I N F S Y S

RESEARCH

R E P O R T

INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG WISSENSBASIERTE SYSTEME

FIXPOINT CHARACTERIZATIONS FOR MANY-VALUED DISJUNCTIVE LOGIC PROGRAMS WITH PROBABILISTIC SEMANTICS

Thomas Lukasiewicz

INFSYS RESEARCH REPORT 1843-01-06

JUNE 2001

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT

INFSYS RESEARCH REPORT 1843-01-06, JUNE 2001

FIXPOINT CHARACTERIZATIONS FOR MANY-VALUED DISJUNCTIVE LOGIC PROGRAMS WITH PROBABILISTIC SEMANTICS

(PRELIMINARY REPORT, JUNE 2001)

Thomas Lukasiewicz[1]

Abstract. In this paper, we continue to explore many-valued disjunctive logic programs with probabilistic semantics. In particular, we newly introduce the least model state semantics for such programs. We show that many-valued disjunctive logic programs under the semantics of minimal models, perfect models, stable models, and least model states can be unfolded to equivalent classical disjunctive logic programs under the respective semantics. Thus, existing technology for classical disjunctive logic programming can be used to implement many-valued disjunctive logic programming. Using these results on unfolding many-valuedness, we then give many-valued fixpoint characterizations for the set of all minimal models and the least model state. We also describe an iterative fixpoint characterization for the perfect model semantics under finite local stratification.

[1]Institut und Ludwig Wittgenstein Labor für Informationssysteme, TU Wien, Favoritenstraße 9-11, A-1040 Vienna, Austria. E-mail: lukasiewicz@kr.tuwien.ac.at.

Acknowledgements: This work has been supported by a DFG grant and the Austrian Science Fund under project N Z29-INF.

1 Introduction

In a previous paper [13], we introduced many-valued disjunctive logic programs with probabilistic semantics. In particular, we defined minimal, perfect, and stable models for such programs, and showed that they have the same properties like their classical counterparts. For example, perfect and stable models are always minimal models. Under local stratification, the perfect model semantics coincides with the stable model semantics. Moreover, we also showed that some special cases of propositional many-valued disjunctive logic programming under minimal, perfect, and stable model semantics have the same complexity as their classical counterparts.

In this paper, we continue this line of research on many-valued disjunctive logic programming with probabilistic semantics. The central topic of the present paper is to elaborate algorithms for many-valued disjunctive logic programming. One way of obtaining such algorithms is to translate many-valued disjunctive logic programs into classical formalisms, and to work with existing algorithms for the classical formalisms. Another way is to simply develop completely new algorithms.

In this paper, we follow both directions. We first show that many-valued disjunctive logic programs under minimal models, perfect models, stable models, and least model states can be unfolded to equivalent classical disjunctive logic programs under the respective semantics. Thus, existing technology for classical disjunctive logic programming can be used to implement many-valued disjunctive logic programming.

Using these results on unfolding many-valuedness, we then develop new many-valued fixpoint characterizations for the semantics of minimal models, least model states, and perfect models under finite local stratification.

It is important to point out that our many-valued disjunctive logic programs have a probabilistic semantics in probabilities over possible worlds. Furthermore, the truth values of all clauses are truth-functionally defined on the truth values of atoms. This gives our many-valued disjunctive logic programs both nice computational properties (compared to purely probabilistic approaches) and a nice probabilistic semantics. The latter is expressed in the fact that our many-valued disjunctive logic programming under the minimal model and the least model state semantics is an approximation of purely probabilistic disjunctive logic programming.

We showed in [14, 15] that many-valued *definite* logic programming with this probabilistic semantics has a model and fixpoint characterization and a proof theory similar to classical definite logic programming. Moreover, special cases of many-valued logic programming with this semantics were shown to have the same computational complexity as their classical counterparts. Interestingly, our approach in [14, 15] is closely related to van Emden's quantitative deduction [30], which interprets the implication connective as conditional probability, while our work uses the material implication.

The main contributions of this paper can be summarized as follows.

- We introduce the least model state semantics for positive many-valued disjunctive logic programs with probabilistic semantics.

- We show that many-valued disjunctive logic programs under minimal model, perfect model, stable model, and least model state semantics can be unfolded to equivalent classical disjunctive logic programs under the respective semantics.

- We provide fixpoint characterizations for the set of all minimal models and the least model state of positive many-valued disjunctive logic programs.
- We describe an iterative fixpoint characterization for the perfect model of many-valued disjunctive logic programs that have a finite local stratification.

The rest of this paper is organized as follows. In Section 2, we give some technical preliminaries. Section 3 concentrates on the least model state semantics. In Section 4, we discuss the unfolding of many-valuedness. Section 5 focuses on the fixpoint characterizations. In Section 6, we summarize the main results and give an outlook on future research.

Note that proofs of all results are given in the appendix.

2 Preliminaries

In this section, we recall some necessary definitions and results from [13].

2.1 Probabilistic Background

We now briefly summarize how (a quantifier-free fragment of) classical first-order logics can be given a probabilistic semantics in which probabilities are defined over a set of possible worlds. We essentially follow the approaches in [3, 8, 9].

Let Φ be a first-order vocabulary that contains a set of function symbols and a set of predicate symbols (as usual, *constant symbols* are function symbols of arity zero). Let $\mathcal{X}$ be a set of variables. We define *terms* by induction as follows. A term is a variable from $\mathcal{X}$ or an expression of the form $f(t_1, \ldots, t_k)$, where f is a function symbol of arity $k \geq 0$ from Φ and $t_1, \ldots, t_k$ are terms. We define *classical formulas* by induction as follows. If p is a predicate symbol of arity $k \geq 0$ from Φ and $t_1, \ldots, t_k$ are terms, then $p(t_1, \ldots, t_k)$ is a classical formula (called *atom*). If F and G are classical formulas, then also $\neg F$ and $(F \wedge G)$. Literals, positive literals, and negative literals are defined as usual. We define *probabilistic formulas* inductively as follows. If F is a classical formula and c is a real number from $[0, 1]$, then $\mathsf{prob}(F) \geq c$ is a probabilistic formula (called *atomic probabilistic formula*). If F and G are probabilistic formulas, then also $\neg F$ and $(F \wedge G)$. We use $(F \vee G)$ and $(F \leftarrow G)$ to abbreviate $\neg(\neg F \wedge \neg G)$ and $\neg(\neg F \wedge G)$, respectively, and adopt the usual conventions to eliminate parentheses. Terms and formulas are *ground* iff they do not contain any variables. Substitutions, ground substitutions, and ground instances of formulas are defined as usual.

A *classical interpretation* I is a subset of the Herbrand base HB_Φ over Φ. A *variable assignment* σ assigns to each $x \in \mathcal{X}$ an element from the Herbrand universe HU_Φ over Φ. It is by induction extended to terms by $\sigma(f(t_1, \ldots, t_k)) = f(\sigma(t_1), \ldots, \sigma(t_k))$ for all terms $f(t_1, \ldots, t_k)$. The *truth* of classical formulas F in I under σ, denoted $I \models_\sigma F$, is inductively defined as follows (we write $I \models F$ when F is ground):

- $I \models_\sigma p(t_1, \ldots, t_k)$ iff $p(\sigma(t_1), \ldots, \sigma(t_k)) \in I$.
- $I \models_\sigma \neg F$ iff not $I \models_\sigma F$, and $I \models_\sigma (F \wedge G)$ iff $I \models_\sigma F$ and $I \models_\sigma G$.

A *probabilistic interpretation* (or *p-interpretation*) $\boldsymbol{p} = (\mathcal{I}, \mu)$ consists of a set $\mathcal{I}$ of classical interpretations (called *possible worlds*) and a discrete probability function μ on $\mathcal{I}$ (that is, a mapping μ from $\mathcal{I}$ to the real interval $[0,1]$ such that all $\mu(I)$ with $I \in \mathcal{I}$ sum up to 1 and that the number of all $I \in \mathcal{I}$ with $\mu(I) > 0$ is countable). The *truth value* of a formula F in a p-interpretation $\boldsymbol{p}$ under a variable assignment σ, denoted $\boldsymbol{p}_\sigma(F)$, is defined as the sum of all $\mu(I)$ such that $I \in \mathcal{I}$ and $I \models_\sigma F$ (we write $\boldsymbol{p}(F)$ when F is ground). The *truth* of probabilistic formulas F in $\boldsymbol{p}$ under σ, denoted $\boldsymbol{p} \models_\sigma F$, is defined as follows (we write $\boldsymbol{p} \models F$ when F is ground):

- $\boldsymbol{p} \models_\sigma \mathsf{prob}(F) \geq c$ iff $\boldsymbol{p}_\sigma(F) \geq c$.
- $\boldsymbol{p} \models_\sigma \neg F$ iff not $\boldsymbol{p} \models_\sigma F$, and $\boldsymbol{p} \models_\sigma (F \wedge G)$ iff $\boldsymbol{p} \models_\sigma F$ and $\boldsymbol{p} \models_\sigma G$.

The probabilistic formula F *is true* in $\boldsymbol{p}$, or $\boldsymbol{p}$ is a *model* of F, denoted $\boldsymbol{p} \models F$, iff F is true in $\boldsymbol{p}$ under all variable assignments σ. The p-interpretation $\boldsymbol{p}$ is a *model* of a set of probabilistic formulas $\mathcal{F}$, denoted $\boldsymbol{p} \models \mathcal{F}$, iff $\boldsymbol{p}$ is a model of all $F \in \mathcal{F}$. A set of p-interpretations $\boldsymbol{P}$ is a *model* of F (resp., $\mathcal{F}$), denoted $\boldsymbol{P} \models F$ (resp., $\boldsymbol{P} \models \mathcal{F}$), iff every member of $\boldsymbol{P}$ is a model of F (resp., $\mathcal{F}$).

