Non–Monotonic Reasoning Based on Minimal Models and its Efficient Implementation

Dietmar Seipel

University of Tübingen
Sand 13, D – 72076 Tübingen, Germany
seipel@informatik.uni-tuebingen.de

Abstract. A common approach to *non–monotonic reasoning* is to choose a set $\mathcal{S}$ of models (e.g. minimal, perfect or stable models) for a logic program P. Then, the meaning of P is defined as the set of all formulas which are implied logically by *all* models in $\mathcal{S}$.
For a disjunctive logic program P, we consider the set $\mathcal{S} = \mathcal{MM}_P$ of minimal models and get the non–montonic consequences $\mathcal{NM}_P$ of ground disjunctions over positive or negative literals. The set $\mathcal{NM}_P$ consists of three subsets: The minimal model state $\mathcal{MS}_P$ (ground disjunctions over positive literals), the negated extended generalized closed world assumption $\neg\mathcal{EGCWA}_P$ (ground disjunctions over negative literals), and the rest (mixed disjunctions over positive and negative literals). We show that the mixed disjunctions are implied logically by $\mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P$. Thus the interesting sets are the logical consequences $\mathcal{MS}_P$ and the non–monotonic consequences $\neg\mathcal{EGCWA}_P$. As described in [16], $\mathcal{MS}_P$ consists of the positive ground disjunctions, which are *true* in all minimal models, and $\mathcal{EGCWA}_P$ consists of the positive ground conjunctions, which are *false* in all minimal models.
We will present a compact data structure, called *clause tree*, for disjunctive Herbrand states (ground disjunctions over positive literals). It is used for deriving $\mathcal{MS}_P$ as the least fixpoint of the disjunctive consequence operator $\mathcal{T}_P^S$ on disjunctive Herbrand states, and, on the other hand, for deriving $\mathcal{EGCWA}_P$ based on the support–for–negation sets, which are certain disjunctive Herbrand states.

1 Introduction

Deductive databases use logic programming and techniques from non–monotonic reasoning for building a powerful, declarative language for specifying data and queries in relational databases. There has been a lot of research on the processing of definite Horn clauses with definite facts, cf. [3], [22], [4]. Recursive programs are evaluated iteratively in a set–oriented fashion based on the primitive operations of relational algebra, which can be executed efficiently on databases. The desired result is the least fixpoint of the definite consequence operator $\mathcal{T}_P$, which equals the minimal Herbrand model $\mathcal{M}_P$, cf. [15]. More advanced applications require the support of disjunctive rules and disjunctive facts: The deduction of independencies in probabilistic reasoning, cf. [17], [21], and the overlapping of rectangles in spatial databases, cf. [1], are examples falling into this category.

Various semantics for disjunctive normal logic programs are given in the literature: well–founded semantics [23], generalized well–founded semantics and generalized disjunctive well–founded semantics [16], stationary semantics [18] and stable model semantics [12], [11]. A classification of these semantics w.r.t. certain criteria is given by [8].

A disjunctive logic program consists of rules with positive disjunctive rule heads and positive atoms in the rule bodies. Its semantics is determined by the set $\mathcal{MM}_P$ of its *minimal models*. The disjunctive consequence operator $\mathcal{T}_P^S$ allows for the derivation of the set $\mathcal{MS}_P$ of all positive disjunctive clauses logically implied by the program P, the *minimal model state*. On the other hand, disjunctive clauses over negated atoms are derived by the *extended generalized closed world assumption* $\mathcal{EGCWA}_P$. Such a clause is the negation of a conjunction of positive atoms. $\mathcal{EGCWA}_P$ contains all conjunctions which are *false* in all minimal Herbrand models of the program. In contrast to the logical implication of the *truth* of the facts in $\mathcal{MS}_P$, the closed world assumptions are non–logical inference rules: the *falsity* of the conjunctions in $\mathcal{EGCWA}_P$ is not implied logically by P. Whereas logical implication is monotonic ($\mathcal{MS}_P$ grows with P), closed world assumptions form one of the major formalisms of *non–monotonic reasoning* in artificial intelligence ($\mathcal{EGCWA}_P$ can shrink or grow with growing P).

