In: Proceedings of the Twelth ACM SIGACT SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-93), 1993, pp. 158–167.

Complexity Aspects of Various Semantics for Disjunctive Databases

Thomas Eiter and Georg Gottlob

(eiter|gottlob)@vexpert.dbai.tuwien.ac.at

Christian Doppler Laboratory for Expert Systems
Information Systems Department, Vienna University of Technology
Paniglgasse 16, A-1040 Wien, Austria

Abstract

This paper addresses complexity issues for important problems arising with disjunctive databases. In particular, the complexity of inference of a literal and a formula from a propositional disjunctive database under a variety of well-known disjunctive database semantics is investigated, as well deciding whether a disjunctive database has as model under a particular semantics. The problems are located in appropriate slots of the polynomial hierarchy.

1 Introduction

Allowing to store disjunctions in a logical database is indispensable for dealing with disjunctive information. Accordingly, the meaning of a disjunctive database is expressed by a set of models instead of a single model as in case of nondisjunctive databases. A variety of different semantics for disjunctive databases has been proposed in the literature; see [9] for a comprehensive overview. We will deal with the following ones.

- The Generalized Closed World Assumption (GCWA) by Minker [16].
- The Extended Generalized Closed World Assumption (EGCWA) by Yahya and Henschen [30].
- The Careful Closed World Assumption (CCWA) by Gelfond and Przymusinska [11].
- The Disjunctive Database Rule (DDR) by Ross and Topor [23], which is equivalent to the Weak GCWA of Rajasekar, Lobo, and Minker [21].
- The Extended Closed World Assumption (ECWA) by Gelfond, Przymusinska, and Przymusinski [12], which is in the finite propositional case equivalent to McCarthy's circumscription as defined by Lifschitz in [14] (CIRC).
- The Possible Worlds Semantics (PWS) by Chan [5], which turned out equivalent to the independently developed Possible Models Semantics (PMS) of Sakama [24].
- The Perfect Models Semantics (PERF) by Przymusinski [19] and the Iterated Closed World Assumption (ICWA) by Gelfond, Przymusinska, and Przymusinski [12] introduced for capturing PERF under stratified negations.
- The (Partial) Disjunctive Stable Model Semantics (DSM, PDSM) as defined by Przymusinski [20] to extend Gelfond and Lifschitz's Stable Semantics [10] and the Well-Founded Semantics of van Gelder, Ross, and Schlipf [29].

Much work has been spent on studying the logical properties of these semantics, but less on complexity issues and tractability in the propositional case, cf. [4, 27]. Cadoli and Lenzerini present in [3] a careful complexity analysis of inference from a propositional database under several semantics from above (GCWA, EGCWA, CCWA, ECWA, CIRC) for syntactically restricted cases. Eiter and Gottlob complemented their work with results for arbitrary propositional databases [7]. Marek and Truszczyński [15] and Bidoit and Froidevaux [2] independently showed that deciding whether a nondisjunctive database has a stable model is NP-complete. Chan [5] considered the complexity of inference of a literal under GCWA, DDR and PWS. He proved that inferring a literal from a propositional database in which no integrity clauses (i.e. clauses with

ACM-PODS-5/93/Washington, D.C.

empty heads) occur is coNP-hard under GCWA, but polynomial under DDR and PWS. Chan proved that if integrity clauses are present, the problem becomes coNP-complete for DDR as well as PWS. In recent papers, Schaerf derived complexity results for non-Horn logic programs [25, 26]. In [25], using results in [7] he derived Σ_2^P- and Π_2^P-completeness results for positivistic models and minimally-supported models. In [26] he analyzed the complexity of weakly-stable and weakly-supported models of non-Horn programs, showing that in both cases deciding whether a formula is true in at least one such model is NP-complete and that deciding whether a formula is true in all such models is coNP-complete.

In this paper, we present complexity results for inference tasks under the above mentioned semantics for disjunctive databases. As in [5], we limit our analysis to propositional (i.e. grounded) databases. Some of the results concerning GCWA, EGCWA, CCWA, CCWA, and CIRC appear in weaker form in [7]; note that the results in the present paper constitute a substantial improvement as they are derived for databases without integrity clauses. The proofs given in [7] do not apply to the cases we consider here.

In particular, we consider the following inference problem: Given a finite disjunctive propositional database DB and a propositional formula F, is F satisfied by all disjunctive models of DB ? We pay special attention to the traditionally most important subcase where F is a single literal L. Besides the inference problem, we address the problem of deciding whether the database has a model. As [5], we consider both problems also for databases without integrity clauses.

The results of this paper, together with previous results and results by other authors (marked by *), are summarized in Tables 1 and 2. They may be interpreted as follows. If the polynomial hierarchy does not collapse into some class below Π_2^P, the inference problem is for almost all semantics strictly harder than classical inference, even in case of positive databases, i.e. databases without integrity clauses and negation "$\neg$". An intuitive explanation for Π_2^P-hardness is that due to some minimality criterion, the problem of identifying a disjunctive model is difficult and involves an (at least) coNP-hard test. This constitutes a source of complexity "orthogonal" to the source given by the potentially exponentially many candidates for a disjunctive model of DB. As a consequence of the results, under the assumption polynomial inference algorithms using an oracle for classical inference do not exist.