2.2 Positively Correlated Probabilistic Interpretations

We restrict our attention to the following kind of p-interpretations (that is, we assume another axiom besides the axioms of probability). A *positively correlated probabilistic interpretation* (or *pcp-interpretation*) is a p-interpretation $\boldsymbol{p}$ such that

$$\boldsymbol{p}(A \wedge B) = \min(\boldsymbol{p}(A), \boldsymbol{p}(B)) \text{ for all } A, B \in HB_\Phi\,. \tag{1}$$

Note that the condition $\boldsymbol{p}(A \wedge B) = \min(\boldsymbol{p}(A), \boldsymbol{p}(B))$ is just assumed for ground atoms A and B. It brings probabilistic logics over possible worlds closer to truth-functional logics. We do not assume that (1) always holds in the part of the real world that we want to model. The axiom (1) is simply a *technical assumption* that carries us to a form of many-valued logic programming that *approximates* probabilistic logic programming. It makes a global probabilistic semantics over possible worlds match with the truth-functionality behind logic programming techniques. Differently from many other axioms, the axiom (1) is compatible with logical implication. Note that pcp-interpretations are uniquely determined by the truth values they give to all ground atoms [13], and thus they can be identified with mappings from HB_Φ to $[0,1]$.

A probabilistic formula F is a *pc-consequence* of a set of probabilistic formulas $\mathcal{F}$, denoted $\mathcal{F} \models^{pc} F$, iff each pcp-interpretation that is a model of $\mathcal{F}$ is also a model of F.

2.3 Many-Valued Disjunctive Logic Programs

We are now ready to define many-valued disjunctive logic programs. We start by defining many-valued disjunctive logic program clauses, which are special atomic probabilistic formulas that are interpreted under pcp-interpretations. A *many-valued disjunctive logic program clause* (or *mvd-clause*) is a probabilistic formula of the kind

$$\mathsf{prob}(A_1 \vee \cdots \vee A_l \leftarrow B_1 \wedge \cdots \wedge B_m \wedge \neg C_1 \wedge \cdots \wedge \neg C_n) \geq c\,,$$

where $A_1, \ldots, A_l, B_1, \ldots, B_m, C_1, \ldots, C_n$ are atoms, $l, m, n \geq 0$, and $c \in [0,1]$ is rational. It is abbreviated by $(A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, \mathit{not}\, C_1, \ldots, \mathit{not}\, C_n)[c, 1]$. We call $A_1 \vee \cdots \vee A_l$ its *head*, $B_1, \ldots, B_m, \mathit{not}\, C_1, \ldots, \mathit{not}\, C_n$ its *body*, and c its *truth value*. It is *positive* (resp., *definite*) iff $n = 0$ (resp., $l = 1$ and $n = 0$). It is called an *integrity clause* iff $l = 0$, a *fact* iff $l > 0$ and $m + n = 0$, and a *rule* iff $l > 0$ and $m + n > 0$. A *many-valued disjunctive logic program* (or *mvd-program*) P is a finite set of mvd-clauses. A *positive* (resp., *definite*) mvd-program is a finite set of positive (resp., definite) mvd-clauses. Given an mvd-program P, we identify Φ with the vocabulary $\Phi(P)$ of all function and predicate symbols in P. Denote by HB_P the Herbrand base over $\Phi(P)$, and by $ground(P)$ the set of all ground instances of members of P w.r.t. $\Phi(P)$. The *set of truth values* of P, denoted $TV(P)$, is the least set of rational numbers $\{\frac{0}{n-1}, \frac{1}{n-1}, \ldots, \frac{n-1}{n-1}\}$ that contains all the rational numbers in P, where $n \geq 2$ is a natural number. Denote by $\boldsymbol{I}_P$ the set of all pcp-interpretations over HB_P into $TV(P)$.

The following result shows that the truth of a ground mvd-clause under a pcp-interpretation is a function of the truth values of the contained ground atoms.

Theorem 2.1 *Let $C = (A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, \mathit{not}\, C_1, \ldots, \mathit{not}\, C_n)[c, 1]$ be a ground mvd-clause, and let $\mathbf{p}$ be a pcp-interpretation. Then, $\mathbf{p}$ is a model of C iff*

$$\max(\max_{1 \leq i \leq l} \boldsymbol{p}(A_i), \max_{1 \leq i \leq n} \boldsymbol{p}(C_i)) \;\geq\; c - 1 + \min_{1 \leq i \leq m} \boldsymbol{p}(B_i)\,.$$

We finally define queries and their correct and tight answers. A *many-valued query* (or simply *query*) is an expression $\exists(F)[t, 1]$, where F is a ground classical formula and t is a variable or a rational number from $[0, 1]$. Given the queries $\exists(F)[c, 1]$ and $\exists(F)[x, 1]$ to an mvd-program P, where $c \in [0, 1]$ and $x \in \mathcal{X}$, we define their desired semantics in terms of correct and tight answers with respect to a set $\boldsymbol{M}(P)$ of models of P as follows. The *correct answer* for $\exists(F)[c, 1]$ to P *under* $\boldsymbol{M}(P)$ is Yes if $c \leq \inf\{\boldsymbol{p}(F) \mid \boldsymbol{p} \in \boldsymbol{M}(P)\}$ and No otherwise. The *tight answer* for $\exists(F)[x, 1]$ to P *under* $\boldsymbol{M}(P)$ is the substitution $\theta = \{x/d\}$, where $d = \inf\{\boldsymbol{p}(F) \mid \boldsymbol{p} \in \boldsymbol{M}(P)\}$.

In the rest of this section, we recall minimal, perfect, and stable models from [13] as some ways of describing the meaning of an mvd-program.

2.4 Minimal Models

For pcp-interpretations $\boldsymbol{p}$ and $\boldsymbol{q}$, we say $\boldsymbol{p}$ is a *subset* of $\boldsymbol{q}$, denoted $\boldsymbol{p} \subseteq \boldsymbol{q}$, iff $\boldsymbol{p}(A) \leq \boldsymbol{q}(A)$ for all $A \in HB_\Phi$. We use $\boldsymbol{p} \subset \boldsymbol{q}$ as an abbreviation for $\boldsymbol{p} \subseteq \boldsymbol{q}$ and $\boldsymbol{p} \neq \boldsymbol{q}$. A model $\boldsymbol{p}$ of an mvd-program P is a *minimal model* of P iff no model of P is a proper subset of $\boldsymbol{p}$. Denote by $\boldsymbol{MM}(P)$ the set of all minimal models of P.

2.5 Perfect Models

We first define the two relations $\prec$ and $\preceq$ on ground atoms. For an mvd-program P, the priority relation $\prec$ and the auxiliary relation $\preceq$ are the least binary relations on HB_P with the following properties. If $ground(P)$ contains an mvd-clause with the atom A in the head and the negative literal $not\, C$ in the body, then $A \prec C$. If $ground(P)$ contains an mvd-clause with the atom A in the head and the positive literal B in the body, then $A \preceq B$. If $ground(P)$ contains an mvd-clause with the atoms A and A' in the head, then $A \preceq A'$. If $A \prec B$, then $A \preceq B$. If $A \preceq B$ and $B \preceq C$, then $A \preceq C$. If $A \preceq B$ and $B \prec C$, then $A \prec C$. If $A \prec B$ and $B \preceq C$, then $A \prec C$. We say that the ground atom B *has higher priority than* the ground atom A iff $A \prec B$.

We next define the preference relation $\ll$ on pcp-interpretations as follows. For pcp-interpretations $\boldsymbol{p}$ and $\boldsymbol{q}$, we say $\boldsymbol{p}$ is *preferable* to $\boldsymbol{q}$, denoted $\boldsymbol{p} \ll \boldsymbol{q}$, iff $\boldsymbol{p} \neq \boldsymbol{q}$ and for each $A \in HB_P$ with $\boldsymbol{p}(A) > \boldsymbol{q}(A)$ there is some $B \in HB_P$ with $\boldsymbol{q}(B) > \boldsymbol{p}(B)$ and $A \prec B$. We write $\boldsymbol{p} \lll \boldsymbol{q}$ iff $\boldsymbol{p} \ll \boldsymbol{q}$ or $\boldsymbol{p} = \boldsymbol{q}$.

A model $\boldsymbol{q}$ of an mvd-program P is a *perfect model* of P iff no model of P is preferable to $\boldsymbol{q}$. We use $\boldsymbol{PM}(P)$ to denote the set of all perfect models of P.

Not every mvd-program has a perfect model. We next define locally stratified mvd-programs without integrity clauses, which always have a perfect model.

An mvd-program P without integrity clauses is *locally stratified* iff HB_P can be partitioned into sets $H_1, H_2, \ldots$ (called *strata*) such that for each mvd-clause

$$(A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, not\, C_1, \ldots, not\, C_n)[c, 1] \in ground(P)\,,$$

there exists an $i \geq 1$ such that all $A_1, \ldots, A_l$ belong to H_i, all $B_1, \ldots, B_m$ belong to $H_1 \cup \cdots \cup H_i$, and all $C_1, \ldots, C_n$ belong to $H_1 \cup \cdots \cup H_{i-1}$. For such a partition $H_1, H_2, \ldots$ of HB_P (called a *local stratification* of P) and every $i \geq 1$, we use P_i to denote the set of all mvd-clauses from $ground(P)$ whose heads belong to H_i.