The rest of this paper is organized as follows: We will review some of the definitions and theoretical foundations for disjunctive logic programming and their fixpoint semantics given in [16]. Then we introduce the concept of non–monotonic consequences $\mathcal{NM}_P$. We present efficient bottom–up Δ–iteration techniques for computing $\mathcal{MS}_P$ and $\mathcal{EGCWA}_P$ based on an iterative computation of certain resolvents of disjunctive Herbrand states with ground conjunctions. The computation of these resolvents can be supported by a compact data structure for disjunctive Herbrand states, the *clause tree*.

2 Foundations of Disjunctive Logic Programming

A *disjunctive deductive database DDDB* or *disjunctive logic program* is given by a set of facts, which resemble a relational database and a set of rules for deriving new facts. A disjunctive logic *rule* r is given by

$$A_1 \vee \ldots \vee A_k \;\text{:-}\; B_1, \ldots, B_m$$

where the disjunction $A_1 \vee \ldots \vee A_k$ of atoms A_i ($1 \leq i \leq k$, $k \in \mathbb{N}_0$) is called its head and the conjunction $B_1 \wedge \ldots \wedge B_m$ of atoms B_i ($1 \leq i \leq m$, $m \in \mathbb{N}_0$) its body. Iff $m = 0$, r is a disjunctive *fact*, also denoted by $A_1 \vee \ldots \vee A_k$. The disjunctive logic program containing all the ground instances of rules in P is denoted by $Ground(P)$.

The disjunctive (conjunctive) *Herbrand base* DHB_P (CHB_P) of P is a set of positive ground disjunctions $A_1 \vee \ldots \vee A_k$ (conjunctions $A_1 \wedge \ldots \wedge A_k$) of atoms

$A_i \in HB_P$ ($1 \le i \le k$, $k \in \mathbb{N}_0$) from the Herbrand base HB_P. A disjunctive (conjunctive) *Herbrand state* of P is a subset $T \subseteq DHB_P$ ($F \subseteq CHB_P$).

The *canonical form* and the *expansion* of a disjunctive (conjunctive) Herbrand state $S \subseteq DHB_P$ ($S \subseteq CHB_P$) are

$$can(S) = \{\ C \in S \mid \text{there is no proper subclause } C' \text{ of } C \text{ in } S\ \},$$
$$exp(S) = \{\ C \in H \mid \text{there is a subclause } C' \text{ of } C \text{ in } S\ \},$$

where subclause means disjunction (conjunction) and $H = DHB_P$ ($H = CHB_P$). That is, $can(S) \subseteq S \subseteq exp(S)$. S is called expanded, iff $exp(S) = S$.

The *set difference with subsumption* $T_1 \setminus_s T_2$ and the *set disjunction* $T_1 \vee T_2$, respectively, of two disjunctive Herbrand states T_1 and T_2 are

$$T_1 \setminus_s T_2 = \{\ C \in T_1 \mid \text{there is no subclause } C' \text{ of } C \text{ in } T_2\ \},$$
$$T_1 \vee T_2 = \{\ C_1 \vee C_2 \mid C_1 \in T_1,\ C_2 \in T_2\ \}.$$

The *negation* of a conjunctive Herbrand state F is

$$\neg F = \{\ \neg A_1 \vee \ldots \vee \neg A_k \mid A_1 \wedge \ldots \wedge A_k \in F\ \}.$$

Example 1. For the Herbrand base $HB_P = \{a, b, c\}$ and the disjunctive Herbrand states $T_1 = \{a \vee b, a \vee c\}$ and $T_2 = \{a \vee b, b \vee c\}$ we get $T_3 = T_1 \vee T_2 = \{a \vee b, a \vee b \vee c\}$, $T_1 \setminus_s T_3 = \{a \vee c\}$, and for $T_4 = \{a \vee b\}$ we get $exp(T_4) = T_3$ and $can(T_3) = T_4$.

2.1 The Disjunctive Consequence Operator

The set of ground disjunctions logically implied by a disjunctive logic program P can be derived by bottom–up or naive evaluation algorithms as the least fixpoint of the disjunctive consequence operator $\mathcal{T}_P^S$, cf. [16].

