of a dd–database $\mathcal{D}$ contains all ground atoms over the language of $\mathcal{D}$, and $HL_{\mathcal{D}} = HB_{\mathcal{D}} \cup \{ \neg A \mid A \in HB_{\mathcal{D}} \}$ contains all literals over $HB_{\mathcal{D}}$. The semantics of dd–databases will be based on a set of ground disjunctions, a so–called Herbrand state, or a set of Herbrand interpretations.

1. The *disjunctive*, the *negative*, the *general*, and the *conjunctive Herbrand base* of $\mathcal{D}$ are given by $DHB_{\mathcal{D}}$, $NHB_{\mathcal{D}}$, $GHB_{\mathcal{D}}$, and $CHB_{\mathcal{D}}$, respectively:

$$\begin{aligned}
DHB_{\mathcal{D}} &= \{ A_1 \vee \ldots \vee A_k \mid A_i \in HB_{\mathcal{D}}, \text{ for } i \in \langle 1, k \rangle, \ k \in \mathbb{N}_0 \}, \\
NHB_{\mathcal{D}} &= \{ \neg A_1 \vee \ldots \vee \neg A_k \mid A_i \in HB_{\mathcal{D}}, \text{ for } i \in \langle 1, k \rangle, \ k \in \mathbb{N}_0 \}, \\
GHB_{\mathcal{D}} &= \{ L_1 \vee \ldots \vee L_k \mid L_i \in HL_{\mathcal{D}}, \text{ for } i \in \langle 1, k \rangle, \ k \in \mathbb{N}_0 \}, \\
CHB_{\mathcal{D}} &= \{ L_1 \wedge \ldots \wedge L_k \mid L_i \in HL_{\mathcal{D}}, \text{ for } i \in \langle 1, k \rangle, \ k \in \mathbb{N}_0 \}.
\end{aligned}$$

2. An *Herbrand state* is a set S of formulas. It is called *disjunctive*, *negative*, *general* and *conjunctive*, if $S \subseteq DHB_{\mathcal{D}}$, $S \subseteq NHB_{\mathcal{D}}$, $S \subseteq GHB_{\mathcal{D}}$, and $S \subseteq CHB_{\mathcal{D}}$. respectively.

We will sometimes also write $HB_{\mathcal{L}}$, $HL_{\mathcal{L}}$, $DHB_{\mathcal{L}}$, $NHB_{\mathcal{L}}$, $GHB_{\mathcal{L}}$, and $CHB_{\mathcal{L}}$, if we are interested in the language $\mathcal{L}$ rather than the database $\mathcal{D}$. Disjunctions and conjunctions will be treated like sets of literals. Thus, we will identify a disjunction $L_1 \vee \ldots \vee L_k \in GHB_{\mathcal{D}}$ with its permutations $L_{\pi(1)} \vee \ldots \vee L_{\pi(k)}$. The empty disjunction (conjunction) with $k = 0$ literals will be denoted by $\Box_d$ ($\Box_c$). Note that $\Box_d$ is an element of all of $DHB_{\mathcal{D}}$, $NHB_{\mathcal{D}}$, and $GHB_{\mathcal{D}}$, and that $\Box_c$ is an element of $CHB_{\mathcal{D}}$. Note also that singleton disjunctions or conjunctions with $k = 1$ literals are identified with literals; as a consequence we get $GHB_{\mathcal{D}} \cap CHB_{\mathcal{D}} = HL_{\mathcal{D}}$. The common *subsumption* relation for ground disjunctions $\alpha = L_1 \vee \ldots \vee L_n \in GHB_{\mathcal{D}}$, $\alpha' = L'_1 \vee \ldots \vee L'_m \in GHB_{\mathcal{D}}$, is defined by $\alpha \trianglerighteq \alpha'$ iff $\{ L_1, \ldots, L_n \} \subseteq \{ L'_1, \ldots, L'_m \}$. The *canonical form* of a general Herbrand state $S \subseteq GHB_{\mathcal{D}}$ is defined as the set $can_{\trianglerighteq}(S)$ of minimal elements of S w.r.t. subsumption, and the *expanded form* of S is defined as the set $exp_{\trianglerighteq}(S)$ of all $\alpha \in GHB_{\mathcal{D}}$ that are subsumed by some $\alpha' \in S$:

$$\begin{aligned}
can_{\trianglerighteq}(S) &= \{ \alpha \in S \mid \ \nexists \alpha' \in S : \alpha' \neq \alpha, \alpha' \trianglerighteq \alpha \}, \\
exp_{\trianglerighteq}(S) &= \{ \alpha \in GHB_{\mathcal{D}} \mid \exists \alpha' \in S : \alpha' \trianglerighteq \alpha \}.
\end{aligned}$$

[4] Note that γ is a disjunction, and, according to De Morgan's law, $not \cdot \gamma$ is taken to be a conjunction.

Obviously, it holds $can_{\trianglerighteq}(S) \subseteq S \subseteq exp_{\trianglerighteq}(S)$. For a literal $L \in HL_{\mathcal{D}}$ we define the *formal complement* $\neg \cdot L = \neg A$, if $L = A \in HB_{\mathcal{D}}$ is an atom, and $\neg \cdot L = A$, if $L = \neg A \in HL_{\mathcal{D}}$ is a negated atom. For a general disjunction $\alpha = L_1 \vee \ldots \vee L_k \in GHB_{\mathcal{D}}$ we define the conjunction

$$\neg \cdot \alpha = \neg \cdot (L_1 \vee \ldots \vee L_k) = \neg \cdot L_1 \wedge \ldots \wedge \neg \cdot L_k \in CHB_{\mathcal{D}}$$

to be the *formal complement* of α according to the De Morgan pattern, and likewise we define $\neg \cdot \beta = \neg \cdot L_1 \vee \ldots \vee \neg \cdot L_k$ for $\beta = L_1 \wedge \ldots \wedge L_k \in CHB_{\mathcal{D}}$.

2.2 Stable Model Semantics

An *Herbrand interpretation* of a dd–database $\mathcal{D}$ is given by a subset $I \subseteq HB_{\mathcal{D}}$. It defines a mapping I: $HB_{\mathcal{D}} \to \{\mathsf{t}, \mathsf{f}\}$ assigning a truth value "t" (*true*) or "f" (*false*) to each $A \in HB_{\mathcal{D}}$: $I(A) = \mathsf{t} \Leftrightarrow A \in I$. [5] I *satisfies* of a ground rule $r = \alpha \leftarrow \beta \wedge not \cdot \gamma$ if

$$I(\beta) \wedge I(not \cdot \gamma) = \mathsf{t} \Rightarrow I(\alpha) = \mathsf{t},$$

where $I(not \cdot \gamma) = \neg I(\gamma)$. I is an *Herbrand model* of $\mathcal{D}$ if I satisfies all ground instances $r \in gnd\,(\mathcal{D})$ of the rules of $\mathcal{D}$. I is a *minimal model* of $\mathcal{D}$ if I is an Herbrand model of $\mathcal{D}$ and there is no other Herbrand model J of $\mathcal{D}$ such that $J \subsetneq I$. The set of all minimal models of $\mathcal{D}$ is denoted by $\mathcal{MM}_2(\mathcal{D})$. For positive–disjunctive deductive databases $\mathcal{D}$ the semantics is given by $\mathcal{MM}_2(\mathcal{D})$.

The semantics of dd–databases in general is defined based on a more general class of Herbrand models, the *stable models*. These are defined in terms of the minimal models M of the *Gelfond–Lifschitz transformation* (GL–transformation, [8, 16]) of $\mathcal{D}$, which is a special ground positive–disjunctive database $\mathcal{D}^M$ derived from those rules of $gnd\,(\mathcal{D})$ whose negative body is true in M (i.e. $C_i \notin M$, for all C_i in $not \cdot \gamma = not\, C_1 \wedge \ldots \wedge not\, C_n$).

Definition 2.1 (Stable Models, [8, 16]).
Let M be a Herbrand interpretation of a disjunctive deductive database $\mathcal{D}$.