2.6 Stable Models

An *extended many-valued disjunctive logic program clause* (or *emvd-clause*) is an expression $(A_1 \vee \cdots \vee A_l\,;\, d \leftarrow B_1, \ldots, B_m, not\, C_1, \ldots, not\, C_n)[c, 1]$, where $A_1, \ldots, A_l, B_1, \ldots, B_m, C_1, \ldots, C_n$ are atoms, $l, m, n \geq 0$, $c \in [0, 1]$ is rational, and $d \in [0, 1]$. It *is true* in a pcp-interpretation $\boldsymbol{p}$ under a variable assignment σ iff

$$\max(\max_{1 \leq i \leq l} \boldsymbol{p}_\sigma(A_i), \max_{1 \leq i \leq n} \boldsymbol{p}_\sigma(C_i), d) \;\geq\; c - 1 + \min_{1 \leq i \leq m} \boldsymbol{p}_\sigma(B_i)\,.$$

Thus, differently from mvd-clauses, emvd-clauses may also contain truth values in their heads.

For an mvd-program P and a pcp-interpretation $\boldsymbol{q}$, the expression $P/\boldsymbol{q}$ denotes the set of emvd-clauses that is obtained from $ground(P)$ by replacing every mvd-clause $(A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, not\, C_1, \ldots, not\, C_n)[c, 1]$ by the emvd-clause

$$(A_1 \vee \cdots \vee A_l\,;\, \max_{1 \leq i \leq n} \boldsymbol{q}(C_i) \leftarrow B_1, \ldots, B_m)[c, 1]\,.$$

A pcp-interpretation $\boldsymbol{q}$ is a *stable model* of an mvd-program P iff $\boldsymbol{q}$ is a minimal model of $P/\boldsymbol{q}$. We use $\boldsymbol{SM}(P)$ to denote the set of all stable models of P.

2.7 Example

We now give an illustrative example. The following mvd-program P is taken from [13] (r, s, a, b, and c are constant symbols, and R, X, Y, and Z are variables):

$$P = \{(\mathit{closed}(r) \vee \mathit{closed}(s) \leftarrow)[.5,1], (\mathit{road}(r,a,b) \leftarrow)[.8,1], (\mathit{road}(s,b,c) \leftarrow)[.7,1],$$
$$(\mathit{reach}(X,Y) \leftarrow \mathit{road}(R,X,Y), \mathit{not}\ \mathit{closed}(R))[.9,1],$$
$$(\mathit{reach}(X,Z) \leftarrow \mathit{reach}(X,Y), \mathit{reach}(Y,Z))[.9,1]\}\,.$$

The set of truth values of P is given by $TV(P) = \{0, 0.1, \ldots, 1\}$.

A query to P may be given by $\exists(\mathit{reach}(a,c))[U,1]$, where U is a variable. To determine its tight answer, we must specify a set of models of P. Some models $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_4$ of P are shown in Table 1 (we assume $\boldsymbol{p}_i(A) = 0$ for all unmentioned $A \in HB_P$). More precisely, the models $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_4$ are some minimal models of P, whereas the models $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are the only perfect and stable models of the locally stratified mvd-program P. The tight answer for $\exists(\mathit{reach}(a,c))[U,1]$ to P under $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \boldsymbol{p}_4\}$ and $\{\boldsymbol{p}_1, \boldsymbol{p}_2\}$ is given by $\{U/0\}$ and $\{U/0.5\}$, respectively.

Table 1: Some models of the mvd-program P

	$\mathit{closed}(r)$	$\mathit{closed}(s)$	$\mathit{road}(r,a,b)$	$\mathit{road}(s,b,c)$	$\mathit{reach}(a,b)$	$\mathit{reach}(b,c)$	$\mathit{reach}(a,c)$
$\boldsymbol{p}_1$	0.5	0	0.8	0.7	0.7	0.6	0.5
$\boldsymbol{p}_2$	0	0.5	0.8	0.7	0.7	0.6	0.5
$\boldsymbol{p}_3$	0	0.6	0.8	0.7	0.7	0	0
$\boldsymbol{p}_4$	0	0.7	0.8	0.7	0	0	0

3 Least Model States

We now define least model states for positive mvd-programs, which are a generalization of their classical counterparts by Minker and Rajasekar [20, 12].

In the sequel, we use A^α to abbreviate atomic probabilistic formulas of the form $\mathsf{prob}(A) \geq \alpha$. Given an mvd-program P, the *disjunctive Herbrand base* for P, denoted DHB_P, is the set of all disjunctions of atomic probabilistic formulas $A_1^{\alpha_1} \vee \cdots \vee A_k^{\alpha_k}$ with pairwise distinct ground atoms $A_1, \ldots, A_k \in HB_P$, $\alpha_1, \ldots, \alpha_k \in TV(P) \setminus \{0\}$, and $k \geq 1$. A *disjunctive Herbrand state* (or *state*) S is a subset of DHB_P. A state S is a *model state* of a positive mvd-program P iff

$$\{D \in DHB_P \mid S \cup P \models^{pc} D\} \subseteq S\,.$$

A model $\boldsymbol{p}$ of a state S is a *minimal model* of S iff no model of S is a proper subset of $\boldsymbol{p}$. We use $\boldsymbol{MM}(S)$ to denote the set of all minimal models of S. The *canonical form* (resp., *expansion*) of a state S, denoted $\mathit{can}(S)$ (resp., $\mathit{exp}(S)$), is defined by:

$$\begin{aligned} \mathit{can}(S) &= \{D \in S \mid \forall D' \in S,\ D' \neq D \colon \{D'\} \not\models^{pc} D\}\,, \\ \mathit{exp}(S) &= \{D \in DHB_P \mid \exists D' \in S \colon \{D'\} \models^{pc} D\}\,. \end{aligned}$$

A state S is *in canonical form* (resp., *expanded*) iff $S = can(S)$ (resp., $S = exp(S)$).

The following theorem shows that the intersection of a set of model states of a positive mvd-program P is also a model state of P.

Theorem 3.1 *Let P be a positive mvd-program, and let $\mathcal{S}$ be a set of model states of P. Then, the intersection of all $S \in \mathcal{S}$ is a model state of P.*

Clearly, each positive mvd-program P has the model state DHB_P. Thus, there exist model states of P, and the intersection of all of them is the least model state of P.

Definition 3.2 Denote by MS_P the least model state of a positive mvd-program P.

The following result shows that MS_P is the set of all disjunctions $D \in DHB_P$ that are pc-consequences of P. Moreover, it shows that this set coincides with the set of all disjunctions $D \in DHB_P$ that are true in all minimal models of P.

Theorem 3.3 *Let P be a positive mvd-program. Then,*

(a) $MS_P = \{D \in DHB_P \mid P \models^{pc} D\}$.

(b) $MS_P = \{D \in DHB_P \mid \boldsymbol{MM}(P) \models D\}$.

As shown in [14, 15], definite mvd-programs P have a unique least model M_P. The next theorem shows that for such P, the model M_P corresponds to $can(MS_P)$.

Theorem 3.4 *Let P be a definite mvd-program, and let M_P be the least model of P. Then, $can(MS_P) = S_P$ where $S_P = \{A^\alpha \in DHB_P \mid \alpha = M_P(A)\}$.*

We give an illustrative example.

Example 3.5 Consider the following positive mvd-program P:

$$\begin{aligned} P = \{ & (closed(r) \vee closed(s) \leftarrow)[.5, 1], (road(r, a, b) \leftarrow)[.8, 1], (road(s, b, c) \leftarrow)[.7, 1], \\ & (reach(X, Y) \vee closed(R) \leftarrow road(R, X, Y))[.9, 1], \\ & (reach(X, Z) \leftarrow reach(X, Y), reach(Y, Z))[.9, 1]\} . \end{aligned}$$

The set of truth values of P is given by $TV(P) = \{0, 0.1, \ldots, 1\}$. The canonical form of the least model state MS_P of P is given as follows:

$$\begin{aligned} can(MS_P) \;=\; \{ & closed^{0.5}(r) \vee closed^{0.5}(s),\ road^{0.8}(r, a, b),\ road^{0.7}(s, b, c), \\ & reach^{0.7}(a, b) \vee closed^{0.7}(r),\ reach^{0.6}(b, c) \vee closed^{0.6}(s), \\ & reach^{0.5}(a, c) \vee closed^{0.7}(r) \vee closed^{0.6}(s)\} . \end{aligned}$$

4 Unfolding Many-Valuedness

In this section, we give translations of mvd-programs under the semantics of minimal models, perfect models, stable models, and least model states into classical disjunctive logic programs under the respective classical semantics.

4.1 Program Translations

We now formally define translations of mvd-programs and pcp-interpretations into classical disjunctive logic programs and classical interpretations, respectively.

Given an mvd-program P, the *many-valued alphabet for* P, denoted $\Phi^m(P)$, is obtained from $\Phi(P)$ by replacing each predicate symbol p by the new predicate symbols p^α with $\alpha \in TV(P)\backslash\{0\}$. The *many-valued Herbrand base* for P, denoted HB^m_P, is the Herbrand base over $\Phi^m(P)$. For atoms $A = p(t_1, \ldots, t_k)$ and $\alpha \in TV(P)$, the atom A^α over $\Phi^m(P)$ is defined as $p^\alpha(t_1, \ldots, t_k)$.

Every mvd-program P is translated into the following classical disjunctive logic program $\mathrm{Tr}(P) = \mathrm{Tr}_1(P) \cup \mathrm{Tr}_2(P)$ over $\Phi^m(P)$ (based on Theorem 2.1):

$$\begin{aligned}
\mathrm{Tr}_1(P) \;&=\; \{A_1^\alpha \vee \cdots \vee A_l^\alpha \leftarrow B_1^{\beta_1}, \ldots, B_m^{\beta_m}, not\, C_1^\alpha, \ldots, not\, C_n^\alpha \mid \\
&\qquad (A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, not\, C_1, \ldots, not\, C_n)[c, 1] \in P, \\
&\qquad \beta_1, \ldots, \beta_m \in TV(P),\ \alpha = c - 1 + \min(\beta_1, \ldots, \beta_m) > 0\}\,, \\
\mathrm{Tr}_2(P) \;&=\; \{A^\alpha \leftarrow A^\beta \mid A^\alpha,\, A^\beta \in HB^m_P,\, \alpha < \beta\}\,.
\end{aligned}$$

Every pcp-interpretation $\boldsymbol{p}$ is translated into the following classical interpretation:

$$\mathrm{Tr}(\boldsymbol{p}) \;=\; \{A^\alpha \in HB^m_P \mid \boldsymbol{p}(A) \geq \alpha\}\,.$$

The following example illustrates the above program translation.