Inference of a formula under GCWA or CCWA is an interesting problem: the best upper bound we can provide by a nontrivial membership proof is $\mathrm{P}^{\Sigma_2^P[O(\log n)]}$, which is "mildly" harder than Π_2^P or Σ_2^P. Completeness for this class would entail Σ_2^P-hardness; however, it is not clear how to reduce a Σ_2^P-complete problem to these problems. Literal inference under GCWA is in Π_2^P, as it suffices to check a restricted set of DB models.

Chan [5] shows that inference under DDR and PWS (which exclude "$\neg$") is tractable if integrity clauses are not allowed; this in fact constitutes the only cases of tractability for one of the considered semantics. If integrity clauses are permitted, both semantics are "merely" coNP-complete and thus still "easier" than all others. Since PWS is more intuitive than DDR, the complexity point of view supports choosing PWS.

It is interesting to observe that for PERF, DSM, and PDSM semantics, deciding whether DB has a model is likely to be more complex than for the other semantics. For all of them this is equivalent to deciding if the database has a model under classical semantics, from which the respective entries in Tables 1 and 2 are easily verified. Σ_2^P-hardness of DSM and PDSM may be intuitively explained by an additional source of complexity in terms of a fixpoint condition on a model which interacts with an independent minimality criterion. PERF similarly imposes a minimality condition on models, which again leads to the lower bound of Σ_2^P-hardness.

The rest of the paper is organized as follows. Section 2 provides definitions and notation. All further sections are devoted to justify the remaining entries in the tables. Following the classification in [9], we deal in Sections 3–5 with semantics for disjunctive deductive DBs, disjunctive stratified DBs, and disjunctive normal DBs. The various semantics can be alternatively characterized proof-theoretically, by fixpoints, or in terms of models, cf. [9]. We refer here only to model theoretic characterizations, which are heavily used for deriving the results. Full proofs of the results can be collected from [8], which extends on the material of this paper in the context of logic programming.

2 Preliminaries and Notation

We consider propositional databases over a finite set V of propositional variables. Denote by $\mathcal{C}$ the set of clauses

$$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_k \wedge \neg b_{k+1} \wedge \cdots \wedge \neg b_m, \quad n, m \geq 0,$$

where all a_i and b_j are from V. The set of clauses in which "$\neg$" does not occur is denoted by $\mathcal{C}^+$. A *disjunctive database* (DDB) DB is a finite set of clauses from $\mathcal{C}$. According to the classification of [9], any such DB is a *disjunctive normal database* (DNDB) in general; it is a *disjunctive deductive database* (DDDB) if $DB \subseteq \mathcal{C}^+$, and a *disjunctive stratified database* (DSDB)

Table 1: Complexity results for positive propositional DDBs (i.e. without integrity clauses and negation)

Semantics	Inference of literal	Inference of formula	$\exists$ model
GCWA	Π_2^P-complete	Π_2^P-hard, in $\mathrm{P}^{\mathbf{\Sigma}_2^P[O(\log n)]}$	
DDR ($\equiv$WGCA)	in P *[5]	coNP-complete	
PWS ($\equiv$PMS)	in P *[5]	coNP-complete	
EGCWA	Π_2^P-complete		$O(1)$
CCWA	Π_2^P-hard, in $\mathrm{P}^{\mathbf{\Sigma}_2^P[O(\log n)]}$		
ECWA ($\equiv$CIRC)	Π_2^P-complete		
ICWA	Π_2^P-complete		
PERF	Π_2^P-complete		
DSM, PDSM	Π_2^P-complete		

Table 2: Complexity results for propositional DDBs (with integrity clauses)

Semantics	Inference of literal	Inference of formula	$\exists$ model
GCWA	Π_2^P-complete	Π_2^P-hard, in $\mathrm{P}^{\mathbf{\Sigma}_2^P[O(\log n)]}$	
DDR ($\equiv$WGCA)	coNP-complete *[5]	coNP-complete	
PWS ($\equiv$PMS)	coNP-complete *[5]	coNP-complete	
EGCWA	Π_2^P-complete		NP-complete
CCWA	Π_2^P-hard, in $\mathrm{P}^{\mathbf{\Sigma}_2^P[O(\log n)]}$		
ECWA ($\equiv$CIRC)	Π_2^P-complete		
ICWA$^\diamond$	Π_2^P-complete		$O(1)$
PERF	Π_2^P-complete		Σ_2^P-complete
DSM, PDSM	Π_2^P-complete		Σ_2^P-complete

$^\diamond$ defined here without integrity clauses

if negation is allowed in a structured way. A clause with $n = 0$ is called an *integrity clause*. We refer to a DDB which is a DDDB and in which no integrity clauses occur as *positive* DDB.