Definition 1. Given a disjunctive logic program P.
(i) The disjunctive *consequence operator* $\mathcal{T}_P^S : 2^{DHB_P} \to 2^{DHB_P}$ of P derives

$$\begin{aligned} \mathcal{T}_P^S(D)\ &= \{\ C' \vee C_1 \vee \ldots \vee C_n \mid C' \text{ :- } B_1, \ldots, B_n \text{ is in } Ground(P), \\ &\qquad B_1 \vee C_1, \ldots, B_n \vee C_n \text{ are in } D\ \}, \text{ for } D \subseteq DHB_P. \end{aligned}$$

(ii) The *powers* of $\mathcal{T}_P^S$ are disjunctive Herbrand states $\mathcal{T}_P^S \uparrow n \subseteq DHB_P$:

$$\mathcal{T}_P^S \uparrow 0 = \emptyset,\ \ \mathcal{T}_P^S \uparrow n{+}1 = \mathcal{T}_P^S(\mathcal{T}_P^S \uparrow n), \text{ for } n \in \mathbb{N}_0,\ \ \mathcal{T}_P^S \uparrow \omega = \cup_{n<\omega} \mathcal{T}_P^S \uparrow n.$$

Example 2. A very frequently used logic program P_{tc} is the one that specifies the transitive closure of a graph given by a disjunctive *arc*–relation $D = \{\ arc(a, b) \vee arc(a, c),\ arc(b, d),\ arc(c, d)\ \}$:

$$P_{tc}\ =\ \{\ path(X, Y) \text{ :- } path(X, Z), path(Z, Y);\ \ path(X, Y) \text{ :- } arc(X, Y)\ \}.$$

For $P = P_{tc} \cup D$ we get $\mathcal{T}_P^S \uparrow n = D_1 \cup \ldots \cup D_n$, where $D_1 = D$ and

$$D_2 = \{\ arc(a, b) \vee path(a, c),\ path(a, b) \vee arc(a, c),\ path(b, d),\ path(c, d)\ \},$$
$$D_3 = \{\ path(a, b) \vee path(a, c),\ path(a, d) \vee arc(a, c),\ path(a, d) \vee arc(a, b)\ \},$$
$$D_4 = \{\ path(a, d) \vee path(a, c),\ path(a, d) \vee path(a, b)\ \}.$$

Finally, $D_5 = \{\ path(a, d)\ \}$, $D_6 = \emptyset$, and $\mathcal{T}_P^S \uparrow \omega = \mathcal{T}_P^S \uparrow 5$. Thus, $\mathcal{T}_P^S$ can also derive definite facts from disjunctive facts.

2.2 The Minimal Model State

For every definite logic program P the minimal Herbrand model $\mathcal{M}_P$ of P is equal to the logical consequences $A \in HB_P$ of P and the least fixpoint $lfp(\mathcal{T}_P) = \mathcal{T}_P \uparrow \omega$ of the definite consequence operator $\mathcal{T}_P$, cf. [15].

In general, a disjunctive logic program P does not have a unique minimal Herbrand model, but a set $\mathcal{MM}_P$ of minimal Herbrand models. For disjunctive logic programs, model states correspond to models for logic programs.

Definition 2. Given a disjunctive logic program P.

(i) A *model state* D of P is an expanded disjunctive Herbrand state of P, where $\mathcal{T}_P^S(D) \subseteq D$.

(ii) The *minimal model state* $\mathcal{MS}_P$ of P is given by the intersection $\bigcap_{D \in MS_P} D$ of the set MS_P of all model states of P.

As for Herbrand models of definite logic programs, every intersection of model states again is a model state. Thus, $\mathcal{MS}_P$ is the unique minimal element in the set of all model states of P. Model states can also be characterized based on Herbrand models: an expanded disjunctive Herbrand state D of P is a model state of P, iff every minimal Herbrand model of D is a Herbrand model of P. According to [16], $\mathcal{MS}_P$ is equivalent to the set of logical consequences $C \in DHB_P$ of P, the expansion of the least fixpoint $lfp(\mathcal{T}_P^S) = \mathcal{T}_P^S \uparrow \omega$, and the set

$$\{ \; C \in DHB_P \mid \forall M \in \mathcal{MM}_P : M \models C \; \}$$

of all disjunctions $C \in DHB_P$ logically implied by all minimal Herbrand models $M \in \mathcal{MM}_P$ of P. For a definite logic program P, $\mathcal{M}_P = can(\mathcal{MS}_P)$.