Example 4.1 The mvd-program P given in Section 2.7 is translated into the classical disjunctive logic program $\mathrm{Tr}(P) = \mathrm{Tr}_1(P) \cup \mathrm{Tr}_2(P)$, where $\mathrm{Tr}_1(P)$ is given by:

$$\begin{aligned}
\mathrm{Tr}_1(P) \;=\; \{&closed^{0.5}(r) \vee closed^{0.5}(s) \leftarrow;\ road^{0.8}(r, a, b) \leftarrow;\ road^{0.7}(s, b, c) \leftarrow; \\
&reach^{0.1}(X, Y) \leftarrow road^{0.2}(R, X, Y), not\ closed^{0.1}(R); \\
&reach^{0.2}(X, Y) \leftarrow road^{0.3}(R, X, Y), not\ closed^{0.2}(R); \ldots; \\
&reach^{0.9}(X, Y) \leftarrow road^{1}(R, X, Y), not\ closed^{0.9}(R); \\
&reach^{0.1}(X, Z) \leftarrow reach^{0.2}(X, Y), reach^{0.2}(Y, Z); \\
&reach^{0.1}(X, Z) \leftarrow reach^{0.2}(X, Y), reach^{0.3}(Y, Z); \\
&reach^{0.1}(X, Z) \leftarrow reach^{0.3}(X, Y), reach^{0.2}(Y, Z); \\
&reach^{0.2}(X, Z) \leftarrow reach^{0.3}(X, Y), reach^{0.3}(Y, Z); \ldots; \\
&reach^{0.9}(X, Z) \leftarrow reach^{1}(X, Y), reach^{1}(Y, Z)\}\,.
\end{aligned}$$

Note that $\mathrm{Tr}_1(P)$ may be quite large. It generally has a manageable size when there are few truth values in $TV(P)$ and few positive literals in the bodies of clauses in P.

4.2 Minimal Models

The following lemma shows that every mvd-program P is equivalent to its translation $\mathrm{Tr}(P)$, under all pcp-interpretations into $TV(P)$.

Lemma 4.2 *Let P be an mvd-program, and let $\boldsymbol{p}$ be a pcp-interpretation into $TV(P)$. Then, $\boldsymbol{p}$ is a model of P iff $\boldsymbol{p}$ is a model of* $\mathrm{Tr}(P)$.

The next lemma shows that pcp-interpretations $\boldsymbol{p}$ into $TV(P)$ can be identified with their translation $\mathrm{Tr}(\boldsymbol{p})$, concerning classical disjunctive logic programs over HB_P^m.

Lemma 4.3 *Let P be an mvd-program. Let L be a classical disjunctive logic program over the alphabet $\Phi^m(P)$, and let $\boldsymbol{p}$ be a pcp-interpretation into $TV(P)$. Then, $\boldsymbol{p}$ is a model of L iff $\mathrm{Tr}(\boldsymbol{p})$ is a model of L.*

The following theorem shows that Tr translates mvd-programs under the minimal model semantics into equivalent classical disjunctive logic programs under the minimal model semantics. It can be proved using the two lemmata above.

Theorem 4.4 *Let P be an mvd-program, and let $\boldsymbol{p}$ be a pcp-interpretation. Then, $\boldsymbol{p}$ is a minimal model of P iff $\mathrm{Tr}(\boldsymbol{p})$ is a minimal model of $\mathrm{Tr}(P)$.*

4.3 Perfect Models

The alphabet $\Phi_0^m(P)$ is obtained from $\Phi(P)$ by replacing each predicate symbol p by the new predicate symbols p^α with $\alpha \in TV(P)$.

We slightly modify the translation of mvd-programs and pcp-interpretations as follows. Every mvd-program P is translated into the following classical disjunctive logic program $\mathrm{Tr}^\star(P) = \mathrm{Tr}(P) \cup \mathrm{Tr}_3(P)$ over $\Phi_0^m(P)$:

$$\mathrm{Tr}_3(P) \;=\; \{A^0 \vee A^1 \vee A^{n-1} \leftarrow \mid A \in HB_P\} \cup \{A^0 \leftarrow \mid A \in HB_P\}\,.$$

Every pcp-interpretation $\boldsymbol{p}$ is translated into the following classical interpretation:

$$\mathrm{Tr}^\star(\boldsymbol{p}) \;=\; \mathrm{Tr}(\boldsymbol{p}) \cup \{A^0 \mid A \in HB_P\}\,.$$

Roughly speaking, the next lemma shows that pcp-interpretations $\boldsymbol{p}$ into $TV(P)$ can be identified with their translation $\mathrm{Tr}^\star(\boldsymbol{p})$.

Lemma 4.5 *Let P be an mvd-program. Let L be a classical disjunctive logic program over the alphabet $\Phi_0^m(P)$, and let $\boldsymbol{p}$ be a pcp-interpretation into $TV(P)$. Then, $\boldsymbol{p}$ is a model of L iff $\mathrm{Tr}^\star(\boldsymbol{p})$ is a model of $L \cup \mathrm{Tr}_3(P)$.*

The following theorem shows that $\mathrm{Tr}^\star$ translates mvd-programs under the perfect model semantics into equivalent classical counterparts.

Theorem 4.6 *Let P be an mvd-program, and let $\boldsymbol{p}$ be a pcp-interpretation. Then, $\boldsymbol{p}$ is a perfect model of P iff $\mathrm{Tr}^\star(\boldsymbol{p})$ is a perfect model of $\mathrm{Tr}^\star(P)$.*

The following theorem shows that the translation $\mathrm{Tr}(P)$ of a locally stratified mvd-program P is also locally stratified.

Theorem 4.7 *Let P be an mvd-program. If P is locally stratified, then also* $\mathrm{Tr}(P)$.

The next theorem shows that Tr translates locally stratified mvd-programs under the perfect model semantics into equivalent classical counterparts.

Theorem 4.8 *Let P be a locally stratified mvd-program, and let $\boldsymbol{p}$ be a pcp-interpretation. Then, $\boldsymbol{p}$ is a perfect model of P iff* $\mathrm{Tr}(\boldsymbol{p})$ *is a perfect model of* $\mathrm{Tr}(P)$.

4.4 Stable Models

For classical disjunctive logic programs L and classical interpretations I, denote by L/I the classical Gelfond-Lifschitz transform of L with respect to I.

The next lemma shows that for mvd-programs P and pcp-interpretations $\boldsymbol{q}$, the transform $P/\boldsymbol{q}$ is equivalent to $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{q})$, under all pcp-interpretations into $TV(P)$.

Lemma 4.9 *Let P be an mvd-program, and let $\boldsymbol{p}$ and $\boldsymbol{q}$ be two pcp-interpretations into $TV(P)$. Then, $\boldsymbol{p}$ is a model of $P/\boldsymbol{q}$ iff $\boldsymbol{p}$ is a model of* $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{q})$.

The next theorem shows that Tr translates mvd-programs under the stable model semantics into equivalent classical counterparts.

Theorem 4.10 *Let P be an mvd-program, and let $\boldsymbol{p}$ be a pcp-interpretation. Then, $\boldsymbol{p}$ is a stable model of P iff* $\mathrm{Tr}(\boldsymbol{p})$ *is a stable model of* $\mathrm{Tr}(P)$.

4.5 Least Model States

The following lemma shows that every mvd-program P is equivalent to its translation $\mathrm{Tr}(P)$, concerning disjunctive Herbrand states.

Lemma 4.11 *Let P be an mvd-program, and let S be a state. Then, S is a model state of P iff S is a model state of* $\mathrm{Tr}(P)$.

The following theorem shows that Tr translates an mvd-program into a classical counterpart that has the same least model state.

Theorem 4.12 *Let P be an mvd-program, and let S be a state. Then, S is the least model state of P iff S is the least model state of* $\mathrm{Tr}(P)$.

5 Fixpoint Characterizations

In this section, we provide many-valued fixpoint characterizations for the semantics of minimal models, least model states, and perfect models under finite local stratification.

5.1 Minimal Models for Positive Programs

We now give a fixpoint characterization for the set of all minimal models of a positive mvd-program, which is a generalization of the classical counterpart given in [7, 29].

In the sequel, let P be a positive mvd-program. The *canonical form* (resp., *expansion*) of a set of pcp-interpretations $\boldsymbol{P}$, denoted $can(\boldsymbol{P})$ (resp., $exp(\boldsymbol{P})$), is defined by:

$$\begin{aligned} can(\boldsymbol{P}) &= \{\boldsymbol{p} \in \boldsymbol{P} \mid \neg\exists \boldsymbol{q} \in \boldsymbol{P}\colon \boldsymbol{q} \subset \boldsymbol{p}\}\,, \\ exp(\boldsymbol{P}) &= \{\boldsymbol{p} \in \boldsymbol{I}_P \mid \exists \boldsymbol{q} \in \boldsymbol{P}\colon \boldsymbol{q} \subseteq \boldsymbol{p}\}\,. \end{aligned}$$

We say $\boldsymbol{P}$ is *in canonical form* (resp., *expanded*) iff $\boldsymbol{P} = can(\boldsymbol{P})$ (resp., $\boldsymbol{P} = exp(\boldsymbol{P})$).