For any semantics C, by $C(DB)$ we denote the set of models of DB under this semantics. $M(DB)$ denotes the set of all Herbrand models of DB (i.e. model M is the set of propositional variables true in M). The partial order $\leq$ on $M(DB)$ is defined by $M \leq M'$ iff $M \subseteq M'$, i.e., all variables true in M are also true in M', and for a partition $\langle P; Q; Z\rangle$ of V, the preorder $\leq_{P;Z}$ by $M \leq_{P;Z} M'$ iff $M \cap Q = M' \cap Q$ and $M \cap P \subseteq M' \cap P$. Note that $\leq_{P;Z}$ coincides with $\leq$ if $P = V$. Model $M \in M(DB)$ is minimal (resp. $\langle P; Z\rangle$-minimal) iff no $M' \in M(DB)$ satisfies $M' \leq M$ and $M \not\leq M'$, (resp. $M' \leq_{P;Z} M$ and $M \not\leq_{P;Z} M'$). The minimal (resp. $\langle P; Z\rangle$-minimal) models of DB are denoted by $MM(DB)$ (resp. $MM(DB; P; Z)$).

Example 2.1 *Consider the database*

$$DB = \quad \{a \vee b \leftarrow c, \quad b \leftarrow \neg a \wedge \neg c\},$$

where $V = \{a, b, c\}$. *Then,* $M(DB) = \{\{b\}, \{a\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. *Furthermore* $MM(DB) = \{\{a\}, \{b\}\}$, *and for the partition* $\langle\{a\}, \{b\}, \{c\}\rangle$ *of* V, *we have* $MM(P; \{a\}; \{c\}) = \{\{b\}, \{b, c\}, \{a\}, \{a, c\}\}$.

Refer to [13] for concepts and notation of complexity theory. Recall that the classes $\mathbf{\Sigma}_k^P$, $\mathbf{\Pi}_k^P$, of the polynomial hierarchy are defined by $\Sigma_0^P = \Pi_0^P = \mathrm{P}$, and for $k \geq 0$, $\Sigma_{k+1}^P = \mathrm{NP}^{\Sigma_k^P}$, $\Pi_{k+1}^P = \mathrm{co}\Sigma_{k+1}^P$. In particular, $\Sigma_2^P = \mathrm{NP}^{\mathrm{NP}}$ and $\Pi_2^P = \mathrm{coNP}^{\mathrm{NP}}$.

The class of decision problems that are polynomially solvable with $\leq f(n)$ calls to a Σ_k^P oracle is $\mathrm{P}^{\Sigma_k^P[f(n)]}$, where $f(n)$ is a function in the size n of the problem instance.

3 Disjunctive deductive databases

Prior to reviewing the different semantics defined for such databases, we present the following fundamental theorem on a lower complexity bound of inference for many DDB semantics.

Theorem 3.1 *Let DB be a positive DDB and let $w \in V$. Deciding if $MM(DB) \models \neg w$ is Π_2^P-hard.*

Proof. (Sketch) We show this by the following reduction of deciding the validity of a quantified Boolean formula $\Phi = \forall x_1 \cdots \forall x_n \exists y_1 \cdots \exists y_m E$, $n, m \geq 1$. We may assume that $E = C_1 \wedge \cdots \wedge C_r$ and each $C_i = L_{i,1} \vee L_{i,2} \vee L_{i,3}$ is a disjunction of literals $L_{i,j}$ over variables $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ as deciding if such a Φ is valid is still Π_2^P-hard. Let $v_1, \ldots, v_n$ and $z_1, \ldots, z_m, w$ be new propositional variables and define the following database DB:

$$
\begin{aligned}
DB \;=\; \{ & x_1 \vee v_1 \leftarrow, \; \ldots \;, x_n \vee v_n \leftarrow, \\
& y_1 \vee z_1 \leftarrow, \; \ldots \;, y_m \vee z_m \leftarrow, \\
& w \leftarrow y_1 \wedge z_1, \; \ldots \;, w \leftarrow y_m \wedge z_m, \\
& y_1 \leftarrow w, \; \ldots \;, y_m \leftarrow w, \\
& z_1 \leftarrow w, \; \ldots \;, z_m \leftarrow w, \\
& w \leftarrow \tau(L_{1,1}) \wedge \tau(L_{1,2}) \wedge \tau(L_{1,3}), \\
& \qquad \vdots \\
& w \leftarrow \tau(L_{r,1}) \wedge \tau(L_{r,2}) \wedge \tau(L_{r,3}) \}
\end{aligned}
$$

where τ maps literals from $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ to literals as follows:

$$
\tau(L) = \begin{cases} v_i & \text{if } L = x_i \\ z_j & \text{if } L = y_j \\ x_i & \text{if } L = \neg x_i \\ y_j & \text{if } L = \neg y_j \end{cases}
$$

Intuitively, v_i corresponds to $\neg x_i$ and z_j corresponds to $\neg y_j$.

It can be shown that Φ is valid iff $MM(T) \models \neg w$. □

3.1 GCWA and CCWA

Reiter's CWA [22] adds to DB each literal $\neg x$, $x \in V$, such that $M(DB) \not\models x$. This is not suitable for disjunctive databases since it enforces a unique model of the DB if the result is consistent. (It is interesting to note that deciding whether CWA(DB) is nonempty is coNP-hard and in $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$, but not in coD^P [18] ($\mathrm{coD}^P \supseteq \mathrm{NP} \cup \mathrm{coNP}$) unless the polynomial hierarchy collapses [7].)