3 Non–Monotonic Reasoning Based on Minimal Models

Closed world assumptions are one of the major formalisms of non–monotonic reasoning in artificial intelligence. The *extended generalized closed world assumption* $\mathcal{EGCWA}_P$ of a disjunctive logic program P is a conjunctive Herbrand state containing all ground conjunctions which are *false* in all $M \in \mathcal{MM}_P$.

Definition 3. Given a disjunctive logic program P. Then

$$\mathcal{EGCWA}_P \; = \; \{ \; E \in CHB_P \mid \forall M \in \mathcal{MM}_P : M \models \neg E \; \}.$$

Example 3. The disjunctive logic program $P \; = \; \{ \; c \text{ :- } a, \quad a \vee b \; \}$ derives the minimal model state $\mathcal{MS}_P = \{a \vee b, b \vee c, a \vee b \vee c\}$ and has the set $\mathcal{MM}_P = \{\{a, c\}, \{b\}\}$ of minimal Herbrand models. Thus, $\mathcal{EGCWA}_P = \{a \wedge b, b \wedge c, a \wedge b \wedge c\}$. But, the stronger disjunctive logic program $P' = P \cup \{c \vee a\}$ derives the weaker set $\mathcal{EGCWA}_{P'} = \{a \wedge b \wedge c\}$ of *false* conjunctions (*Non–Monotonicity*).

$\neg\mathcal{EGCWA}_P$ is maximal consistent with the minimal model state $\mathcal{MS}_P$: it contains exactly the ground disjunctions $\neg A_1 \vee \ldots \vee \neg A_k$ which can be assumed to be *true* without making it possible to derive new positive ground disjunctions from $\mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P$.

So far, based on the set $\mathcal{MM}_P$ of minimal models of a disjunctive logic program P, we can derive the set $\mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P$ of ground disjunctions C, which are either purely positive ($C = A_1 \vee \ldots \vee A_k \in \mathcal{MS}_P$) or purely negative ($C = \neg A_1 \vee \ldots \vee \neg A_k \in \neg\mathcal{EGCWA}_P$). The concept of non–montonic reasoning based on minimal models can easily be extended to derive the set $\mathcal{NM}_P$ of all ground disjunctions $C = L_1 \vee \ldots \vee L_k$ over positive and negative literals, which are logically implied by all minimal Herbrand models of P.

Definition 4. Given a disjunctive logic program P.

(i) The *extended disjunctive Herbrand base* $EDHB_P$ of P is

$$EDHB_P \;=\; \{\, L_1 \vee \ldots \vee L_k \mid k \in \mathbb{N}_0,\ (L_i = A_i \text{ or } L_i = \neg A_i),\ A_i \in HB_P,\ 1 \le i \le k \,\}.$$

(ii) The set of *non–monotonic consequences* of P is

$$\mathcal{NM}_P \;=\; \{\, C \in EDHB_P \mid \forall M \in \mathcal{MM}_P : M \models C \,\}.$$

Obviously, $(\mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P) \subseteq \mathcal{NM}_P$. We will even show the logical equivalence $(\mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P) \iff \mathcal{NM}_P$. This will be done by considering logical formulas which are equivalent to the clause sets, cf. Example 4. We will use the following notations for a set $M \subseteq HB_P$ of atoms:

$$p_M \;=\; \bigwedge_{A \in M} A, \qquad n_M \;=\; \bigwedge_{A \in HB_P \setminus M} \neg A.$$