The fixpoint operator is defined on the complete lattice $(\mathcal{E}, \sqsubseteq)$, where $\mathcal{E}$ is the set of all expanded sets of pcp-interpretations, and $\boldsymbol{P} \sqsubseteq \boldsymbol{Q}$ iff $\boldsymbol{Q} \supseteq \boldsymbol{P}$ for all $\boldsymbol{P}, \boldsymbol{Q} \in \mathcal{E}$. The bottom element $\bot$ is the set of all pcp-interpretations, and the top element $\top$ is the empty set. The greatest lower bound of any subset of elements is the union of the elements in the set, and the least upper bound is the intersection of the elements.

The operator T_P^M on expanded sets of pcp-interpretations $\boldsymbol{P}$ is defined by:

$$T_P^M(\boldsymbol{P}) \;=\; \bigcup \{models_{\boldsymbol{p}}(state_P(\boldsymbol{p})) \mid \boldsymbol{p} \in \boldsymbol{P}\},$$

where $state_P$ and $models_{\boldsymbol{p}}$ are given as follows:

$$\begin{aligned} state_P(\boldsymbol{p}) &= \{A_1^\alpha \vee \cdots \vee A_l^\alpha \mid (A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m)[c,1] \in ground(P), \\ &\qquad \alpha = c - 1 + \min(\boldsymbol{p}(B_1), \ldots, \boldsymbol{p}(B_m)) > 0\}\,, \\ models_{\boldsymbol{p}}(S) &= \{\boldsymbol{q} \in \boldsymbol{I}_P \mid \boldsymbol{q} \models S,\ \boldsymbol{q} \supseteq \boldsymbol{p}\}\,. \end{aligned}$$

The next lemma shows the immediate result that T_P^M is monotonic.

Lemma 5.1 *T_P^M is monotonic.*

We now define the powers of T_P^M. For every expanded set of pcp-interpretations $\boldsymbol{P}$:

$$T_P^M\!\uparrow \alpha(\boldsymbol{P}) \;=\; \begin{cases} \boldsymbol{P} & \text{if } \alpha = 0; \\ T_P^M(T_P^M\!\uparrow (\alpha{-}1)(\boldsymbol{P})) & \text{if } \alpha > 0 \text{ is a successor ordinal;} \\ \bigcap\{T_P^M\!\uparrow \beta(\boldsymbol{P}) \mid \beta < \alpha\} & \text{if } \alpha > 0 \text{ is a limit ordinal.} \end{cases}$$

As usual, we use $T_P^M\!\uparrow \alpha$ to abbreviate $T_P^M\!\uparrow \alpha(\bot)$.

The following lemma shows that the operator T_P^M is not continuous. This result is immediate by the fact that the classical counterpart of T_P^M is not continuous [29].

Lemma 5.2 *T_P^M is not continuous.*

Even though the operator T_P^M is not continuous, its least fixpoint is attained at the first limit ordinal. This is shown by the following theorem, which follows from a similar result for classical disjunctive logic programs [29].

Theorem 5.3 $\mathit{lfp}(T_P^M) = T_P^M \uparrow \omega$.

The next theorem shows that the set of minimal models of P is given by the canonical form of the least fixpoint of T_P^M.

Theorem 5.4 $\boldsymbol{MM}(P) = \mathit{can}(\mathit{lfp}(T_P^M))$.

5.2 Least Model States for Positive Programs

We now give a fixpoint characterization for the least model state of a positive mvd-program, which is a generalization of the classical counterpart given in [20, 12].

In the sequel, let P be a positive mvd-program. We now identify every disjunction $D \in \mathit{DHB}_P$ with the set of all contained atoms $A^\alpha \in \mathit{HB}_P^m$.

The operator T_P^s on expanded disjunctive Herbrand states S is defined by:

$$\begin{aligned} T_P^s(S) \;=\; & \mathit{exp}(\{A_1^\alpha \vee \cdots \vee A_l^\alpha \vee D_1 \vee \cdots \vee D_m \mid D_1, \ldots, D_m \in \mathit{DHB}_P, \\ & (A_1 \vee \cdots \vee A_l \leftarrow B_1 \wedge \cdots \wedge B_m)[c, 1] \in \mathit{ground}(P), \\ & B_1^{\beta_1} \vee D_1, \ldots, B_m^{\beta_m} \vee D_m \in S,\; \alpha = c - 1 + \min(\beta_1, \ldots, \beta_m) > 0\}) \,. \end{aligned}$$

The following lemma shows that the model states of P correspond exactly to the pre-fixpoints of the operator T_P^s.

Lemma 5.5 *Let S be an expanded state. Then, S is a model state of P iff $T_P^s(S) \subseteq S$.*

The next lemma shows that the operator T_P^s is continuous. This result follows immediately from the continuity of the classical counterpart of T_P^s [20].

Lemma 5.6 *T_P^s is continuous.*

The powers of T_P^s are defined as usual: For all Herbrand states S, define $T_P^s \uparrow \omega(S)$ as the union of all $T_P^s \uparrow n(S)$ with $n < \omega$, where $T_P^s \uparrow 0(S) = S$ and $T_P^s \uparrow (n+1)(S) = T_P^s(T_P^s \uparrow n(S))$ for all $n < \omega$. We use $T_P^s \uparrow \omega$ to abbreviate $T_P^s \uparrow \omega(\emptyset)$.

The following theorem shows that the least model state of P coincides with the least fixpoint of T_P^s, and that the least fixpoint is attained at the first limit ordinal. This result follows immediately from Lemmata 5.5 and 5.6.

Theorem 5.7 $\mathit{MS}_P = \mathit{lfp}(T_P^s) = T_P^s \uparrow \omega$.

We give an illustrative example.

Example 5.8 Consider again the positive mvd-program P given in Example 3.5. Its least model state MS_P is given by $T_P^s \uparrow \omega = T_P^s \uparrow 3$:

$$\begin{aligned} \mathit{can}(T_P^s \uparrow 1) &= S_1 = \{\mathit{closed}^{0.5}(r) \vee \mathit{closed}^{0.5}(s),\; \mathit{road}^{0.8}(r, a, b),\; \mathit{road}^{0.7}(s, b, c)\}, \\ \mathit{can}(T_P^s \uparrow 2) &= S_2 = S_1 \cup \{\mathit{reach}^{0.7}(a, b) \vee \mathit{closed}^{0.7}(r),\; \mathit{reach}^{0.6}(b, c) \vee \mathit{closed}^{0.6}(s)\}, \\ \mathit{can}(T_P^s \uparrow 3) &= S_3 = S_2 \cup \{\mathit{reach}^{0.5}(a, c) \vee \mathit{closed}^{0.7}(r) \vee \mathit{closed}^{0.6}(s)\} \,. \end{aligned}$$

5.3 Perfect Models under Finite Local Stratification

We now give an iterative fixpoint characterization of perfect models of mvd-programs with finite local stratification. It generalizes the classical counterpart in [29].

For sets of emvd-clauses P and sets of expanded interpretations $\boldsymbol{P}$, we define:

$$\overline{T}_P^M(\boldsymbol{P}) \;=\; \bigcup\{\mathit{models}_{\boldsymbol{p}}(\overline{\mathit{state}}_P(\boldsymbol{p})) \mid \boldsymbol{p} \in \boldsymbol{P}\}\,,$$

where $\mathit{models}_{\boldsymbol{p}}$ is defined as in Section 5.1 and $\overline{\mathit{state}}_P$ is given by:

$$\begin{aligned}\overline{\mathit{state}}_P(\boldsymbol{p}) \;=\; \{A_1^\alpha \vee \cdots \vee A_l^\alpha \mid (A_1 \vee \cdots \vee A_l; d \leftarrow B_1, \ldots, B_m)[c,1] \in \mathit{ground}(P)\,,\\ \alpha = c - 1 + \min(\boldsymbol{p}(B_1), \ldots, \boldsymbol{p}(B_m)) > d\}\,.\end{aligned}$$

The following theorem formulates the iterative fixpoint characterization.

Theorem 5.9 *Let P be an mvd-program and let $H_1, H_2, \ldots, H_n$ be a finite local stratification of P. For pcp-interpretations $\boldsymbol{p}$, we define:*

$$P_i(\boldsymbol{p}) = P_i/\boldsymbol{p} \cup \{A^\alpha \leftarrow \mid A^\alpha \in HB_P^m,\, \boldsymbol{p}(A) \geq \alpha\}\,.$$

Then, the set of perfect models of P is given as $\boldsymbol{P}_n$, where

$$\boldsymbol{P}_1 = \mathit{can}(T_{P_1}^M\!\uparrow\omega)\,,$$

$$\boldsymbol{P}_i = \bigcup\{\mathit{can}(\overline{T}_{P_i(\boldsymbol{p})}^M\!\uparrow\omega) \mid \boldsymbol{p} \in \boldsymbol{P}_{i-1}\} \text{ for all } i \in \{2, \ldots, n\}\,.$$

6 Summary and Outlook

We introduced least model states for many-valued disjunctive logic programs. We then showed how to unfold many-valuedness under the semantics of minimal models, perfect models, stable models, and least model states. Thus, existing technology for classical disjunctive logic programming can be used to implement many-valued disjunctive logic programming. Using these results, we gave many-valued fixpoint characterizations for the set of all minimal models and the least model state. We also gave an iterative fixpoint characterization for the perfect model semantics under finite local stratification.

An interesting topic of future research is to elaborate other semantics for many-valued disjunctive logic programs, for example, to define partial stable models. Moreover, it would be very interesting to work out fixpoint characterizations for stable (and partial stable) models. This may be done by generalizing the evidential transformation in [6] or the 3-S transformation in [28]. Finally, another topic of future research is to elaborate proof theories for the various semantics.