Minker adapted the CWA for disjunctive databases by introducing the *Generalized CWA* (GCWA) in [16], which adds all literals $\neg x$ to DB such that atom x is false in all minimal models of DB. The respective models of DB can be characterized as follows.

$$
\begin{aligned}
\mathrm{GCWA}(DB) \;=\; & \{M \in M(DB) : \forall x \in V. \\
& \quad MM(DB) \models \neg x \Rightarrow M \models \neg x\}
\end{aligned}
$$

The *Careful Closed World Assumption* (CCWA) of Gelfond and Przymusinska [11] generalizes the GCWA as follows. For a partition $\langle P; Q; Z\rangle$ of V, each literal $\neg x$, $x \in P$, is added to DB such that $MM(DB; P; Z) \models \neg x$. Thus,

$$
\begin{aligned}
\mathrm{CCWA}_{P;Z}(DB) = \{ & M \in M(DB) : \forall x \in P. \\
& MM(DB; P; Z) \models \neg x \Rightarrow M \models \neg x\}
\end{aligned}
$$

Notice that if $P = V$, CCWA is identical to GCWA. It is immediate from the definition that GCWA(DB) $\models \neg x$ iff $MM(DB) \models \neg x$ for any $x \in V$. Thus from Theorem 3.1,

Theorem 3.2 *Inference of a literal from a positive DDDB DB without integrity clauses is Π_2^P-hard under GCWA as well as CCWA.*

For GCWA, this lower bound is also an upper bound.

Theorem 3.3 *Literal inference under GCWA is in Π_2^P.*

Proof. A guess for $M \in MM(DB)$ such that $M \not\models L$ for L a literal, can be verified in polynomial time with an NP oracle, since deciding $M_1 \leq M_2$ is clearly polynomial. □

To derive an Π_2^P upper bound for literal inference under CCWA in an analogous way fails since for deciding $CCWA_{P;Z}(DB) \models \neg x$, if $x \in Q \cup Z$, checking only $\langle P; Z\rangle$-minimal models is not correct in general. (However, for $x \in P$ this is possible). Formula inference can be easily done with $O(|P|)$ calls to a Σ_2^P oracle by first deciding $MM(DB) \models \neg x$ for each $x \in P$ using the oracle. This upper bound can be improved to a logarithmic number of oracle calls.

Theorem 3.4 *Formula inference under CCWA (resp. GCWA) can be done with $O(\log|P|)$ (resp. $O(\log|V|)$) calls to a Σ_2^P oracle; thus the problem is in $\mathrm{P}^{\Sigma_2^P[O(\log n)]}$.*

Proof. (Sketch) We outline a polynomial-time algorithm that makes only $O(\log|P|)$ calls to a Σ_2^P oracle for CCWA (cf. [7] for this method). Notice that GCWA coincides with CCWA for $Q = Z = \emptyset$, hence $P = V$.

The basic idea is to proceed in two steps. Given DB and F, first the number r of variables x from P such that $MM(DB; P; Z) \not\models \neg x$ is computed. This can be done in a binary search with $O(\log|P|)$ calls to the Σ_2^P oracle: $r \geq k$ iff there exist $M_1, \ldots M_k \in MM(DB; P; Z)$ such that $|\bigcup_{i=1}^k M_i| \geq k$; since deciding $M \in MM(DB; P; Z)$ is in coNP, deciding $r \geq k$ is in Σ_2^P.

If r is known, a guess for $M \in CCWA_{P;Z}(DB)$ such that $M \not\models F$ can be witnessed by a polynomial sized structure $\mathcal{S}$ constituting a "proof" of $M \in CCWA_{P;Z}(DB)$, which can be checked in polynomial time with an NP oracle. $\mathcal{S}$ is as follows:

$$\mathcal{S} = \langle \{x_1, \ldots, x_r\}, (M_1, \ldots, M_r), M \rangle$$

where $x_1, \ldots, x_r$ are pairwise distinct variables, $M_1, \ldots, M_r \in MM(DB; P; Z)$ satisfying $x_i \in M_i$ for $1 \leq i \leq r$, and $M \in M(DB)$ such that $M \subseteq \{x_1, \ldots, x_r\}$ and $M \not\models F$.

Thus the second step is to query the Σ_2^P oracle whether such an M exists. □

3.2 DDR, WGCWA, PWS, and PMS

Given DB and a Herbrand interpretation I of DB, define

$$\begin{aligned} T_{DB}(I) &= \{a_i : a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_k \\ &\qquad \text{and} b_j \in I, \text{ for all } j = 1, \ldots k\}, \end{aligned}$$

and

$$\begin{aligned} T_{DB}{\uparrow}\, 0 &= \emptyset, \\ T_{DB}{\uparrow}\, n+1 &= T_{DB}(T_{DB}{\uparrow}\, n), \quad \text{and} \\ T_{DB}{\uparrow}\, \omega &= \bigcup_{n=0}^{\infty} T_{DB}{\uparrow}\, n. \end{aligned}$$

The *Disjunctive Database Rule* (DDR) of Ross and Topor [23] can be characterized as follows.

$$\begin{aligned} \mathrm{DDR}(DB) &= \{M \in M(DB) : \\ &\qquad x \in V - T_{DB}{\uparrow}\, \omega \Rightarrow M \models \neg x\}, \end{aligned}$$

i.e. the DDR adds to DB all literals $\neg x$ where atom x does not occur in $T \uparrow \omega$. The DDR semantics is equivalent to the Weak GCWA of Rajasekar, Lobo, and Minker [21]. Since $T_{DB}{\uparrow}\, \omega$ can be computed efficiently, the complexity results on formula inference in Tables 1 and 2 are obvious; the entries on literal inference were proven by Chan [5].