1. The GL–transformation $\mathcal{D}^M$ of $\mathcal{D}$ w.r.t. M is

$$\mathcal{D}^M = \{\ \alpha \leftarrow \beta \mid \text{there exists } \alpha \leftarrow \beta \wedge not \cdot \gamma \in gnd\,(\mathcal{D}),\ \text{such that } M(not \cdot \gamma) = \mathsf{t}\ \}.$$

2. M is called *stable model* of $\mathcal{D}$ if $M \in \mathcal{MM}_2(\mathcal{D}^M)$. The set of all stable models of $\mathcal{D}$ is denoted by $\mathcal{STABLE}_2(\mathcal{D})$.

[5] For $A_i \in HB_{\mathcal{D}}$, $i \in \langle\, 1, k \,\rangle$, and a connective $\oplus \in \{\vee, \wedge\}$ we define $I(\, A_1 \oplus \ldots \oplus A_k\,) = I(A_1) \oplus \ldots \oplus I(A_k)$. For $k = 0$, the empty disjunction $\square_d$ (i.e. $\oplus = \vee$) evaluates to $I(\square_d) = \mathsf{f}$, whereas the empty conjunction $\square_c$ (i.e. $\oplus = \wedge$) evaluates to $I(\square_c) = \mathsf{t}$.

Example 2.1. For the dd–database

$$\mathcal{D}_{win} = \{\ win(X) \leftarrow move(X,Y) \wedge not\ win(Y)\ \} \cup S,$$
$$S = \{\ move(a,b) \vee move(a,c),\ move(b,d),\ move(c,d)\ \},$$

we get a set with two stable models, which coincide on the subset I of atoms:

$$\mathcal{STABLE}_2(\mathcal{D}_{win}) = \{\ \{\ move(a,b)\ \} \cup I,\ \{\ move(a,c)\ \} \cup I\ \},$$
$$I = \{\ move(b,d),\ move(c,d),\ win(b),\ win(c)\ \}.$$

For $M = \{\ move(a,b)\ \} \cup I$ we get that $M(not\ win(y)) = \mathsf{t}$ iff $y \in \{a,d\}$, and thus we get $M \in \mathcal{MM}_2(\mathcal{D}^M_{win})$ for the GL–transformation $\mathcal{D}^M_{win}$:

$$\mathcal{D}^M_{win} = \{\ win(x) \leftarrow move(x,y) \mid x \in \{a,b,c,d\},\ y \in \{a,d\}\ \} \cup S.$$

All of the atoms in I and the disjunction $move(a,b) \vee move(a,c)$ are consequences of $\mathcal{D}_{win}$ under the stable model semantics.

3 Clausal Deductive Databases

In the context of default negation and disjunction one can generalize the notion of a (total) Herbrand interpretation in two ways.

Firstly, one can allow for a third truth value "u" (*undefined*), which leads to partial Herbrand interpretations I: $HB_{\mathcal{D}} \rightarrow \{\ \mathsf{t}, \mathsf{f}, \mathsf{u}\ \}$. If a semantics is given by a set $\mathcal{I}$ of partial Herbrand interpretations, then a general Herbrand state $S \subseteq GHB_{\mathcal{D}}$ containing all disjunctive consequences $\alpha \in GHB_{\mathcal{D}}$ that hold in all $I \in \mathcal{I}$ can be assigned to $\mathcal{D}$. Partial Herbrand interpretations have been used by Przymusinski, cf. [16], who generalized the concept of stable model to *partial stable models*. There exists a characterization of the partial stable models of a disjunctive deductive database $\mathcal{D}$ in terms of the stable models of a naturally transformed database $\mathcal{D}^{\mathsf{tu}}$, cf. [21].

Secondly, a semantics can also be given by a set $\mathcal{S}$ of so–called *clausal Herbrand interpretations*, which are special general Herbrand states, without refering to total or partial Herbrand interpretations at all. In this paper we will introduce the new concept of a *clausal deductive databases*, which are special normal deductive databases over a restricted set of predicate symbols. In Section 5, we will transform a dd–database $\mathcal{D}$ into a corresponding cd–database $\mathcal{D}^{\mathsf{cd}}$ and then derive the clausal Herbrand models of $\mathcal{D}$ by simply applying the standard stable model semantics to $\mathcal{D}^{\mathsf{cd}}$.

3.1 Partial and Clausal Herbrand Interpretations

A *partial Herbrand interpretation* of a disjunctive deductive database $\mathcal{D}$ is given by a subset $I \subseteq HL_{\mathcal{D}}$, such that I does not contain both an atom $A \in HB_{\mathcal{D}}$ and

its negation $\neg A$, i.e. $I \cap \neg I = \emptyset$. It defines a mapping I: $HB_{\mathcal{D}} \to \{ \mathsf{t}, \mathsf{f}, \mathsf{u} \}$ on ground atoms by

$$I(A) = \begin{cases} \mathsf{t}, \text{ iff } A \in I \\ \mathsf{f}, \text{ iff } \neg A \in I \\ \mathsf{u}, \text{ otherwise.} \end{cases}$$

If for each atom $A \in HB_{\mathcal{D}}$, I contains either A or $\neg A$, then I defines a *total Herbrand interpretation* or Herbrand interpretation for short. In this case, it is sufficient to represent I by its set of true atoms (since there are no undefined atoms), which we did in Section 2. The truth values of disjunctions $\alpha \in G_{HB_{\mathcal{D}}}$ and conjunctions $\beta \in C_{HB_{\mathcal{D}}}$ are defined based on the *Boolean operations* "$\vee$", "$\wedge$" and "$\neg$" in three–valued logic, which are given in Figure 1. The *truth ordering* on the three truth values is given by $\mathsf{f} \leq_{\mathsf{t}} \mathsf{u} \leq_{\mathsf{t}} \mathsf{t}$.

$\wedge$	t	f	u
t	t	f	u
f	f	f	f
u	u	f	u

$\vee$	t	f	u
t	t	t	t
f	t	f	u
u	t	u	u

$\neg$	
t	f
f	t
u	u

Fig. 1. Boolean operations in three–valued logic

A *clausal Herbrand interpretation* is given by a general Herbrand state $S \subseteq G_{HB_{\mathcal{D}}}$ that is closed under binary resolution and subsumption. Obviously, we can *abbreviate* a clausal Herbrand interpretation S simply by its set $can_{\trianglerighteq}(S)$ of minimal elements w.r.t. subsumption. If $can_{\trianglerighteq}(S) \subseteq HL_{\mathcal{D}}$, then we can consider S as a partial Herbrand interpretation $I = can_{\trianglerighteq}(S)$ of $\mathcal{D}$. Conversely, a partial Herbrand interpretation I can be considered as a special clausal Herbrand interpretation $exp_{\trianglerighteq}(I)$. Every set $\mathcal{S}$ of clausal (or partial) Herbrand interpretations defines a clausal Herbrand interpretation $\mathcal{S}^{\cap} = \cap_{S \in \mathcal{S}} S \subseteq G_{HB_{\mathcal{D}}}$, where $\mathcal{S}^{\cap} = G_{HB_{\mathcal{D}}}$, if $\mathcal{S} = \emptyset$

The *semantics* of a disjunctive deductive database $\mathcal{D}$ will be given by a set $\mathcal{S}$ of clausal Herbrand interpretations, and a formula α is a consequence under $\mathcal{S}$ if $\alpha \in \mathcal{S}^{\cap}$.

3.2 Syntax of Clausal Deductive Databases

Consider a first order language $\mathcal{L}$ and the set $Lit_{\mathcal{L}}$ of all literals that can be built in the language $\mathcal{L}$. Additionally, consider a set $\mathcal{V}$ of variable symbols for formulas. Disjunctive (conjunctive) *formulas* are defined inductively: every atom $A \in Lit_{\mathcal{L}}$ and every $X \in \mathcal{V}$ is both a disjunctive and a conjunctive formula, and, if $t_1, \ldots, t_n$ are disjunctive (conjunctive) formulas, then so is $t_1 \vee \ldots \vee t_n$ ($t_1 \wedge \ldots \wedge t_n$).