A Appendix: Proofs

A.1 Proofs for Section 3

Proof of Theorem 3.1. Towards a contradiction, assume $\bigcap\mathcal{S}$ is not a model state of P. That is, $P \cup \bigcap\mathcal{S} \models^{pc} D$ for some $D \in \mathit{DHB}_P \setminus \bigcap\mathcal{S}$. Thus, some $S \in \mathcal{S}$ exists such that $D \notin S$. By the monotonicity of $\models^{pc}$, it follows $P \cup S \models^{pc} D$. But this contradicts S being a model state of P. It thus follows that $\bigcap\mathcal{S}$ is a model state of P. □

For the proof of Theorem 3.3, we recall the following result from [13].

Lemma A.1 *Let P be an mvd-program. Then, for every model $\boldsymbol{q}$ of P, there exists some minimal model $\boldsymbol{p}$ of P such that $\boldsymbol{p} \subseteq \boldsymbol{q}$.*

Proof of Theorem 3.3. (a) Clearly, $S = \{D \in \mathit{DHB}_P \mid P \models^{pc} D\}$ is a model state of P. As $\models^{pc}$ is monotonic, S is contained in all model states of P. Thus, S is equal to MS_P.

(b) Using (a), it is sufficient to show that $P \models^{pc} D$ iff $\boldsymbol{MM}(P) \models D$, for all $D \in \mathit{DHB}_P$. Trivially, $P \models^{pc} D$ implies that $\boldsymbol{p} \models D$ for all $\boldsymbol{p} \in \boldsymbol{MM}(P)$. Conversely, assume that $\boldsymbol{p} \models D$ for all $\boldsymbol{p} \in \boldsymbol{MM}(P)$. Suppose now that there exists a model $\boldsymbol{p}$ of P that is not a model of D. By Lemma A.1, some $\boldsymbol{q} \in \boldsymbol{MM}(P)$ exists such that $\boldsymbol{q} \subseteq \boldsymbol{p}$. As $\boldsymbol{p}$ is not a model of D, also $\boldsymbol{q}$ is not a model of D. But this contradicts $\boldsymbol{p} \models D$ for all $\boldsymbol{p} \in \boldsymbol{MM}(P)$. Hence, $\boldsymbol{p} \models D$ for all models $\boldsymbol{p}$ of P, that is, $P \models^{pc} D$. □

Proof of Theorem 3.4. By Theorem 3.3 (b), we get $\mathit{MS}_P = \{D \in \mathit{DHB}_P \mid M_P \models D\}$. Thus, $D = A_1^{\alpha_1} \vee \cdots \vee A_k^{\alpha_k} \in \mathit{MS}_P$ implies that $M_P(A_i) \geq \alpha_i$ for some $i \in \{1, \ldots, k\}$, and thus $\{A_i^{\alpha}\} \models^{pc} D$ for $\alpha = M_P(A_i)$. Hence, we get $\mathit{can}(\mathit{MS}_P) \subseteq S_P$. Clearly, $\{A^{\alpha}\} \not\models^{pc} B^{\beta}$ for any two distinct $A^{\alpha}, B^{\beta} \in S_P$. Hence, $\mathit{can}(\mathit{MS}_P) = S_P$. □

A.2 Proofs for Section 4

Proof of Lemma 4.2. ($\Rightarrow$) Assume $\boldsymbol{p}$ is a model of P. That is, $\boldsymbol{p}$ is a model of every $C = (A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, \mathit{not}\, C_1, \ldots, \mathit{not}\, C_n)[c, 1] \in P$. Equivalently,

$$\max(\max_{1 \leq i \leq l} \boldsymbol{p}(A_i), \max_{1 \leq i \leq n} \boldsymbol{p}(C_i)) \;\geq\; c - 1 + \min_{1 \leq i \leq m} \boldsymbol{p}(B_i) \tag{2}$$

for every such $C \in P$. We now show that $\boldsymbol{p}$ is a model of $\mathrm{Tr}(P) = \mathrm{Tr}_1(P) \cup \mathrm{Tr}_2(P)$. Clearly, $\boldsymbol{p}$ is a model of $\mathrm{Tr}_2(P)$. We next show that $\boldsymbol{p}$ is also a model of $\mathrm{Tr}_1(P)$. Let $D = A_1^{\alpha} \vee \cdots \vee A_l^{\alpha} \leftarrow B_1^{\beta_1}, \ldots, B_m^{\beta_m}, \mathit{not}\, C_1^{\alpha}, \ldots, \mathit{not}\, C_n^{\alpha} \in \mathrm{Tr}_1(P)$. Suppose first $\boldsymbol{p}(B_i) < \beta_i$ for some $1 \leq i \leq m$. That is, $\boldsymbol{p}$ is a model of $\neg B_i^{\beta_i}$ for some $1 \leq i \leq m$, and thus $\boldsymbol{p}$ is also a model of D. Suppose next $\boldsymbol{p}(B_i) \geq \beta_i$ for all $1 \leq i \leq m$. By (2), we get:

$$\max(\max_{1 \leq i \leq l} \boldsymbol{p}(A_i), \max_{1 \leq i \leq n} \boldsymbol{p}(C_i)) \;\geq\; c - 1 + \min_{1 \leq i \leq m} \beta_i \;=\; \alpha\,.$$

Thus, also in this case, p is a model of D. Hence, $\boldsymbol{p}$ is a model of $\mathrm{Tr}_1(P)$.

($\Leftarrow$) Assume $\boldsymbol{p}$ is a model of $\mathrm{Tr}(P)$. We now show that $\boldsymbol{p}$ is also a model of P. Let $C = (A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, not\, C_1, \ldots, not\, C_n)[c,1] \in P$. Let $\alpha, \beta_1, \ldots, \beta_m \in TV(P)$ be defined by $\beta_i = \boldsymbol{p}(B_i)$ for all $1 \leq i \leq m$ and $\alpha = c - 1 + \min(\beta_1, \ldots, \beta_m)$. Suppose first $\alpha \leq 0$. Then, $\boldsymbol{p}$ is clearly a model of C. Suppose next $\alpha > 0$. As $\boldsymbol{p}$ is a model of $\mathrm{Tr}(P)$, it is in particular a model of $A_1^\alpha \vee \cdots \vee A_l^\alpha \leftarrow B_1^{\beta_1}, \ldots, B_m^{\beta_m}, not\, C_1^\alpha, \ldots, not\, C_n^\alpha \in \mathrm{Tr}_1(P)$. Thus, as $\boldsymbol{p}$ is a model of all $B_i^{\beta_i}$ with $1 \leq i \leq m$, it is also a model of some A_i^α with $1 \leq i \leq l$ or some C_i^α with $1 \leq i \leq n$. Thus,

$$\max(\max_{1 \leq i \leq l} \boldsymbol{p}(A_i), \max_{1 \leq i \leq n} \boldsymbol{p}(C_i)) \;\geq\; \alpha \;=\; c-1+\min_{1 \leq i \leq m} \beta_i \;=\; c-1+\min_{1 \leq i \leq m} \boldsymbol{p}(B_i)\,.$$

Thus, also in this case, $\boldsymbol{p}$ is a model of C. Hence, $\boldsymbol{p}$ is a model of P. $\Box$

Proof of Lemma 4.3. Consider any $A^\alpha \in HB_P^m$. Then, $\boldsymbol{p} \models A^\alpha$ iff $\boldsymbol{p}(A) \geq \alpha$ iff $A^\alpha \in \mathrm{Tr}(\boldsymbol{p})$ iff $\mathrm{Tr}(\boldsymbol{p}) \models A^\alpha$. This proves the claim. $\Box$

For the proof of Theorem 4.4, we recall the following result from [13].

Theorem A.2 *Every minimal model of an mvd-program P maps into $TV(P)$.*

Proof of Theorem 4.4. Observe that for all pcp-interpretations $\boldsymbol{p}$ and $\boldsymbol{q}$ into $TV(P)$, it holds that $\boldsymbol{p} \subseteq \boldsymbol{q}$ iff $\mathrm{Tr}(\boldsymbol{p}) \subseteq \mathrm{Tr}(\boldsymbol{q})$. Assume now $\boldsymbol{p}$ is a minimal model of P. By Lemma A.2, this is equivalent to $\boldsymbol{p}$ being a minimal model of P among all pcp-interpretations into $TV(P)$. By Lemma 4.2, this is in turn equivalent to $\boldsymbol{p}$ being a minimal model of $\mathrm{Tr}(P)$ among all pcp-interpretations into $TV(P)$. Equivalently, by Lemma 4.3 and the above observation, $\mathrm{Tr}(\boldsymbol{p})$ is a minimal model of $\mathrm{Tr}(P)$. $\Box$

Proof of Lemma 4.5. Immediate by Lemma 4.3 and the observation that $\mathrm{Tr}(\boldsymbol{p})$ is a model of L iff $\mathrm{Tr}^\star(\boldsymbol{p})$ is a model of $L \cup \mathrm{Tr}_3(P)$. $\Box$

For the proof of Theorem 4.6, we recall the following result from [13].