The conjunctive normal form $CNF_P = \bigwedge_{C \in can(\mathcal{MS}_P)} C$ of P is the conjunction of all minimal disjunctions of $\mathcal{MS}_P$. The disjunctive normal form DNF_P and the *completed disjunctive normal form* $CDNF_P$ of P are defined using $\mathcal{MM}_P$:

$$DNF_P \;=\; \bigvee_{M \in \mathcal{MM}_P} p_M, \qquad CDNF_P \;=\; \bigvee_{M \in \mathcal{MM}_P} p_M \wedge n_M.$$

CNF_P and DNF_P both are logically equivalent to $\mathcal{MS}_P$. $CDNF_P$ is stronger, and it can be verified that it is equivalent to the non–monotonic consequences $\mathcal{NM}_P$ of P, i.e. $\mathcal{NM}_P \iff CDNF_P$. Finally, the *closed world form* is

$$CWF_P \;=\; \bigvee_{M \in \mathcal{MM}_P} n_M.$$

It can be shown that CWF_P is the disjunctive equivalent of the conjunction $\bigwedge_{E \in can(\mathcal{EGCWA}_P)} \neg E$. Thus $DNF_P \wedge CWF_P \iff \mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P$.

Example 4. For the minimal model state $\mathcal{MS}_P = exp(\{a \vee c, b \vee c, b \vee d, c \vee d\})$ we get $\mathcal{MM}_P = \{\{a,b,d\},\{b,c\},\{c,d\}\}$, $\mathcal{EGCWA}_P = exp(\{a \wedge c, b \wedge c \wedge d\})$ and

$$\begin{aligned}
CNF_P &= (a \vee c) \wedge (b \vee c) \wedge (b \vee d) \wedge (c \vee d),\\
DNF_P &= (a \wedge b \wedge d) \vee (b \wedge c) \vee (c \wedge d),\\
CWF_P &= (\neg c) \vee (\neg a \wedge \neg d) \vee (\neg a \wedge \neg b),\\
CDNF_P &= (a \wedge b \wedge d \wedge \neg c) \vee (b \wedge c \wedge \neg a \wedge \neg d) \vee (c \wedge d \wedge \neg a \wedge \neg b).
\end{aligned}$$

Since $p_{M_1} \wedge n_{M_2}$ is *false* for $M_1 \neq M_2$ ($M_1, M_2 \in \mathcal{MM}_P$), we get by distributivity

$$DNF_P \wedge CWF_P \;=\; (\bigvee_{M_1 \in \mathcal{MM}_P} p_{M_1}) \wedge (\bigvee_{M_2 \in \mathcal{MM}_P} n_{M_2}) \iff$$

$$\iff \bigvee_{M_1, M_2 \in \mathcal{MM}_P} p_{M_1} \wedge n_{M_2} \iff \bigvee_{M \in \mathcal{MM}_P} p_M \wedge n_M \;=\; CDNF_P.$$

Chaining the above equivalences yields:

Theorem 5. *For a disjunctive logic program P: $\mathcal{NM}_P \iff \mathcal{MS}_P \cup \neg\mathcal{EGCWA}_P$.*

4 Reasoning with Disjunctive Herbrand States

The basic operators for deriving $\mathcal{MS}_P$ and $\mathcal{EGCWA}_P$ of a disjunctive logic program P are the state resolvent Res^S and the state reduction Red^S of a disjunctive Herbrand state D w.r.t. an atom A. They partition D into disjoint sets D_1 and D_2: $D_1 = \{A\} \vee Res^S(D, A)$ consists of all disjunctions containing A and $D_2 = Red^S(D, A)$ consists of all disjunctions lacking A.

When reasoning with the facts and rules of the database, another data structure for the factual information seems to be appropriate: the clause tree. Every path from the root of the clause tree to a leaf represents a disjunctive clause. This tree structure can also be seen as another normal form for boolean formulas over $\wedge$ and $\vee$. Whereas the *conjunctive normal form CNF* (conjunction of disjunctive clauses) and the *disjunctive normal form DNF* (disjunction of conjunctions) represent formulas in a two level hierarchy using $\wedge$ and $\vee$, formulas in *tree normal form TNF* are built recursively as the disjunction of a disjunctive clause with a conjunction of formulas in *TNF*.

The trees support the efficient derivation of state resolvents and reductions.