A *cd–atom* is given by $dis(t_d)$ or $con(t_c)$, where t_d is a disjunctive formula and t_c is a conjunctive formula. If X_d, X'_d, X''_d are disjunctive formulas, and

X_c, X_c', X_c'' are conjunctive formulas, and $X \in Lit_{\mathcal{L}} \cup \mathcal{V}$, then the following are *built–in atoms* (bi–atoms):

$$\begin{array}{l} positive(X_d),\ negative(X_d), \\ subsumes(X_d, X_d'),\ resolvent(X_d, X_d', X_d''),\ empty_dis(X_d), \\ conjunction(X_c, X_c', X_c''),\ empty_con(X_c), \\ literal(X),\ complement(X_d, X_c). \end{array}$$

A *clausal deductive database* (*cd–database*) $\mathcal{C}$ consists of *cd–rules* of the form

$$r = A \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, C_1 \wedge \ldots \wedge not\, C_n, \tag{2}$$

where A is a cd–atom, and the B_i, $i \in \langle 1, m \rangle$, and C_i, $i \in \langle 1, n \rangle$, are cd–atoms or bi–atoms, and *not* is the negation–by–default operator. A cd–rule is called a cd–fact if $m = n = 0$. For example, the following is a cd–rule:

$$\begin{array}{l} dis(X_d' \vee \neg p(Y, a)) \leftarrow \\ \quad dis(X_d \vee q(c)) \wedge subsumes(X_d, X_d') \wedge \\ \quad complement(X_d, X_c \wedge \neg q(d)) \wedge not\, con(X_c \wedge \neg q(d)) \wedge \\ \quad not\, literal(X_c). \end{array}$$

Built–in predicate symbols are not allowed in rule heads, although built–in predicates could of course be defined by logical rules as well. For simplicity reasons, however, we assume that their definition is given by a standard *built–in Herbrand interpretation*

$$\begin{array}{l} I_{\mathcal{L}}^{bi} = \{\, positive(\alpha) \mid \alpha \in D_{HB_{\mathcal{L}}} \,\} \cup \{\, negative(\alpha) \mid \alpha \in N_{HB_{\mathcal{L}}} \,\} \cup \\ \quad \{\, literal(\gamma) \mid \gamma \in H_{B_{\mathcal{L}}} \,\} \cup \{\, complement(\alpha, \neg \cdot \alpha) \mid \alpha \in G_{HB_{\mathcal{L}}} \,\} \cup \\ \quad \{\, empty_dis(\square_d) \,\} \cup \{\, empty_con(\square_c) \,\} \cup \\ \quad \{\, subsumes(\alpha_1, \alpha_2) \mid \alpha_1, \alpha_2 \in G_{HB_{\mathcal{L}}},\ \alpha_1 \trianglerighteq \alpha_2 \,\} \cup \\ \quad \{\, resolvent(A \vee \alpha_1, \neg A \vee \alpha_2, \alpha_1 \vee \alpha_2) \mid A \in H_{B_{\mathcal{L}}},\ \alpha_1, \alpha_2 \in G_{HB_{\mathcal{L}}} \,\} \cup \\ \quad \{\, conjunction(\beta_1, \beta_2, \beta_1 \wedge \beta_2) \mid \beta_1, \beta_2 \in C_{HB_{\mathcal{L}}} \,\}. \end{array}$$

The built–in predicate symbols *conjunction* and *resolvent* are not really necessary in cd–databases, since they can be replaced using the junctors "$\wedge$" and "$\vee$", respectively. However, it will be very convenient to have them.

The *ground instance gnd* ($\mathcal{C}$) of a cd–database $\mathcal{C}$ is obtained replacing all variable symbols for formulas by ground formulas from $G_{HB_{\mathcal{L}}} \cup C_{HB_{\mathcal{L}}}$ and by also grounding the literals from $Lit_{\mathcal{L}}$ in $\mathcal{C}$, such that the resulting rules are proper cd–rules.[6] For a general Herbrand state $S \subseteq G_{HB_{\mathcal{L}}}$ we define $dis(S) = \{\, dis(\alpha) \mid \alpha \in S \,\}$ and $con(S) = \{\, con(L_1 \wedge \ldots \wedge L_k) \mid L_i \in S \cap H_{L_{\mathcal{L}}},$ for $i \in \langle 1, k \rangle,\ k \in \mathbb{N}_0 \,\}$. Note that $con(\square_c) \in con(S)$. Finally, $S^{\mathsf{cd}} = dis(S) \cup con(S) \cup I_{\mathcal{L}}^{bi}$ denotes the total Herbrand interpretation of $\mathcal{C}$ that is derived from S.

[6] This implies that the obvious typing condition for variable symbols for formulas must be obeyed.

4 Clausal Deductive Databases without Default Negation

The idea of clausal logic programming without default negation will be based on the observation that the disjunctive hyperresolution consequence operator, which has been defined by Minker and Rajasekar [13] for positive–disjunctive deductive databases $\mathcal{D}$, can be formulated using a generic definite hyperresolution consequence operator $\mathcal{T}_{\mathcal{D}}^{\odot,\geq}$, which is parameterized with a suitable subsumption relation. Thus reasoning with disjunctive hyperresolution can be reduced to reasoning with definite hyperresolution on a suitable domain. For hypothetical and for probabilistic deductive databases, cf. [3, 25], the generic operator $\mathcal{T}_{\mathcal{D}}^{\odot,\geq}$ can be used, too, which will be shown in another paper.

We will consider the following definition[7] of the disjunctive hyperresolution consequence operator $\mathcal{T}_{\mathcal{D}}^{s} : \ 2^{DHB_{\mathcal{D}}} \to 2^{DHB_{\mathcal{D}}}$:

$$\begin{aligned}\mathcal{T}_{\mathcal{D}}^{s}(S) = \{\ \alpha \vee \alpha_1 \vee \ldots \vee \alpha_m \in DHB_{\mathcal{D}} \mid \\ \text{there exists a rule } \alpha \leftarrow B_1 \wedge \ldots \wedge B_m \in gnd\,(\mathcal{D}), \\ \text{such that } \{\ B_1 \vee \alpha_1, \ldots, B_m \vee \alpha_m\ \} \subseteq S\ \}.\end{aligned}$$

Minker and Rajasekar [13] have shown that $\mathcal{T}_{\mathcal{D}}^{s}$ is *monotonic* and *continuous* on the complete lattice[8] $\langle 2^{DHB_{\mathcal{D}}}, \subseteq \rangle$. In [20], $\mathcal{T}_{\mathcal{D}}^{s}$ has been related to the model–generation operator of Fernández and Minker [7], and it was shown that the latter is monotonic but not continuous.

4.1 The Generic $\mathcal{T}_{\mathcal{D}}$–Operator

Consider a deductive database $\mathcal{D}$ without default negation (e.g. a positive–disjunctive, probabilistic or hypothetical deductive database, cf. [20, 25, 3]), and a partial ordering $\mathcal{O}_{\mathcal{D}} = \langle \mathcal{X}, \geq \rangle$ that is assigned to $\mathcal{D}$. The generic $\mathcal{T}_{\mathcal{D}}$–operator

$$\mathcal{T}_{\mathcal{D}}^{\odot,\geq} : \ 2^{\mathcal{X}} \to 2^{\mathcal{X}}$$

is based on a transformation "$\odot$" and the subsumption relation "$\geq$" of $\mathcal{O}_{\mathcal{D}}$. Firstly, $\mathcal{D}$ is transformed, such that $gnd\,(\mathcal{D}^{\odot})$ consists of definite ground rules $\alpha \leftarrow \beta_1 \wedge \ldots \wedge \beta_m$, where $\{\ \alpha, \beta_1, \ldots, \beta_m\ \} \subseteq \mathcal{X}$. Secondly, a set $X \subseteq \mathcal{X}$ is expanded to a set $exp_{\geq}(X) = \{\ \alpha \in \mathcal{X} \mid \exists\, \alpha' \in X : \ \alpha' \geq \alpha\ \} \subseteq \mathcal{X}$, cf. Subsection 2.1. Then the $\mathcal{T}_{\mathcal{D}}$–operator induced by $\langle \odot, \geq \rangle$ is

$$\begin{aligned}\mathcal{T}_{\mathcal{D}}^{\odot,\geq}(X) = \{\ \alpha \in \mathcal{X} \mid \text{there exists } \alpha \leftarrow \beta_1 \wedge \ldots \wedge \beta_m \in gnd\,(\mathcal{D}^{\odot}), \\ \text{such that } \{\ \beta_1, \ldots, \beta_m\ \} \subseteq exp_{\geq}(X)\ \}.\end{aligned}$$

[7] It is equivalent to the definition of [13]. Since disjunctions are considered as sets of atoms, it is not necessary to work with smallest factors.