Chan improved the DDR semantics by taking care of integrity clauses in DB, which are not respected by DDR.

Example 3.1 *Let*

$$DB = \{a \vee b \leftarrow, \quad \leftarrow a \wedge b, \quad c \leftarrow a \wedge b\}.$$

Then DDR$(DB) \not\models \neg c$. *Under Chan's* Possible Worlds Semantics *(PWS), however,* PWS$(DB) \models \neg c$ *as suggested by intuition.*

Informally, PWS augments DB by all literals $\neg x$ such that x is false in all possible worlds of DB. We use Chan's characterization of PWS in terms of the equivalent *Possible Models Semantics* of Sakama [24]. A split database of DB is any database obtained if every clause

$$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_m, \quad n \geq 2,$$

in DB is replaced by Horn clauses

$$a_p \leftarrow b_1 \wedge \cdots \wedge b_m, \quad \text{for } p \in N,$$

for some nonempty $N \subseteq \{1, \ldots, n\}$. Let $\mathcal{SDB}(DB)$ be the family of all split databases of DB. Then,

$$\mathrm{PWS}(DB) = \{M \in MM(DB') : DB' \in \mathcal{SDB}(DB)\}.$$

Theorem 3.5 *Formula inference for PWS and PMS is* coNP*-complete.*

Proof. (Sketch) Guess $M \in \mathrm{PWS}(DB)$ and $DB' \in \mathcal{SDB}(DB)$ to verify M; as DB' is Horn, it has in case of consistency a unique minimal model M', which is efficiently computable. If $M' = M$ and $M \not\models F$, then PWS$(DB) \not\models F$. [1] Hardness follows from Chan's result on literal inference [5]. □

3.3 EGCWA, ECWA, and CIRC

The *Extended GCWA* (EGCWA) was introduced by Yahya and Henschen [30] for inferring integrity clauses. DB is augmented by each integrity clause $\leftarrow a_1 \wedge \cdots \wedge a_n$ that is true in every minimal model of DB. It holds that

$$\mathrm{EGCWA}(DB) = MM(DB).$$

The EGCWA is generalized by the *Extended CWA* (ECWA) of Gelfond, Przymusinska, and Przymusinski [12] in the following way. For a partition $\langle P; Q; Z \rangle$ of V,

$$\mathrm{ECWA}_{P;Z}(DB) = MM(DB; P; Z).$$

[1] Chan follows a different proof.

EGCWA results if $Q = Z = \emptyset$. ECWA is equivalent to *circumscription* (CIRC) as defined by Lifschitz in [14]. For $\langle P; Q; Z \rangle$, let

$$\begin{aligned} Circ(DB; P; Z) &= DB[P; Z] \wedge \\ &\quad \neg \exists P' Z' (DB[P'; Z'] \wedge (P' < P)) \end{aligned}$$

(cf. [14] for details). Then

$$\text{CIRC}_{P;Z}(DB) = M(Circ(DB; P; Z)).$$

It holds that $\text{CIRC}_{P;Z}(DB) = MM(DB; P; Z)$ [14], hence $\text{CIRC}_{P;Z}(DB) = \text{ECWA}_{P;Z}(DB)$. Notice that this holds as we have propositional databases, cf. [12].

Theorem 3.6 *Literal inference from a positive DDB DB is Π_2^P-hard under EGCWA, ECWA, and CIRC.*

Proof. Immediate from Theorem 3.1 □

Theorem 3.7 *Formula inference under EGCWA, ECWA, and CIRC is in Π_2^P.*

Proof. (Sketch) Clearly, $M_1 \leq_{P;Z} M_2$ is efficiently decidable for any models M_1 and M_2. Thus a guess for $M \in MM(DB; P; Z)$ such that $M \not\models F$ can be verified in polynomial time with an NP oracle. □

4 Disjunctive stratified databases

The concept of stratification of DNDBs, which has been discussed by Chandra and Harel [6], was independently introduced for logic programs by Apt, Blair, and Walker [1] and van Gelder [28]; Przymusinski generalized it to DDB [19]. A DNDB DB without integrity clauses is *stratified* iff it is possible to partition V into strata $\langle S_1, \ldots, S_r \rangle$, such that for every

$$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_k \wedge \neg c_1 \wedge \cdots \wedge \neg c_m$$

in DB there exists a constant c, $1 \leq c \leq r$, such that $\forall i$ Stratum$(a_i) = c$, $\forall j$ Stratum$(b_j) \leq c$, and $\forall l$ Stratum$(c_l) < c$, where Stratum$(x) = i$ iff $x \in S_i$. Any such $\langle S_1, \ldots, S_r \rangle$ is a *stratification* of DB.[2] Notice that a stratification of DB can be efficiently found.