4.1 Resolvents and Reductions of Disjunctive Herbrand States

Definition 6. Given a disjunctive logic program P, a disjunctive Herbrand state $D \subseteq DHB_P$, an atom $A \in HB_P$, and a conjunction $E = A_1 \wedge \ldots \wedge A_k \in CHB_P$.
(i) The *state resolvent* $Res^S(D, A)$ and the *state reduction* $Red^S(D, A)$ are

$$Res^S(D, A) = \{\, C \in DHB_P \mid C \vee A \in D \,\},$$
$$Red^S(D, A) = \{\, C \in DHB_P \mid C \in D,\ A \text{ does not occur in } C \,\}.$$

(ii) The *state resolvent* $Res^S(D, E)$ and the *state reduction* $Red^S(D, E)$ are

$$Res^S(D, E) = \vee_{i=1}^{k} Res^S(D, A_i), \quad Red^S(D, E) = \cap_{i=1}^{k} Red^S(D, A_i).$$

Example 5. For the disjunctive Herbrand state $D = \{a \vee b \vee d, a \vee b \vee e, a \vee c\}$, we get $Res^S(D, b) = \{a \vee d, a \vee e\}$ and $Red^S(D, b) = \{a \vee c\}$.

The minimal model state $\mathcal{MS}_P$ of a disjunctive logic program P can be derived by repeated computations of resolvents of derived disjunctive Herbrand states with the ground conjunctions formed by the bodies of the ground rules, since

$$\mathcal{T}_P^S(D) \;=\; \bigcup_{C :- E \,\in\, Ground(P)} \{C\} \vee Res^S(D, E).$$

The extended generalized closed world assumption $\mathcal{EGCWA}_P$ can be derived from the resolvents $Res^S(\mathcal{MS}_P, E)$, which are called support–for–negation sets in [16], of the minimal model state $\mathcal{MS}_P$ with the conjunctions $E \in CHB_P$:

$$\mathcal{EGCWA}_P \; = \; \{ \; E \in CHB_P \mid Res^S(\mathcal{MS}_P, E) \setminus_s \mathcal{MS}_P = \emptyset \; \}.$$

4.2 Representing Herbrand States by Clause Trees

The computation of minimal model states and extended generalized closed world assumptions can be efficiently supported by a data structure for disjunctive facts, the so–called *clause tree*, cf. [19]. Another data structure, the *model tree* proposed by [16] and [10], represents the set of all minimal models of a disjunctive Herbrand state S instead of the clauses of S. Both types of trees are used for a fast computation of state resolvents and state reductions, and for purposes of subsumption elimination. W.r.t. the following general definition of set trees, a clause tree is a set tree for a disjunctive Herbrand state S and a model tree is a set tree for a set S of models. Set trees can be derived by insertion heuristics or by Steiner minimal tree heuristics, cf. [24], [19].

Definition 7 (Set Tree). Given a set $S = \{S_1, \ldots, S_n\}$ ($n \in \mathbb{N}_0$) of some finite sets S_i ($1 \leq i \leq n$). Let $\mathcal{A}_S = \cup_{i=1}^n S_i$. A *set tree* $\mathcal{T} = (V, E)$ with a node labeling $l : V \rightarrow 2^{\mathcal{A}_S}$ *represents* S, iff

(i) each leaf $v \in V$ represents a set $S(v)$ as follows: $S(v)$ is the union of the sets $l(v')$ which label the nodes v' on the path from the root to v,

(ii) the sets $S(v)$ of the leaves $v \in V$ are in S, and every set $S_i \in S$ is represented by exactly one leaf v with $S(v) = S_i$ of the set tree.

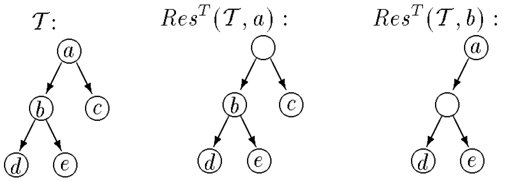

Fig. 1. A Clause Tree $\mathcal{T}$ and Tree–Resolvents

Example 6. The disjunctive Herbrand state $D = \{a \vee b \vee d, a \vee b \vee e, a \vee c\}$ is represented by the clause tree $\mathcal{T}$ of Figure 1. $Res^T(\mathcal{T}, a)$ and $Res^T(\mathcal{T}, b)$ represent the states $Res^S(D, a) = \{b \vee d, b \vee e, c\}$ and $Res^S(D, b) = \{a \vee d, a \vee e\}$, respectively.