[8] For a definition of monotonic and continuous operators $\mathcal{T}$ and complete lattices see Lloyd [10]. Continuity ensures that fixpoint iteration terminates with the least fixpoint of $\mathcal{T}$ within at most ω steps.

Theorem 4.1 (Properties of the Generic $\mathcal{T}_\mathcal{D}$–Operator).
The operator $\mathcal{T}_\mathcal{D}^{\odot,\geq} : \ 2^\mathcal{X} \to 2^\mathcal{X}$ is *monotonic* and *continuous* on the complete lattice $\langle\, 2^\mathcal{X}, \subseteq \,\rangle$.

The proof is given in the appendix. The properties of fixpoint iteration with subsumption have also been analysed in [9] with special interest in the process of *differential* fixpoint iteration.

4.2 Positive–Disjunctive Deductive Databases

For positive–disjunctive deductive databases $\mathcal{D}$ we consider $\mathcal{X} = dis(D_{HB_\mathcal{D}})$, and we will present two natural ways for formulating the disjunctive consequence operator $\mathcal{T}_\mathcal{D}^s$ as a generic $\mathcal{T}_\mathcal{D}$–operator. This gives an alternative proof that $\mathcal{T}_\mathcal{D}^s$ is monotonic and continuous. Moreover, in the next section the second transformation will be extended to dd–databases with default negation.

Transformation 1. The first approach uses the built–in predicate *subsumes*:

$$\mathcal{D}^\odot = \{\ dis(X) \leftarrow dis(B_1 \vee X) \wedge \ldots \wedge dis(B_m \vee X) \wedge subsumes(\alpha, X) \mid \text{there exists } \alpha \leftarrow B_1 \wedge \ldots \wedge B_m \in \mathcal{D}\ \} \ \cup\ I_\mathcal{D}^{bi},$$

where $X \in \mathcal{V}$ is a variable symbol for formulas, and the common concept of subsumption for ground disjunctions $\alpha, \alpha' \in D_{HB_\mathcal{D}}$ is used, i.e. $\alpha \geq \alpha'$ iff $\alpha \trianglerighteq \alpha'$. Then the $\mathcal{T}_\mathcal{D}^s$–operator on disjunctive Herbrand states S can be characterized using the generic $\mathcal{T}_\mathcal{D}$–operator and the expansion for disjunctive Herbrand states:

$$\mathcal{T}_\mathcal{D}^{\odot,\trianglerighteq}(dis(S) \cup I_\mathcal{D}^{bi}) = dis(exp_\trianglerighteq(\mathcal{T}_\mathcal{D}^s(S))) \cup\ I_\mathcal{D}^{bi}.$$

Example 4.1. For $\mathcal{D} = \{\ a \vee b \leftarrow c,\ c \vee d\ \}$ we get

$$\mathcal{D}^\odot = \{\ dis(X) \leftarrow dis(c \vee X) \wedge subsumes(a \vee b, X),\ dis(X) \leftarrow subsumes(c \vee d, X)\ \} \ \cup\ I_\mathcal{D}^{bi}.$$

The rule $dis(X) \leftarrow dis(B_1 \vee X) \wedge subsumes(\alpha, X)$ that is derived from $a \vee b \leftarrow c$ has one disjunction $B_1 \vee X$ in its body, which extends $B_1 = c$ by some disjunction X that must be subsumed by $\alpha = a \vee b$. In a fixpoint iteration with $\mathcal{T}_\mathcal{D}^{\odot,\geq}$, first the fact $dis(c \vee d)$ is derived (amoung other facts). In the second iteration, using the ground instance $dis(a \vee b \vee d) \leftarrow dis(c \vee a \vee b \vee d) \wedge subsumes(a \vee b, a \vee b \vee d)$, which is obtained for $X = a \vee b \vee d$, and the fact $dis(c \vee d)$, it is possible to derive the fact $dis(a \vee b \vee d)$, since $c \vee d \trianglerighteq c \vee a \vee b \vee d$ and $subsumes(a \vee b, a \vee b \vee d) \in I_\mathcal{D}^{bi}$.

Transformation 2. Another way of transforming a positive–disjunctive deductive database $\mathcal{D}$ is

$$\mathcal{D}^\odot = \{\ dis(\alpha \vee X_1 \vee \ldots \vee X_m) \leftarrow dis(B_1 \vee X_1) \wedge \ldots \wedge dis(B_m \vee X_m) \mid \text{there exists } \alpha \leftarrow B_1 \wedge \ldots \wedge B_m \in \mathcal{D}\ \},$$

with variable symbols $X_i \in \mathcal{V}$ $(i \in \mathbb{N}_+)$ for formulas. If subsumption "$\geq$" for ground disjunctions $\alpha, \alpha' \in \mathit{DHB}_\mathcal{D}$ is defined as identity (where disjunctions are treated as sets), i.e. $\alpha \geq \alpha'$ iff $\alpha = \alpha'$, then the disjunctive consequence operator $\mathcal{T}^s_\mathcal{D}$ on disjunctive Herbrand states S coincides with the generic $\mathcal{T}_\mathcal{D}$–operator:

$$\mathcal{T}^{\odot,\trianglerighteq}_\mathcal{D}(S) = \mathit{dis}(\mathcal{T}^s_\mathcal{D}(S)).$$

Example 4.2. For $\mathcal{D} = \{\ a \vee b \leftarrow c,\ c \vee d\ \}$ we get

$$\mathcal{D}^\odot = \{\ \mathit{dis}(a \vee b \vee X_1) \leftarrow \mathit{dis}(c \vee X_1),\ \mathit{dis}(c \vee d)\ \}.$$

In a fixpoint iteration with $\mathcal{T}^{\odot,\trianglerighteq}_\mathcal{D}$, first the fact $\mathit{dis}(c \vee d)$ is derived. In the second iteration, using the ground instance $\mathit{dis}(a \vee b \vee d) \leftarrow \mathit{dis}(c \vee d)$ of the first rule (obtained for $X_1 = d$) and the fact $\mathit{dis}(c \vee d)$, it is possible to derive the fact $\mathit{dis}(a \vee b \vee d)$.

5 Clausal Deductive Databases with Default Negation

We will transform a disjunctive deductive database $\mathcal{D}$ into a corresponding clausal deductive database $\mathcal{D}^{\mathsf{cd}} = \mathcal{D}^\odot \cup \mathcal{D}^{\mathit{logic}} \cup \mathcal{D}^{\mathit{cwa}} \cup I^{\mathit{bi}}_\mathcal{D}$, which we call the cd–*transformation* of $\mathcal{D}$. It consists of four parts: a set

$$\begin{aligned}\mathcal{D}^\odot = \{\ \mathit{dis}(\alpha \vee X_1 \vee \ldots \vee X_m) \leftarrow\ & \\ & \mathit{dis}(B_1 \vee X_1) \wedge \ldots \mathit{dis}(B_m \vee X_m) \wedge \\ & \mathit{not}\ \mathit{con}(C_1) \wedge \ldots \wedge \mathit{not}\ \mathit{con}(C_n)\ | \\ & \text{there exists}\ \ \alpha \leftarrow B_1 \wedge \ldots \wedge B_m \wedge \mathit{not}\, C_1 \wedge \ldots \wedge \mathit{not}\, C_n \in \mathcal{D}\ \},\end{aligned}$$

of rules obtained from $\mathcal{D}$ (by extending *Transformation 2* of Section 4.2), with variable symbols $X_i \in \mathcal{V}$ $(i \in \mathbb{N}_+)$ for formulas, a set $\mathcal{D}^{\mathit{logic}}$ of logical rules for relating the predicate symbols *dis* and *con*, a set $\mathcal{D}^{\mathit{cwa}} = \{\ r^{\mathit{cwa}}_{\mathit{neg}}\ \}$ with a *closed–world–assumption*, and the built–in interpretation $I^{\mathit{bi}}_\mathcal{D}$. The cd–facts in $I^{\mathit{bi}}_\mathcal{D}$ and the cd–rules in $\mathcal{D}^{\mathit{logic}} \cup \mathcal{D}^{\mathit{cwa}}$ are *generic*; they depend only on the Herbrand base $\mathit{HB}_\mathcal{D}$, and not on the structure of the rules of $\mathcal{D}$.