Gelfond, Przymusinska, and Przymusinski [12] defined the *Iterated CWA* (ICWA) as iterated application of ECWA to a DSDB. Let $\langle P; Q; Z \rangle$ be a partition of V and $S = \langle S_1, \ldots, S_r \rangle$ a stratification of DB, and let $P_i = P \cap S_i$, $Z_i = Z \cap (S_1 \cup \cdots \cup S_i)$, $1 \leq i \leq r$, and let DB_i, $1 \leq i \leq r$, be the clauses from DB that contain only literals from S_j, $j \leq i$, in their heads modified by moving each $\neg x$ in the body to the head. The ICWA of DB can be characterized as follows (cf. [12]).

$$\begin{aligned} &\text{ICWA}_{P_1;Z_1}(DB_1) = \text{ECWA}_{P_1;Z_1}(DB_1), \\ &\text{ICWA}_{P_1 > \cdots > P_{n+1}; Z_{n+1}}(DB_{n+1}) = \\ &\quad \text{ECWA}_{P_{n+1};Z_{n+1}}(DB_{n+1} \cup \\ &\qquad\qquad \mathcal{F}(\text{ICWA}_{P_1 > \cdots > P_n; Z_n}(DB_n))), \\ &\qquad\qquad\qquad n > 0, \\ &\text{ICWA}_{P_1 > \cdots > P_r; Z}(DB) = \text{ICWA}_{P_1 > \cdots > P_r; Z_r}(DB_r), \end{aligned}$$

where $\mathcal{F}(\mathcal{M})$ is some DDDB DB' such that $M(DB') = \mathcal{M}$. Stratifiability asserts consistency; if DB is stratified by S, then ICWA is consistent for any $\langle P; Q; Z \rangle$ [12]. For inference, we have the following upper bound.

Theorem 4.1 *Given $\langle P; Q; Z \rangle$ and S, formula inference under ICWA is in Π_2^P.*

Proof. By results in [12, Section 6], ICWA is given by the following intersection of ECWAs:

$$\text{ICWA}_{P_1 > \cdots > P_r; Z}(DB) = \bigcap_{i=1}^{r} \text{ECWA}_{P_i; P_{i+1} \cup \cdots \cup P_r \cup Z}(DB_1 \cup \cdots \cup DB_r).$$

Since for each P', Z', and DB' a guess for $M \in \text{ECWA}_{P';Z'}(DB')$ can be verified in polynomial time with an NP oracle (cf. proof of Theorem 3.7), the same is possible for a guess of $M \in \text{ICWA}_{P_1 > \cdots > P_r; Z}(DB)$ such that $M \not\models F$. □

From Theorem 3.1 immediately follows (set $S = \langle V \rangle$)

Theorem 4.2 *Given a stratification S, literal inference from a DSDB DB under ICWA is Π_2^P-hard, even if DB is positive.*

5 Disjunctive normal databases

5.1 PERF

Przymusinski introduced in [19] the *Perfect Models Semantics* (PERF) for DNDBs without integrity clauses.[3] The priority relation $<$ on atoms [19], which is similar to stratification, is defined from the structure of DB using an auxiliary relation $\preceq$ as follows. (Intuitively, $x < y$ means that y has higher priority than x.) For each clause

$$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_k \wedge \neg c_1 \wedge \cdots \wedge \neg c_m$$

from DB, it holds that

[2] It is not clear how to extend stratification to programs with integrity clauses. Hence, we do not consider such an extension.

[3] The same definition can be readily extended to general DNDBs by ignoring integrity clauses in the definition of "$<$". The results in the general case remain unchanged [8].

(i) $a_i < c_j$, for all $1 \leq i \leq n, 1 \leq j \leq m$,

(ii) $a_i \preceq b_j$, for all $1 \leq i \leq n, 1 \leq j \leq k$, and

(iii) $a_i \preceq a_j$ for all $1 \leq i, j \leq n$.

Now $<$ and $\preceq$ are the smallest extensions to (i)–(iii) which satisfy $a < b \Rightarrow a \preceq b$ and which are closed under transitivity, i.e. $a \preceq b, b \preceq c \Rightarrow a \preceq c$ and $a \preceq b$, $b < c$ (resp. $d < a$) $\Rightarrow a < c$ (resp. $d < b$).

It is not hard to see that $<$ can be computed in polynomial time.

Definition 5.1 *The preference order $\ll$ on $M(DB)$ is defined by $M_1 \ll M_2$ iff $M_1 \neq M_2$ and for each $x \in M_1 - M_2$ there exists $y \in M_2 - M_1$ with $x < y$.*

Notice that $\ll$ is efficiently decidable, as easily seen from the definition of $<$. The perfect models of DNDB DB are as follows.

$$\mathrm{PERF}(DB) = \{M \in M(DB) : \forall M' \in M(DB).\ M' \not\ll M\}.$$

It holds that $M_1 < M_2 \Rightarrow M_1 \ll M_2$, hence $\mathrm{PERF}(DB) \subseteq MM(DB)$. Przymusinski showed that PERF(DB)= $MM(DB)$ if DB is positive; thus from Theorem 3.1,

Theorem 5.1 *Literal inference from a DNDB DB under PERF is Π_2^P-hard, even if DB is positive.*

On the other hand, since $\ll$ is polynomially decidable (cf. proof of Theorem 3.7),

Theorem 5.2 *Formula inference from a DNDB DB under PERF is in Π_2^P.*

Unlike the semantics considered so far, PERF semantics is not necessarily consistent if $M(DB)$ is nonempty.