5 Summary

We have developed a system called DisLog for the efficient evaluation of disjunctive normal logic programs under the generalized disjunctive well–founded semantics $\mathcal{GDWFS}_P$. It is implemented as a meta–interpreter in Sicstus–Prolog.

Computing the semantics $\mathcal{GDWFS}_P$ for a disjunctive, normal logic program P requires the repeated derivation of the minimal model state and the extended generalized closed world assumption w.r.t. certain disjunctive transformations of P. The basic operators for computing minimal model states are the disjunctive consequence operator $\mathcal{T}_P^S$, the Δ–operator $\setminus_s$ with subsumption and the canonization operator *can* for disjunctive Herbrand states. Tree–based versions of these operators are applied in a Δ–iteration for computing $\mathcal{MS}_P$ and $\mathcal{EGCWA}_P$.

References

[1] *A. Abdelmoty, M. Williams, N. Paton:* Deduction and Deductive Databases for Geographic Data Handling, Proc. Intl. Symposium on Advances in Spatial Databases 1993 (SSD'93), pp. 443-464.

[3] *F. Bancilhon, R. Ramakrishnan:* An Amateur's Introduction to Recursive Query Processing Strategies, Proc. ACM SIGMOD 1986, pp. 16-52.

[4] *S. Ceri, G. Gottlob, L. Tanca:* Logic Programming and Databases, Springer, 1990.

[6] *W. Chen, D.S. Warren:* Query Evaluation under the Well–Founded Semantics, Proc. ACM PODS 1993.

[8] *J. Dix:* Classifying Semantics of Disjunctive Logic Programs, Proc. Joint. Intl. Conf. and Symp. on Logic Programming 1992, pp. 798-812.

[9] *U. Fuhrbach:* Computing Answers for Disjunctive Logic Programs, Proc. ILPS Workshop on Disjunctive Logic Programs 1991.

[10] *J.A. Fernández, J. Minker:* Theory and Algorithms for Disjunctive Deductive Databases, Programmirovanie J. 1993, Academy of Sciences of Russia.

[11] *M. Fitting:* The Family of Stable Models, Journal of Logic Programming, vol. 17(3,4), 1993, pp. 197-225.

[12] *M. Gelfond, V. Lifschitz:* The Stable Model Semantics for Logic Programming, Proc. Intl. Conf. and Symp. on Logic Programming 1988, pp. 1070-1080.

[15] *J.W. Lloyd:* Foundations of Logic Programming, Springer, second edition, 1987.

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

[17] *J. Pearl:* Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufman, 1988.

[18] *T.C. Przymusinski*: Stationary Semantics for Disjunctive Logic Programs and Deductive Databases, Proc. North–American Conf. of Logic Prog. 1990, pp. 42-59.

[19] *D. Seipel*: Tree–Based Fixpoint Iteration for Disjunctive Logic Programs, Proc. Workshop on Logic Prog. with Incomplete Inf., Intl. Symp. on Logic Prog. 1993.

[20] *D. Seipel*: An Efficient Computation of the Extended Generalized Closed World Assumption by Support–for–Negation Sets, Proc. Intl. Conf. on Logic Programming and Automated Reasoning 1994 (LPAR'94).

[21] *D. Seipel, H. Thöne*: An Application of Disjunctive Logic Programming with Incomplete Information, Proc. Intl. Conf. on Expert Syst. for Development 1994.

[22] *J.D. Ullman*: Principles of Database and Knowledge-Base Systems, Volume I,II, Computer Science Press, 1988/89.

[23] *A. Van Gelder, K.A. Ross, J.S. Schlipf*: The Well–Founded Semantics for General Logic Programs, JACM, vol. 38(3), 1991, pp. 620-650.

[24] *S. Voß:* Steiner–Probleme in Graphen, Anton Hain, Frankfurt am Main, 1990.