We will prove that the stable models of the cd–transformation $\mathcal{D}^{\mathsf{cd}}$ correspond to the stable models of $\mathcal{D}$. Thus, we have shown a very natural way of computing the stable models of a disjunctive deductive database $\mathcal{D}$ based on the stable models of a normal deductive database $\mathcal{D}^{\mathsf{cd}}$. Furthermore, we will extend the set $\mathcal{STABLE}_2(\mathcal{D})$ to a set $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D})$ of so–called *stable states*, which are clausal Herbrand interpretations of $\mathcal{D}$ that are derived using a special *fixpoint condition* for $\mathcal{D}^{\mathsf{cd}}$. Stable states form a useful new semantics for dd–databases, which is different from the stable model semantics. In contrast, the well–founded model I_{wfs} of $\mathcal{D}^{\mathsf{cd}}$, which is the least partial stable model in the knowledge ordering, is not suitable for defining a state semantics.

5.1 The Generic Rules

The following cd–rules represent *closed–world–assumptions*; they allow for deriving certain disjunctions if their complemantary conjunctions cannot be derived:

$$\begin{aligned}
r_{neg}^{cwa} &= dis(X_d) \leftarrow \\
&\qquad negative(X_d) \wedge complement(X_d, X_c) \wedge not\, con(X_c), \\
r_{pos}^{cwa} &= dis(X_d) \leftarrow \\
&\qquad positive(X_d) \wedge complement(X_d, X_c) \wedge not\, con(X_c), \\
r_{gen}^{cwa} &= dis(X_d) \leftarrow \\
&\qquad complement(X_d, X_c) \wedge not\, con(X_c).
\end{aligned}$$

In $\mathcal{D}^{\mathsf{cd}}$ we will only use the rule r_{neg}^{cwa} for deriving negative disjunctions $\alpha \in N_{HB_{\mathcal{D}}}$. In a more general framework for extended–disjunctive deductive databases ([14]) one could as well derive positive or general disjunctions by a closed–world–assumption.

The following axiomatic clauses in the set

$$\mathcal{D}^{logic} = \{\, r_{res}^{logic},\ r_{sub}^{logic},\ r_{con}^{logic},\ r_{lit}^{logic},\ r_{emp}^{logic} \,\}$$

describe the correlation between the atoms for disjunctions and conjunctions:

$$\begin{aligned}
r_{res}^{logic} &= dis(X_d) \leftarrow \\
&\qquad dis(X'_d) \wedge dis(X''_d) \wedge resolvent(X'_d, X''_d, X_d), \\
r_{sub}^{logic} &= dis(X_d) \leftarrow \\
&\qquad dis(X'_d) \wedge subsumes(X'_d, X_d), \\
\\
r_{con}^{logic} &= con(X_c) \leftarrow \\
&\qquad con(X'_c) \wedge con(X''_c) \wedge conjunction(X'_c, X''_c, X_c), \\
r_{lit}^{logic} &= con(X_c) \leftarrow \\
&\qquad literal(X_c) \wedge dis(X_c), \\
r_{emp}^{logic} &= con(X_c) \leftarrow \\
&\qquad empty_con(X_c).
\end{aligned}$$

5.2 Two–Valued Analysis of $\mathcal{D}^{\mathsf{cd}}$

We want to investigate the total Herbrand models I of the clausal deductive database $\mathcal{D}^{\mathsf{cd}}$. Since I is total, it must assign one of the truth values t or f to each of the atoms $dis(\alpha)$ and $con(\beta)$, for $\alpha \in G_{HB_{\mathcal{D}}}$, $\beta \in C_{HB_{\mathcal{D}}}$ – in particular for literals $A, \neg A \in H_{L_{\mathcal{D}}}$. Due to the subsumption rule r_{sub}^{logic} it must hold $I(\alpha_1 \vee \alpha_2) = I(\alpha_1) \vee I(\alpha_2)$, for all $\alpha_1, \alpha_2 \in G_{HB_{\mathcal{D}}}$. If $I(dis(\square_d)) = \mathsf{t}$, then we define $I^{\otimes} = J^{\top}$, where $J^{\top}(A) = \top$, for all $A \in H_{B_{\mathcal{D}}}$, with a fourth truth value

$\top$ (*overdefined*) denoting *inconsistency*. $J^\top$ is considered as a special total Herbrand interpretation of $\mathcal{D}$, where all atoms A are both true and false, and where $J^\top(\Box_d) = \mathsf{t}$. If $I(dis(\Box_d)) = \mathsf{f}$, then I can be represented by an (ordinary) total Herbrand interpretation $I^\otimes$ of $\mathcal{D}$, where

$$I^\otimes(A) = I(dis(A)), \text{ for all } A \in \mathit{HB}_\mathcal{D}.$$

Conversely, a total Herbrand interpretation J of $\mathcal{D}$ defines an Herbrand interpretation $J^\odot$ of $\mathcal{D}^{\mathsf{cd}}$ by

$$\begin{aligned} J^\odot(dis(\alpha)) &= J(\alpha), \quad \text{for all } \alpha \in \mathit{GHB}_\mathcal{D}, \\ J^\odot(con(\beta)) &= J(\beta), \quad \text{for all } \beta \in \mathit{CHB}_\mathcal{D}. \end{aligned}$$

For a total Herbrand model I of $\mathcal{D}^{\mathsf{cd}}$, it holds $(I^\otimes)^\odot = I$, and for a total Herbrand model J of $\mathcal{D}$, it holds $(J^\odot)^\otimes = J$. A set $\mathcal{I}$ of total Herbrand models of $\mathcal{D}^{\mathsf{cd}}$ can be represented by the set $\mathcal{I}^\otimes = \{\, I^\otimes \mid I \in \mathcal{I} \,\}$ of Herbrand interpretations of $\mathcal{D}$.

Theorem 5.1 (Characterization of Stable Models).
Given a disjunctive deductive database $\mathcal{D}$, then it holds

$$\mathcal{STABLE}_2(\mathcal{D}^{\mathsf{cd}})^\otimes = \mathcal{STABLE}_2(\mathcal{D}) \cup \mathcal{J}, \quad \text{where } \mathcal{J} = \emptyset \text{ or } \mathcal{J} = \{\, J^\top \,\}.$$

The proof is given in the appendix. It would be possible to forbid $(J^\top)^\odot$ as a stable model of $\mathcal{D}^{\mathsf{cd}}$ by adding one of the following rules to $\mathcal{D}^{\mathsf{cd}}$:

$$\begin{aligned} r_{cons}^{logic} &= dis(c_\top) \leftarrow \\ &\qquad dis(X_d) \wedge \mathit{empty_dis}(X_d) \wedge \mathit{not}\, dis(c_\top), \\ r_{cons}^{logic'} &= \leftarrow dis(X_d) \wedge \mathit{empty_dis}(X_d). \end{aligned}$$

$c_\top$ is a new constant symbol for formulas that does not unify with any $\alpha \in \mathit{GHB}_\mathcal{D}$. r_{cons}^{logic} is a cd–rule, and $r_{cons}^{logic'}$ is an equivalent denial rule, which is easier to understand, but is no cd–rule.

5.3 Three–Valued Analysis of $\mathcal{D}^{\mathsf{cd}}$

In the following we will investigate the effects of the propagation rules r_{res}^{logic} and r_{con}^{logic} for disjunctions and conjunctions, respectively, and the closed–world–assumption r_{neg}^{cwa} in three–valued logic. We will see that the well–founded model I_{wfs} of the normal deductive database $\mathcal{D}^{\mathsf{cd}}$ is not appropriate for designing a useful state semantics. Likewise, also the partial stable models of $\mathcal{D}^{\mathsf{cd}}$ are not useful, since the well–founded model I_{wfs} is the least partial stable model of $\mathcal{D}^{\mathsf{cd}}$ in the knowledge–ordering[9].