Example 5.1 *$DB = \{p \leftarrow \neg q,\quad q \leftarrow \neg p\}$ has two minimal models $\{p\}$ and $\{q\}$, but $\{p\} \ll \{q\}$ and $\{q\} \ll \{p\}$, hence no model of DB can be perfect.*

In fact, even if it is known that DB has a model, deciding the existence of a perfect model seems difficult.

Theorem 5.3 *Deciding whether a DNDB DB has a perfect model is Σ_2^P-complete.*

Proof. (Sketch) Σ_2^P is an upper bound for deciding perfect model existence, since a guess for a perfect model can be verified with an NP oracle in polynomial time.

Hardness is shown by a reduction from deciding the validity of a $\exists\forall$-quantified Boolean formula

$$\Phi = \exists x_1 \cdots \exists x_n \forall y_1 \cdots \forall y_m E, \quad n, m \geq 1,$$

where $E = D_1 \vee \cdots \vee D_r$ and each $D_i = L_{i,1} \wedge L_{i,2} \wedge L_{i,3}$ is a conjunction of literals $L_{i,j}$ over atoms $x_1, \ldots, x_n$, $y_1, \ldots, y_m$. Let $v_1, \ldots, v_n$ and $z_1, \ldots, z_m, w, w'$ be new propositional atoms and define the following DNDB DB:

$$\begin{array}{lll}
\{x_i \vee v_i \leftarrow, & & \text{for } 1 \leq i \leq n \\
y_j \vee z_j \leftarrow, & & \text{for } 1 \leq j \leq m \\
y_j \leftarrow w \wedge w', & z_j \leftarrow w \wedge w' & \text{for } 1 \leq j \leq m \\
w \leftarrow y_j \wedge z_j, & w' \leftarrow y_j \wedge z_j & \text{for } 1 \leq j \leq m \\
w \vee w' \leftarrow \sigma(L_{1,1}) \wedge \sigma(L_{1,2}) \wedge \sigma(L_{1,3}) & & \\
\vdots & \vdots & \\
w \vee w' \leftarrow \sigma(L_{r,1}) \wedge \sigma(L_{r,2}) \wedge \sigma(L_{r,3}) & & \\
w \leftarrow \neg w', & w' \leftarrow \neg w\,\} &
\end{array}$$

where σ maps literals from atoms $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ to atoms as follows:

$$\sigma(L) = \begin{cases} v_i & \text{if } L = \neg x_i \text{ for some } i = 1, \ldots, n \\ z_j & \text{if } L = \neg y_j \text{ for some } j = 1, \ldots, m \\ L & \text{otherwise} \end{cases}$$

Notice that the interpretation where all variables are true is a model of DB. The structure of DB entails that there exists no variable p such that $x_i < p$ or $z_i < p$, for any $1 \leq i \leq n$, and that for each variable q from $\{w, w', y_1, z_1, \ldots, y_m, z_m\}$, it holds that $q < w$ and $q < w'$.

It can be shown that DB has a perfect model iff Φ is valid. □

An interesting case, which has to our best knowledge not analyzed yet, is the one where the database is not disjunctive. Call DB a *normal logic program* (NLP) if every clause in DB has exactly one atom in its head. We provide some evidence that deciding for the existence of a perfect model is for such programs not in NP ∪ coNP. We show this by a reduction from UMINSAT, which is the problem to decide if a collection $\mathcal{C} = \{C_1, \ldots, C_n\}$ of propositional clauses has a unique minimal satisfying truth assignment, i.e. whether a logically equivalent DNDB DB has a unique minimal model. The following has been shown in [7].

Proposition 5.4 *UMINSAT is* coNP*-hard and, unless the polynomial hierarchy collapses, UMINSAT does not belong to* coD^P.

coD^P is a complexity class [18] that contains –most likely properly– NP ∪ coNP.

Lemma 5.5 *UMINSAT is polynomially transformable to deciding whether a NLP DB has a unique minimal model.*

Proof. (Sketch) Let a, b, and c be new atoms not occurring in $\mathcal{C}$. Let DB be a DNDB logically equivalent to $\mathcal{C} \cup \{\neg c\}$. Clearly, DB has a unique minimal model iff $\mathcal{C}$ has a unique minimal satisfying truth assignment. Let the NLP DB' contain the following clauses:

(1) $a \leftarrow \neg b \wedge c$

(2) $x \leftarrow c$ for each atom x occurring in $\mathcal{C}$, and

(3) $c \leftarrow \neg C_i$ for each $C_i \in \mathcal{C}$,

where $\neg C_i$ is the conjunction of the opposites of all literals in C_i. It can be shown that DB' has a unique minimal model iff DB has a unique model; hence the lemma follows. □

Theorem 5.6 *Deciding whether an NLP DB has a perfect model is* coNP*-hard and, unless the polynomial hierarchy collapses, not in* NP ∪ coNP.