Theorem A.3 *Every perfect model of an mvd-program P is also a minimal model of P.*

Proof of Theorem 4.6. Let $\ll_P$ and $\ll_{\mathrm{Tr}^\star(P)}$ denote the preference relations on pcp-interpretations w.r.t. P and $\mathrm{Tr}^\star(P)$, respectively. We first show that for all pcp-interpretations $\boldsymbol{p}$ and $\boldsymbol{q}$ into $TV(P)$, it holds that $\boldsymbol{p} \ll_P \boldsymbol{q}$ iff $\mathrm{Tr}^\star(\boldsymbol{p}) \ll_{\mathrm{Tr}^\star(P)} \mathrm{Tr}^\star(\boldsymbol{q})$. Recall that $\boldsymbol{p} \ll_P \boldsymbol{q}$ iff $\boldsymbol{p} \neq \boldsymbol{q}$ and for each $A \in HB_P$ with $\boldsymbol{p}(A) > \boldsymbol{q}(A)$ there is some $B \in HB_P$ with $\boldsymbol{q}(B) > \boldsymbol{p}(B)$ and $A \prec B$. Equivalently, $\mathrm{Tr}^\star(\boldsymbol{p}) \neq \mathrm{Tr}^\star(\boldsymbol{q})$ and for each $A^\alpha \in \mathrm{Tr}^\star(\boldsymbol{p}) \setminus \mathrm{Tr}^\star(\boldsymbol{q})$ there is some $B^\beta \in \mathrm{Tr}^\star(\boldsymbol{q}) \setminus \mathrm{Tr}^\star(\boldsymbol{p})$ and $A^\alpha \prec B^\beta$.

Assume now $\boldsymbol{p}$ is a $\ll_P$-perfect model of P. By Lemmata A.2 and A.3, this is equivalent to $\boldsymbol{p}$ being a $\ll_P$-perfect model of P among all pcp-interpretations into $TV(P)$. By Lemma 4.2, this is equivalent to $\boldsymbol{p}$ being a $\ll_P$-perfect model of $\mathrm{Tr}(P)$ among all pcp-interpretations into $TV(P)$. Equivalently, by Lemma 4.5 and the above observation, $\mathrm{Tr}^\star(\boldsymbol{p})$ is a $\ll_{\mathrm{Tr}^\star(P)}$-perfect model of $\mathrm{Tr}^\star(P)$. $\Box$

Proof of Theorem 4.7. Assume $H_1, H_2, \ldots$ is a local stratification of P. It is easy to verify that $H_1^m = \{A^\alpha \in HB_P^m \mid A \in H_1, \alpha > 0\}$, $H_2^m = \{A^\alpha \in HB_P^m \mid A \in H_2, \alpha > 0\}, \ldots$ is a local stratification of $\mathrm{Tr}(P)$. $\Box$

Proof of Theorem 4.8. By Theorem 4.6, $\boldsymbol{p}$ is a perfect model of P iff $\mathrm{Tr}^{\star}(\boldsymbol{p})$ is a perfect model of $\mathrm{Tr}^{\star}(P)$. As perfect models are iterative minimal models along the local stratification [13], the latter is equivalent to $\mathrm{Tr}(\boldsymbol{p})$ being a perfect model of $\mathrm{Tr}(P)$. $\Box$

Proof of Lemma 4.9. ($\Rightarrow$) Assume that $\boldsymbol{p}$ is a model of $P/\boldsymbol{q}$. That is, for every $C = (A_1 \vee \cdots \vee A_l \leftarrow B_1, \ldots, B_m, \mathit{not}\, C_1, \ldots, \mathit{not}\, C_n)[c, 1] \in P$, it holds that $\boldsymbol{p}$ is a model of $(A_1 \vee \cdots \vee A_l; \max(\boldsymbol{q}(C_1), \ldots, \boldsymbol{q}(C_n)) \leftarrow B_1, \ldots, B_m)[c, 1]$. Equivalently,

$$\max(\max_{1\leq i\leq l} \boldsymbol{p}(A_i), \max_{1\leq i\leq n} \boldsymbol{q}(C_i)) \;\geq\; c - 1 + \min_{1\leq i\leq m} \boldsymbol{p}(B_i) \tag{3}$$

for all such $C \in P$. We now show that $\boldsymbol{p}$ is a model of $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{q}) = \mathrm{Tr}_1(P)/\mathrm{Tr}(\boldsymbol{q}) \cup \mathrm{Tr}_2(P)$. Clearly, $\boldsymbol{p}$ is a model of $\mathrm{Tr}_2(P)$. We next show that $\boldsymbol{p}$ also is a model of $\mathrm{Tr}_1(P)/\mathrm{Tr}(\boldsymbol{q})$. Take any $D = A_1^{\alpha} \vee \cdots \vee A_l^{\alpha} \vee I[C_1^{\alpha}] \vee \cdots \vee I[C_n^{\alpha}] \leftarrow B_1^{\beta_1}, \ldots, B_m^{\beta_m} \in \mathrm{Tr}_1(P)/\mathrm{Tr}(\boldsymbol{q})$, where $I = \mathrm{Tr}(\boldsymbol{q})$, and thus $I[C_i^{\alpha}] = \mathbf{true}$ iff $\boldsymbol{q}(C_i) \geq \alpha$, for all $1 \leq i \leq n$. Suppose first $\boldsymbol{p}(B_i) < \beta_i$ for some $1 \leq i \leq m$. That is, $\boldsymbol{p}$ is a model of $\neg B_i^{\beta_i}$ for some $1 \leq i \leq m$, and thus $\boldsymbol{p}$ is also a model of D. Suppose next $\boldsymbol{p}(B_i) \geq \beta_i$ for all $1 \leq i \leq m$. Assume now $\boldsymbol{q}(C_i) \geq \alpha$ for some $1 \leq i \leq n$. Then, $\boldsymbol{p}$ is clearly a model of D. Assume finally that $\boldsymbol{q}(C_i) < \alpha$ for all $1 \leq i \leq n$. Using (2), we get:

$$\max_{1\leq i\leq l} \boldsymbol{p}(A_i) \;=\; \max(\max_{1\leq i\leq l} \boldsymbol{p}(A_i), \max_{1\leq i\leq n} \boldsymbol{q}(C_i)) \;\geq\; c - 1 + \min_{1\leq i\leq m} \beta_i \;=\; \alpha\,.$$

Thus, also in this case, p is a model of D. Hence, $\boldsymbol{p}$ is a model of $\mathrm{Tr}_1(P)/\mathrm{Tr}(\boldsymbol{q})$.

($\Leftarrow$) Assume $\boldsymbol{p}$ is a model of $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{q})$. We now show that $\boldsymbol{p}$ is also a model of $P/\boldsymbol{q}$. Take any $C = (A_1 \vee \cdots \vee A_l; \max(\boldsymbol{q}(C_1), \ldots, \boldsymbol{q}(C_n)) \leftarrow B_1, \ldots, B_m)[c, 1] \in P/\boldsymbol{q}$. Let $\alpha, \beta_1, \ldots, \beta_m \in TV(P)$ be defined by $\beta_i = \boldsymbol{p}(B_i)$ for all $1 \leq i \leq m$ and by $\alpha = c - 1 + \min(\beta_1, \ldots, \beta_m)$. Suppose first $\alpha \leq 0$. Then, $\boldsymbol{p}$ is clearly a model of C. Suppose next $\alpha > 0$. Assume now $\boldsymbol{q}(C_i) \geq \alpha$ for some $1 \leq i \leq n$. Then, $\boldsymbol{p} \models C$, as

$$\max(\max_{1\leq i\leq l} \boldsymbol{p}(A_i), \max_{1\leq i\leq n} \boldsymbol{q}(C_i)) \;\geq\; \alpha \;=\; c{-}1{+} \min_{1\leq i\leq m} \beta_i \;=\; c{-}1{+} \min_{1\leq i\leq m} \boldsymbol{p}(B_i)\,.$$

Assume finally $\boldsymbol{q}(C_i) < \alpha$ for all $1 \leq i \leq n$. As $\boldsymbol{p}$ is a model of $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{q})$, it is in particular a model of $A_1^{\alpha} \vee \cdots \vee A_l^{\alpha} \vee I[C_1^{\alpha}] \vee \cdots \vee I[C_n^{\alpha}] \leftarrow B_1^{\beta_1}, \ldots, B_m^{\beta_m} \in \mathrm{Tr}_1(P)/\mathrm{Tr}(\boldsymbol{q})$. As $\boldsymbol{p} \models B_i^{\beta_i}$ for all $1 \leq i \leq m$, it follows $\boldsymbol{p} \models A_i^{\alpha}$ for some $1 \leq i \leq l$. Thus,

$$\max(\max_{1\leq i\leq l} \boldsymbol{p}(A_i), \max_{1\leq i\leq n} \boldsymbol{q}(C_i)) \;\geq\; \alpha \;=\; c{-}1{+} \min_{1\leq i\leq m} \beta_i \;=\; c{-}1{+} \min_{1\leq i\leq m} \boldsymbol{p}(B_i)\,.$$

Thus, also in this case, $\boldsymbol{p}$ is a model of C. Hence, $\boldsymbol{p}$ is a model of $P/\boldsymbol{q}$. $\Box$

For the proof of Theorem 4.10, we recall the following result from [13].

Theorem A.4 *Every stable model of an mvd-program P is also a minimal model of P.*

Proof of Theorem 4.10. Observe that for all pcp-interpretations $\boldsymbol{p}$ and $\boldsymbol{q}$ into $TV(P)$, it holds that $\boldsymbol{p} \subseteq \boldsymbol{q}$ iff $\mathrm{Tr}(\boldsymbol{p}) \subseteq \mathrm{Tr}(\boldsymbol{q})$. Assume now that $\boldsymbol{p}$ is a stable model of P. That is, $\boldsymbol{p}$ is a minimal model of $P/\boldsymbol{p}$. By Lemmata A.2 and A.4, this is equivalent to $\boldsymbol{p}$ being a minimal model of $P/\boldsymbol{p}$ among all pcp-interpretations into $TV(P)$. By Lemma 4.9, this is equivalent to $\boldsymbol{p}$ being a minimal model of $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{p})$ among all pcp-interpretations into $TV(P)$. By Lemma 4.3 and the above observation, this is in turn equivalent to $\mathrm{Tr}(\boldsymbol{p})$ being a minimal model of $\mathrm{Tr}(P)/\mathrm{Tr}(\boldsymbol{p})$. $\Box$

Proof of Lemma 4.11. We show that $S \cup P \models^{pc} D$ is equivalent to $S \cup \mathrm{Tr}(P) \models^{pc} D$, for all $D \in DHB_P$. This then proves the claim. Observe first that D is true in all models of $S \cup P$ iff D is true in all models of $S \cup P$ into $TV(P)$. By Lemma 4.2, the latter is equivalent to D being true in all models of $S \cup \mathrm{Tr}(P)$ into $TV(P)$. This is in turn equivalent to D being true in all models of $S \cup \mathrm{Tr}(P)$. $\Box$

Proof of Theorem 4.12. Immediate by Lemma 4.11. $\Box$

A.3 Proofs for Section 5

The following lemma shows that T_P^M produces expanded sets of pcp-interpretations.