[9] i.e. I_{wfs} exactly is the set of consequences under the semantics of partial stable models

The Conjunction Rule r_{con}^{logic}. Due to r_{con}^{logic} the partial stable models I exhibit the following behaviour for conjunctions $\beta = L_1 \wedge \ldots \wedge L_k \in C_{HB_\mathcal{D}}$:

$$I(con(\beta)) \geq_{\mathsf{t}} \mathsf{u} \iff (\,\forall i \in \langle\, 1, k \,\rangle : I(con(L_i)) \geq_{\mathsf{t}} \mathsf{u}\,).$$

r_{neg}^{cwa} supports $dis(\neg L_1 \vee \ldots \vee \neg L_k)$, if $I(con(\beta)) = \mathsf{f}$. In this case $I(con(L_i)) = \mathsf{f}$ must hold for some $i \in \langle\, 1, k \,\rangle$, i.e. r_{neg}^{cwa} also supports $dis(\neg L_i)$. In a state–based semantics it should be possible that the closed–world–assumption can support a negative disjunction $dis(\neg A_1 \vee \neg A_2)$ without supporting one of $dis(\neg A_1)$ and $dis(\neg A_2)$. For three–valued models I, such a behaviour is impossible.[10]

Example 5.1. The *stable states semantics* of $\mathcal{D} = \{\, a \leftarrow not\, b,\ b \leftarrow not\, a \,\}$ will be given by the three clausal Herbrand interpretations $S_1 = \{\, a,\ \neg b \,\}$, $S_2 = \{\, \neg a,\ b \,\}$, and $S_3 = \{\, \neg a \vee \neg b \,\}$. S_3 corresponds to the partial Herbrand interpretation $I = S_3^{\mathsf{cd}}$ of $\mathcal{D}^{\mathsf{cd}}$, where $dis(\neg a \vee \neg b)$ is derived since $con(a \wedge b)$ cannot be derived.

The Resolution Rule r_{res}^{logic}. r_{res}^{logic} has the following consequence for disjunctions $\alpha_1, \alpha_2 \in G_{HB_\mathcal{D}}$ and atoms $A \in H_{B_\mathcal{D}}$:

> If $\alpha_1 \vee \alpha_2$ is the binary resolvent of two disjunctions $A \vee \alpha_1$, $\neg A \vee \alpha_2$, such that $I(dis(A \vee \alpha_1)) \geq_{\mathsf{t}} \mathsf{u}$, $I(dis(\neg A \vee \alpha_2)) \geq_{\mathsf{t}} \mathsf{u}$, then $I(dis(\alpha_1 \vee \alpha_2)) \geq_{\mathsf{t}} \mathsf{u}$.

This is undesirable and unituitive since it could be the case that $I(dis(A)) = \mathsf{u}$, and it does not make sense to resolve on undefined atoms.

Example 5.2. The *stable state semantics* of $\mathcal{D} = \{\, a \leftarrow not\, a,\ b \leftarrow c \,\}$ will be given by the clausal Herbrand interpretation $S = \{\, \neg b,\ \neg c \,\}$ corresponding to the partial Herbrand interpretation $I = S_3^{\mathsf{cd}}$ of $\mathcal{D}^{\mathsf{cd}}$. Here $I(dis(a)) = I(dis(a \vee b)) = I(dis(\neg a \vee c)) = \mathsf{u}$, but for the resolvent we get $I(dis(b \vee c)) = \mathsf{f}$.

5.4 Stable State Semantics

Stable states are clausal Herbrand interpretations S of a dd–database $\mathcal{D}$ that are defined by a special *fixpoint condition* $\Gamma_{\mathcal{C}^{\mathsf{t}}}(\Gamma_{\mathcal{C}^{\mathsf{u}}}(S^{\mathsf{cd}})) = S^{\mathsf{cd}}$ in terms of the $\Gamma_\mathcal{C}$–operators [18] for the cd–transformation $\mathcal{C}^{\mathsf{t}} = \mathcal{D}^{\mathsf{cd}}$ and a variant $\mathcal{C}^{\mathsf{u}}$ of $\mathcal{D}^{\mathsf{cd}}$.

Definition 5.1 (Stable States).
Given a cd–database $\mathcal{C}$ and a dd–database $\mathcal{D}$.

1. The (unique) minimal model of the GL–transformation $\mathcal{C}^I$ of $\mathcal{C}$ w.r.t. a total Herbrand interpretation $I \subseteq H_{B_\mathcal{C}}$ of $\mathcal{C}$ is denoted by $\Gamma_\mathcal{C}(I)$.

[10] $dis(\neg A_1 \vee \neg A_2)$ is supported, if $I(con(A_1 \wedge A_2)) = \mathsf{f}$. On the other hand, assuming that $dis(\neg A_1)$ and $dis(\neg A_2)$ are not supported, it must hold $I(con(A_i)) \geq_{\mathsf{t}} \mathsf{u}$, for $i = 1, 2$, which implies $I(con(A_1 \wedge A_2)) \geq_{\mathsf{t}} \mathsf{u}$, and contradicts $I(con(A_1 \wedge A_2)) = \mathsf{f}$.

2. A clausal Herbrand interpretation S of $\mathcal{D}$ is called a *stable state* of $\mathcal{D}$ if for the total Herbrand interpretation $I = S^{\mathsf{cd}}$ of the cd–databases

$$\mathcal{C}^{\mathsf{t}} = \mathcal{D}^{\mathsf{cd}}, \quad \mathcal{C}^{\mathsf{u}} = I \cup \mathcal{D}^{\mathsf{cd}} \setminus \{\, r_{res}^{logic},\, r_{con}^{logic} \,\}$$

it holds $\Gamma_{\mathcal{C}^{\mathsf{t}}}(\Gamma_{\mathcal{C}^{\mathsf{u}}}(I)) = I$. The set of all stable states of $\mathcal{D}$ is denoted by $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D})$.

Stable models I of $\mathcal{C}^{\mathsf{t}} = \mathcal{D}^{\mathsf{cd}}$ are characterized by $\Gamma_{\mathcal{C}^{\mathsf{t}}}(I) = I$. In the following we will represent clausal Herbrand interpretations (and thus stable states) S by their canonical form $can_{\trianglerighteq}(S)$. As a consequence of Theorem 5.1, for a disjunctive deductive database $\mathcal{D}$, the representation of a stable model of $\mathcal{D}$ as a partial Herbrand interpretation $J \subseteq HL_{\mathcal{D}}$ is a stable state of $\mathcal{D}$.

Theorem 5.2 (Stable States).
Given a disjunctive deductive database $\mathcal{D}$, then it hols

$$\mathcal{STABLE}_2(\mathcal{D}) \subseteq \mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}).$$

This shows that the semantics of stable states is at most as strong as the semantics of stable models, i.e. $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D})^{\cap} \subseteq \mathcal{STABLE}_2(\mathcal{D})^{\cap}$. The following Example 5.3 will give some dd–databases ($\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, $\mathcal{D}_5$) where it is strictly weaker. For many prototypical databases which have been investigated frequently in literature we will compare the stable states to the stable models and – for normal databases – to the well–founded model.

Example 5.3 (Stable States).

1. For the positive–disjunctive deductive database $\mathcal{D}_1 = \{\, a \vee b \,\}$ we get

$$\mathcal{STABLE}_2(\mathcal{D}_1) = \mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_1) = \{\ \{\, a, \neg b \,\},\ \{\, \neg a,\ b \,\}\ \}.$$

   The intersection is $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_1)^{\cap} = \mathcal{STABLE}_2(\mathcal{D}_1)^{\cap} = \{\, a \vee b,\ \neg a \vee \neg b \,\}$.
2. For the normal deductive database $\mathcal{D}_2 = \{\, a \leftarrow not\, b,\ b \leftarrow not\, a \,\}$, we get

$$\mathcal{STABLE}_2(\mathcal{D}_2) = \{\ \{\, a, \neg b \,\},\ \{\, \neg a,\ b \,\}\ \},$$
$$\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_2) = \mathcal{STABLE}_2(\mathcal{D}_2) \cup \{\ \{\, \neg a \vee \neg b \,\}\ \}.$$

   $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_2)^{\cap} = \{\, \neg a \vee \neg b \,\}$, is weaker than the state $\mathcal{STABLE}_2(\mathcal{D}_2)^{\cap} = \{\, a \vee b,\ \neg a \vee \neg b \,\}$ derived from the stable models, but stronger than the well–founded model $\emptyset$.
3. For the normal deductive database $\mathcal{D}_3 = \{\, a \leftarrow not\, a \,\}$, which does not possess any (total) stable model, there is unique stable state $S_3 = \emptyset$, which is identical to the well–founded model. A similar structure may for instance *become active* for "$a = win(x)$" within the database $\mathcal{D}_{win}$ of Subsection 2.2, if the state S contains cyclic moves "$move(x, x)$", namely $win(x) \leftarrow move(x, x) \wedge not\ win(x)$.

4. For the normal deductive database $\mathcal{D}_4 = \{ a \leftarrow not\, b,\ b \leftarrow not\, c,\ c \leftarrow not\, a \}$, we get $\mathcal{STABLE}_2(\mathcal{D}_4) = \emptyset$, but there is a unique stable state

$$S_4 = \{ \neg a \vee \neg b,\ \neg b \vee \neg c,\ \neg a \vee \neg c \},$$

   which is weaker than the inconsistent state $\mathcal{STABLE}_2(\mathcal{D}_4)^{\cap} = \mathit{GHB}_{\mathcal{D}_4}$ derived from the stable models, but stronger than the empty well–founded model $\emptyset$.
5. For the dd–database $\mathcal{D}_5 = \mathcal{D}_4 \cup \{ a \vee b \vee c \}$ which does not possess any stable or partial stable models, there is a unique stable state $S_5 = S_4 \cup \{ a \vee b \vee c \}$.
6. For the stratified–disjunctive deductive database $\mathcal{D}_6 = \{ a \leftarrow not\, b \}$, we get a unqiue stable state $S_6 = \{ a, \neg b \}$, which is the unique (total) stable model of $\mathcal{D}_6$, namely the perfect model of $\mathcal{D}_6$.
7. For both of the stratified–disjunctive deductive databases $\mathcal{D}_7' = \{ c \leftarrow a,\ c \leftarrow b,\ a \vee b \}$ and $\mathcal{D}_7'' = \{ c \leftarrow not\, a,\ c \leftarrow not\, b,\ a \vee b \}$ we get (for $\mathcal{D}_7 \in \{ \mathcal{D}_7', \mathcal{D}_7'' \}$)

$$\mathcal{STABLE}_2(\mathcal{D}_7) = \mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_7) = \{ \{ a, \neg b, c \},\ \{ \neg a, b, c \} \}.$$

   The intersection is $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_7)^{\cap} = \mathcal{STABLE}_2(\mathcal{D}_7)^{\cap} = \{ a \vee b,\ \neg a \vee \neg b,\ c \}$.
8. For the dd–database $\mathcal{D}_8 = \{ a, \leftarrow a \}$, which has got no Herbrand models, there exists one stable state $S_8 = \{ \square_d \}$, i.e. $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D}_8)^{\cap} = \mathit{GHB}_{\mathcal{D}_8}$.

Especially for the dd–databases $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$, $\mathcal{D}_5$, which are inconsistent under stable model semantics, stable states provide a plausible, consistent semantics.

For positive–disjunctive deductive databases the intersection $\mathcal{STABLE}_{\mathsf{cd}}(\mathcal{D})^{\cap}$ of all stable states corresponds to the union of the *minimal model state* $\mathit{MS}_{\mathcal{D}}$ and the *extended generalized closed–world-assumption* $\mathit{EGCWA}_{\mathcal{D}}$, as defined by [11], together with all general disjunctions that can be derived from $\mathit{MS}_{\mathcal{D}}$ – which is a disjunctive Herbrand state – and $\mathit{EGCWA}_{\mathcal{D}}$ – which is represented as a negative Herbrand state – by repeated applications of binary resolution.

6 Conclusions

Extended–disjunctive deductive databases allow literals to occur in all places where atoms are possible in disjunctive deductive databases cf. [14]. Marek and Truszczyński have introduced *clausal logic programming*, cf. [12], as a further extension: clausal logic programs allow for general disjunctions – rather than atoms – in rule bodies, next to general disjunctions in rule heads. But clausal logic programs do not allow for variable symbols that stand for general disjunctions in their meta–language $\mathcal{L}^{\mathsf{cd}}$, and the ideas of transforming disjunctive logic programs to clausal logic programs or relating the semantics of disjunctive logic programs and corresponding clausal logic programs have not been investigated.

In a forthcoming paper we will give a *characterization of the stable states* of a disjunctive deductive database in terms of the stable models of an *annotated* clausal deductive database $\mathcal{D}^{\mathsf{cd,tu}}$ by modifying the concept of the tu–*transformation* known from [21]. We will investigate the new semantics w.r.t.

further desirable properties, such as *modularity* or *independence under partial evaluation (GPPE)*, and compare it with some more semantics.

In the future we will consider the complexity issue of stable state semantics, and we will work on an *efficient implementation*. So far we have used the system Smodels developed by Niemelä [15] for computing well–founded and stable model semantics for normal deductive databases. Smodels uses a *grounding procedure*, which does not work for databases with function symbols like the cd–transformation $\mathcal{D}^{\mathsf{cd}}$. Thus, we have written a new grounding procedure for computing the ground instance $gnd\ (\mathcal{D}^{\mathsf{cd}})$, if the Herbrand base $H_{B\mathcal{D}}$ of $\mathcal{D}$ is finite. For a more efficient implementation, the goal is to *back transform* the declarative approach that is working with cd–databases into a procedure that is directly working with dd–databases. This would allow us to use systems that can handle disjunctions directly, e.g. the dlv–system of Eiter, Leone et al. [4], or our system DisLog [23] for reasoning in disjunctive deductive databases.

In another paper we will also show how reasoning in *hypothetical* [3] and in *probabilistic deductive databases* [25] can be formulated easily in terms of a suitable generic consequence operator $\mathcal{T}_{\mathcal{D}}^{\odot,\geq}$, which gives immediate results about the continuity of the underlying consequence operators without requiring extra proofs.

References

1. *S. Brass, J. Dix:* A Disjunctive Semantics Based on Unfolding and Bottom–Up Evaluation, GI–Jahrestagung / IFIP World Computer Congress 1994: Fachgespräch "Disjunktive logische Programmierung und disjunktive Datenbanken", Springer, 1994, pp. 83–91.
2. *S. Ceri, G. Gottlob, L. Tanca:* Logic Programming and Databases, Springer, 1990.
3. *P.M. Dung*: Declarative Semantics of Hypothetical Logic Programming with Negation as Failure, Proc. Workshop on Extensions of Logic Programming 1992 (ELP'92), 1992, pp. 45–58.
4. *T. Eiter, N. Leone, C. Mateis, G. Pfeifer, F. Scarcello:* A Deductive System for Non–Monotonic Reasoning, Proc. Fourth Intl. Conf. on Logic Programming and Non–Monotonic Reasoning 1997 (LPNMR'97), Springer LNAI 1265, 1997, pp. 363–374.
5. *T. Eiter, N. Leone, D. Sacca:* On the Partial Semantics for Disjunctive Deductive Databases, Annals of Mathematics and Artificial Intelligence, vol. 17(1/2), 1997, pp. 59–96.
6. *J.A. Fernández, J. Lobo, J. Minker, V.S. Subrahmanian:* Disjunctive LP + Integrity Constrains = Stable Model Semantics, Annals of Math. and AI, vol. 8 (3–4), 1993, pp. 449–474.
7. *J.A. Fernández and J. Minker:* Bottom–up computation of perfect models for disjunctive theories, Journal of Logic Programming, vol. 25 (1), 1995, pp. 33–51.
8. *M. Gelfond, V. Lifschitz:* The Stable Model Semantics for Logic Programming, Proc. Fifth Intl. Conference and Symposium on Logic Programming 1988 (ICSLP'88), MIT Press, 1988, pp. 1070–1080.
9. *G. Köstler, W. Kießling, H. Thöne, U. Güntzer:* The differential fixpoint operator with subsumption, Proc. Intl. Conference on Deductive and Object–Oriented Databases 1993 (DOOD'93), Springer LNCS 760, 1993, pp. 35–48.