Lemma A.5 *Let $\boldsymbol{P}$ be a set of pcp-interpretations. Then, $T_P^M(\boldsymbol{P})$ is expanded.*

Proof. Consider any $\boldsymbol{p} \in T_P^M(\boldsymbol{P})$ and any $\boldsymbol{q} \in \boldsymbol{I}_P$ such that $\boldsymbol{q} \supseteq \boldsymbol{p}$. It then follows that $\boldsymbol{p} \in models_{\boldsymbol{p}'}(state_P(\boldsymbol{p}'))$ for some $\boldsymbol{p}' \in \boldsymbol{P}$. That is, $\boldsymbol{p} \models state_P(\boldsymbol{p}')$ and $\boldsymbol{p} \supseteq \boldsymbol{p}'$. Since $\boldsymbol{q} \supseteq \boldsymbol{p}$, it thus follows that $\boldsymbol{q} \models state_P(\boldsymbol{p}')$ and $\boldsymbol{q} \supseteq \boldsymbol{p}'$. But this means that $\boldsymbol{q}$ belongs to $models_{\boldsymbol{p}'}(state_P(\boldsymbol{p}'))$ and thus also to $T_P^M(\boldsymbol{P})$. $\Box$

The next result shows that the expanded sets of pcp-interpretations $\boldsymbol{P}$ that are fixpoints of T_P^M have a set of models of P as their canonical form.

Lemma A.6 *Let $\boldsymbol{P}$ be an expanded set of pcp-interpretations such that $T_P^M(\boldsymbol{P}) = \boldsymbol{P}$. Then, $can(\boldsymbol{P})$ is a set of models of P.*

Proof. Towards a contradiction, assume that $can(\boldsymbol{P})$ contains a pcp-interpretation $\boldsymbol{p}$ that is not a model of P. Since $T_P^M(\boldsymbol{P}) = \boldsymbol{P}$, it follows that $\boldsymbol{p}$ belongs to $T_P^M(\boldsymbol{P})$. That is, $\boldsymbol{p}$ belongs to $models_{\boldsymbol{p}'}(state_P(\boldsymbol{p}'))$ for some $\boldsymbol{p}' \in \boldsymbol{P}$. This shows in particular that $\boldsymbol{p} \supseteq \boldsymbol{p}'$ for some $\boldsymbol{p}' \in \boldsymbol{P}$. But this contradicts $\boldsymbol{p}$ being a member of $can(\boldsymbol{P})$. $\Box$

We next show that the expansion of any set of models of P is a fixpoint of T_P^M.

Lemma A.7 *Let $\boldsymbol{P}$ be a set of models of P. Then, $T_P^M(exp(\boldsymbol{P})) = exp(\boldsymbol{P})$.*

Proof. Observe that for every model $\boldsymbol{p}$ of P, it holds $models_{\boldsymbol{p}}(state_P(\boldsymbol{p})) = exp(\{\boldsymbol{p}\})$. It thus follows $T_P^M(\boldsymbol{P}) = \bigcup\{exp(\{\boldsymbol{p}\}) \mid \boldsymbol{p} \in \boldsymbol{P}\} = exp(\boldsymbol{P})$. As T_P^M is monotonic and $exp(\boldsymbol{P}) \supseteq \boldsymbol{P}$, we get $T_P^M(exp(\boldsymbol{P})) \supseteq exp(\boldsymbol{P})$. Assume now $T_P^M(exp(\boldsymbol{P})) \neq exp(\boldsymbol{P})$. That is, some $\boldsymbol{p} \in T_P^M(exp(\boldsymbol{P}))$ exists such that $\boldsymbol{p} \notin exp(\boldsymbol{P})$. It then follows that $\boldsymbol{p} \in models_{\boldsymbol{p}'}(state_P(\boldsymbol{p}'))$ for some $\boldsymbol{p}' \in exp(\boldsymbol{P})$. Thus, we get in particular $\boldsymbol{p} \supseteq \boldsymbol{p}'$ and $\boldsymbol{p}' \supseteq \boldsymbol{q}$ for some $\boldsymbol{q} \in \boldsymbol{P}$. Hence, $\boldsymbol{p} \supseteq \boldsymbol{q}$ for some $\boldsymbol{q} \in \boldsymbol{P}$. But this contradicts $\boldsymbol{p} \notin exp(\boldsymbol{P})$. It thus follows $T_P^M(exp(\boldsymbol{P})) = exp(\boldsymbol{P})$. $\Box$

Proof of Theorem 5.4. The statement follows immediately from a similar result for classical disjunctive logic programs in [29]. $\Box$

References

[1] K. R. Apt. Logic programming. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 10, pages 493–574. MIT Press, 1990.

[2] J. F. Baldwin. Evidential support logic programming. *Fuzzy Sets Syst.*, 24:1–26, 1987.

[3] R. Carnap. *Logical Foundations of Probability*. University of Chicago Press, Chicago, 1950.

[4] A. Dekhtyar and V. S. Subrahmanian. Hybrid probabilistic programs. In *Proceedings of the 14th International Conference on Logic Programming*, pages 391–405. MIT Press, 1997.

[5] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Inf. Comput.*, 87:78–128, 1990.

[6] J. A. Fernández, J. Lobo, J. Minker, and V. S. Subrahmanian. Disjunctive LP + integrity constraints = stable model semantics. *Ann. Math. Artif. Intell.*, 8(3–4):449–474, 1993.

[7] J. A. Fernández and J. Minker. Bottom-up computation of perfect models for disjunctive theories. *J. Logic Program.*, 25(1):33–51, 1995.

[8] H. Gaifman. Concerning measures in first order calculi. *Israel J. Math.*, 2:1–18, 1964.

[9] J. Y. Halpern. An analysis of first-order logics of probability. *Artif. Intell.*, 46:311–350, 1990.

[10] M. Kifer and V. S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *J. Logic Program.*, 12(3–4):335–367, 1992.

[11] J. W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, 2nd ed., 1987.

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

[13] T. Lukasiewicz. Many-valued disjunctive logic programs with probabilistic semantics. In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, volume 1730 of *LNAI*, pages 277–289. Springer, 1999.

[14] T. Lukasiewicz. Many-valued first-order logics with probabilistic semantics. In *Proceedings of the Annual Conference of the European Association for Computer Science Logic,* 1998, volume 1584 of *LNCS*, pages 415–429. Springer, 1999.

[15] T. Lukasiewicz. Probabilistic and truth-functional many-valued logic programming. In *Proceedings of the 29th IEEE International Symposium on Multiple-Valued Logic*, pages 236–241. IEEE Computer Society, 1999.

[16] T. Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Trans. Computat. Logic*, 2(3):289–337, 2001.

[17] C. Mateis. *A Quantitative Extension of Disjunctive Logic Programming*. Doctoral Dissertation, Technische Universität Wien, 1998.

[18] C. Mateis. Extending disjunctive logic programming by t-norms. In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, volume 1730 of *LNAI*, pages 290–304. Springer, 1999.

[19] J. Minker. Overview of disjunctive logic programming. *Ann. Math. Artif. Intell.*, 12:1–24, 1994.

[20] J. Minker and A. Rajasekar. A fixpoint semantics for disjunctive logic programs. *J. Logic Program.*, 9(1):45–74, 1990.

[21] R. T. Ng. Semantics, consistency, and query processing of empirical deductive databases. *IEEE Trans. Knowl. Data Eng.*, 9(1):32–49, 1997.

[22] R. T. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Inf. Comput.*, 101:150–201, 1992.

[23] R. T. Ng and V. S. Subrahmanian. Stable semantics for probabilistic deductive databases. *Inf. Comput.*, 110:42–83, 1994.

[24] L. Ngo. Probabilistic disjunctive logic programming. In *Proc. of the 12th Conf. on Uncertainty in Artificial Intelligence*, pages 397–404. Morgan Kaufmann, 1996.

[25] N. J. Nilsson. Probabilistic logic. *Artif. Intell.*, 28:71–88, 1986.

[26] T. C. Przymusinski. On the declarative semantics of stratified deductive databases and logic programs. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 193–216. Morgan Kaufmann, 1988.

[27] T. C. Przymusinski. Stable semantics for disjunctive programs. *New Generation Comput.*, 9:401–424, 1991.

[28] C. Ruiz and J. Minker. Computing stable and partial stable models of extended disjunctive logic programs. In *Nonmonotonic Extensions of Logic Programming*, volume 927 of *LNCS*, pages 205–229. Springer, 1995.

[29] D. Seipel, J. Minker, and C. Ruiz. Model generation and state generation for disjunctive logic programs. *J. Logic Program.*, 32(1):49–69, 1997.

[30] M. H. van Emden. Quantitative deduction and its fixpoint theory. *J. Logic Program.*, 3(1):37–53, 1986.