10. *J.W. Lloyd:* Foundations of Logic Programming, second edition, Springer, 1987.
11. *J. Lobo, J. Minker, A. Rajasekar:* Foundations of Disjunctive Logic Programming, MIT Press, 1992.
12. *V.W. Marek, M. Truszczyński:* Nonmonotonic Logic – Context–Dependent Reasoning, Springer, 1993.
13. *J. Minker, A. Rajasekar*: A Fixpoint Semantics for Disjunctive Logic Programs, Journal of Logic Programming, vol. 9(1), 1990, pp. 45–74.
14. *J. Minker, C. Ruiz:* On Extended Disjunctive Logic Programs, Proc. Intl. Symposium on Methodologies for Intelligent Systems 1993 (ISMIS'93), Springer LNAI 689, 1993, pp. 1–18.
15. *I. Niemelä, P. Simons:* Efficient Implementation of the Well–founded and Stable Model Semantics, Technical Report 7/96, Univ. Koblenz – Landau, 1996.
16. *T.C. Przymusinski:* Stable Semantics for Disjunctive Programs, New Generation Computing, vol. 9, 1991, pp. 401–424.
17. *R. Ramakrishnan:* Database Management Systems, McGraw–Hill, 1998.
18. *R. Reiter:* A Logic for Default Reasoning, Artificial Intelligence, vol. 13, 1980, pp. 81–132.
19. *K. Sagonas, T. Swift, D. Warren:* XSB as an Efficient Deductive Database Engine, Proc. ACM SIGMOD Intl. Conf. on the Management of Data 1994 (SIGMOD'94), pp. 442–453.
20. *D. Seipel, J. Minker, C. Ruiz*: Model Generation and State Generation for Disjunctive Logic Programs, Journal of Logic Programming, vol. 32(1), 1997, pp. 48–69.
21. *D. Seipel, J. Minker, C. Ruiz*: A Characterization of Partial Stable Models for Disjunctive Deductive Databases, Proc. Intl. Logic Programming Symposium 1997 (ILPS'97), MIT Press, 1997, pp. 245–259.
22. *D. Seipel*: Partial Evidential Stable Models For Disjunctive Databases, Proc. Workshop on Logic Programming and Knowledge Representation (LPKR'97) at the International Symposium on Logic Programming 1997 (ILPS'97), Springer LNAI 1471, 1998, pp. 66–84.
23. *D. Seipel:* DisLog – A Disjunctive Deductive Database Prototype, Proc. Twelfth Workshop on Logic Programming (WLP'97), 1997, pp. 136–143. DisLog is available at "http://www-info1.informatik.uni-wuerzburg.de/databases/DisLog".
24. *A. Van Gelder, K.A. Ross, J.S. Schlipf:*, Unfounded Sets and Well–Founded Semantics for General Logic Programs, Proc. Seventh ACM Symposium on Principles of Database Systems, 1988 (PODS'88), pp. 221–230.
25. *C. Zaniolo, S. Ceri, C. Faloutsos, R.T. Snodgrass, V.S. Subrahmanian, R. Zicari:* Advanced Database Systems, Morgan Kaufmann, 1997.
26. *U. Zukowski, S. Brass, B. Freitag:* Improving the Alternating Fixpoint: The Transformation Approach, Proc. 4th Intl. Conf. on Logic Programming an Non–Monotonic Reasoning 1997 (LPNMR'97), Springer LNAI 1265, 1997, pp. 40–59.

Appendix

Theorem 4.1 (Properties of the Generic $\mathcal{T}_{\mathcal{D}}$–Operator)
The operator $\mathcal{T}_{\mathcal{D}}^{\odot,\geq} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ is *monotonic* and *continuous* on the complete lattice $\langle 2^{\mathcal{X}}, \subseteq \rangle$.

Proof. (i) Firstly, we prove *monotonicity*:

$$X_1 \subseteq X_2 \;\Rightarrow\; exp_{\geq}(X_1) \subseteq exp_{\geq}(X_2) \;\Rightarrow\; \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(X_1) \subseteq \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(X_2).$$

(ii) Secondly, for proving *continuity*, consider a directed set $\mathcal{Y} \subseteq 2^{\mathcal{X}}$ of subsets of $\mathcal{X}$. Let $X = \cup_{Y \in \mathcal{Y}} Y$. Then $\cup_{Y \in \mathcal{Y}} exp_{\geq}(Y) = exp_{\geq}(X)$. Obviously, it holds $\cup_{Y \in \mathcal{Y}} \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(Y) \subseteq \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(X)$. Conversely, if $\alpha \in \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(X)$, then there must exist a rule $\alpha \leftarrow \beta_1 \wedge \ldots \wedge \beta_m \in gnd\,(\mathcal{D}^{\odot})$, such that $\{\beta_1, \ldots, \beta_m\} \subseteq exp_{\geq}(X)$. Thus, there must exist sets $Y_{j_1}, \ldots, Y_{j_m} \in \mathcal{Y}$, such that $\beta_i \in exp_{\geq}(Y_{j_i})$. I.e. there exist elements $\beta_i' \in Y_{j_i}$, such that $\beta_i' \geq \beta_i$. Since $\mathcal{Y}$ is directed, there must exist a set $Y_{j_0} \in \mathcal{Y}$, such that $\{\beta_1', \ldots, \beta_m'\} \subseteq Y_{j_0}$. Thus, $\{\beta_1, \ldots, \beta_m\} \subseteq exp_{\geq}(Y_{j_0})$, and $\alpha \in \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(Y_{j_0}) \subseteq \cup_{Y \in \mathcal{Y}} \mathcal{T}_{\mathcal{D}}^{\odot,\geq}(Y)$.

□

Theorem 5.1 (Characterization of Stable Models)
Given a disjunctive deductive database $\mathcal{D}$, then it holds

$$\mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D}^{\mathsf{cd}})^{\otimes} = \mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D}) \cup \mathcal{J}, \quad \text{where } \mathcal{J} = \emptyset \text{ or } \mathcal{J} = \{ J^{\top} \}.$$

Proof. (i) Consider a total model I of $\mathcal{D}^{\mathsf{cd}}$, such that $I(dis(\Box_d)) = \mathsf{f}$, and consider the corresponding Herbrand interpretation $I^{\otimes}$ of $\mathcal{D}$. It is not necessary to deal with atoms $dis(\alpha)$ for non–singleton negative disjunctions $\alpha \in N_{HB_{\mathcal{D}}}$ during the fixpoint evaluation of $\mathcal{D}^{\mathsf{cd}}$: amoung the rules $(r_{neg}^{cwa})^I$ the ones in $\{ dis(\neg A) \leftarrow negative(\neg A) \wedge complement(\neg A, A) \mid A \in H_{B_{\mathcal{D}}}, con(A) \notin I \}$ for minimal, i.e. atomary, disjunctions "$\neg A$" are *subsuming* the others. The resolution rule r_{res}^{logic} will consume the atoms $dis(\neg A)$ and produce atoms $dis(\alpha)$ over positive disjunctions $\alpha \in D_{HB_{\mathcal{D}}}$. The evaluation is equivalent to evaluating the disjunctive deductive database $\mathcal{D}^{I^{\otimes}}$ together with the set $\mathcal{D}' = \{ \leftarrow A \mid A \in H_{B_{\mathcal{D}}}, con(A) \notin I \}$ of denial rules, using the disjunctive consequence operator $\mathcal{T}_{\mathcal{D}''}^{s}$ for $\mathcal{D}'' = \mathcal{D}^{I^{\otimes}} \cup \mathcal{D}'$.

(ii) $I \neq (J^{\top})^{\odot}$ is a stable model of $\mathcal{D}^{\mathsf{cd}}$ iff it holds $can_{\rhd}(\mathcal{T}_{\mathcal{D}''}^{s} \uparrow \omega) = \{ A \in H_{B_{\mathcal{D}}} | I^{\otimes}(A) = \mathsf{t} \}$, i.e. iff $I^{\otimes}$ is a minimal model $\mathcal{D}^{I^{\otimes}}$, i.e. iff $I^{\otimes} \in \mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D})$.

(iii) As a consequence of (ii) we get $\mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D}^{\mathsf{cd}})^{\otimes} \subseteq \mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D}) \cup \{ J^{\top} \}$. Every stable model $J \in \mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D})$ can be perceived as $J = I^{\otimes}$ for the total Herbrand interpretation $I = J^{\odot}$ of $\mathcal{D}^{\mathsf{cd}}$. Thus, according to (ii) we get $J \in \mathcal{ST}\mathcal{AB}\mathcal{L}\mathcal{E}_2(\mathcal{D}^{\mathsf{cd}})^{\otimes}$.

□