Proof. (Sketch) Let DB be an NLP as in Lemma 5.5. Let p be a new atom not occurring in DB, and let DB' be the NLP obtained from DB by adding for each pair of atoms q, r that occur in DB the clause $q \leftarrow \neg r \wedge p$. This clause assures that $q < r$ holds in DB'. Notice that no minimal model of DB' contains p. Thus, for any distinct minimal models M and M' of DB', $M \ll M'$ and $M' \ll M$ holds. It can be thus shown that DB' has a perfect model iff DB' has a unique minimal model. It holds that DB' has a unique minimal model iff DB has a unique minimal model. Consequently, DB' has a perfect model iff DB has a unique minimal model. Thus the result follows from the preceding lemma and proposition. □

For the inference problem, we obtain the following interesting lower bound.

Theorem 5.7 *Literal inference from an NLP DB under PERF is* NP*-hard and, unless the polynomial hierarchy collapses, not in* NP ∪ coNP.

Proof. (Sketch) Let DB be an NLP and p a propositional atom not occurring in DB. Then, $\mathrm{PERF}(DB \cup \{p \leftarrow\}) \models \neg p$ iff DB has no perfect model. The result follows from the previous theorem. □

For checking whether a model of a NLP DB is perfect, the following holds.

Theorem 5.8 *Deciding whether a model M of an NLP DB is perfect is* coNP*-complete.*

Proof. (Sketch) Membership in coNP is easy as a model M' such that $M' \ll M$ can be guessed and verified in polynomial time.

For the hardness part, we sketch a polynomial time reduction of deciding whether a DDDB DB' has no model ($M(DB') = \emptyset$), which is coNP-complete, to deciding whether a model M_0 of a nondisjunctive database DB is perfect.

Assume that DB' is built on atoms $x_1, \ldots, x_n$, and contains clauses

$$a_{i,1} \vee \cdots \vee a_{i,n_i} \leftarrow b_{i,1} \wedge \cdots \wedge b_{i,m_i}, \qquad 1 \leq i \leq r,$$

where $n_i, m_i \geq 0$. Without loss of generality we may assume that $\{x_1, \ldots, x_n\}$ is not a model of DB'. Let a, u, v be new atoms. DB is defined as follows:

$$\begin{array}{rcl} DB & = & \{x_1 \leftarrow a, \ldots, x_n \leftarrow a, \\ & & a \leftarrow u \wedge v, \quad u \leftarrow a, \quad v \leftarrow a, \\ & & v \leftarrow \neg u, \quad u \leftarrow \neg v, \\ & & a \leftarrow b_{1,1} \wedge \cdots \wedge b_{1,m_1} \wedge \neg a_{1,1} \wedge \cdots \wedge \neg a_{1,n_1}, \\ & & \qquad \vdots \qquad\qquad\qquad \vdots \\ & & a \leftarrow b_{r,1} \wedge \cdots \wedge b_{r,m_r} \wedge \neg a_{r,1} \wedge \cdots \wedge \neg a_{r,n_r} \}. \end{array}$$

It is easy to see that $M_0 = \{x_1, \ldots, x_n, a, u, v\}$ is a model of DB. Furthermore, it can be checked that for each atom z, it holds that $z < u$ and $z < v$. Every model M of DB such that $M \neq M_0$ must contain exactly one of u and v and satisfy $M \models \neg a$; thus, M satisfies DB'. On the other hand, if M is a model of DB', then $M \cup \{u\}$and $M \cup \{v\}$ are models of DB. Hence, it can be shown that M_0 is a perfect model of DB if and only if DB' has no model. □

5.2 DSM, PDSM

Generalizing the Stable Model Semantics [10], Przymusinski [20] defined the meaning of a DNDB DB by *disjunctive stable models* (DSM), which are a subset of the minimal models of DB as follows. For a DNDB DB and a Herbrand interpretation I, the Gelfond-Lifschitz transformation of DB is the database DB^I, which contains for each clause

$$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_k \wedge \neg b_{k+1} \wedge \cdots \wedge \neg b_m$$

from DB such that $b_i \notin I$, for all $k < i \leq m$, the clause

$$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_k$$

and DB^I contains nothing else. Then,

$$\mathrm{DSM}(DB) = \{M : M \in MM(DB^M)\}$$

In fact, disjunctive stable models of DB are minimal models of DB, i.e. $\mathrm{DSM}(DB) \subseteq MM(DB)$ [20]. Notice that if $DB \subseteq \mathcal{C}^+$, (i.e. "$\neg$" does not occur in DB), then $DB^I = DB$ for every I; hence $\mathrm{DSM}(DB) = MM(DB)$. Thus if DB is positive, deciding model existence is trivial. For the general case,

Theorem 5.9 *Deciding if a DNDB DB has a stable model is Σ_2^P-hard.*

Proof. (Sketch) Let DB be a positive DDB and $x \in V$. Add the clause $x \leftarrow \neg x$ to DB and let DB' be the resulting DB. Then, it holds that DB' has a stable model iff $MM(DB) \not\models \neg x$. The result follows from Theorem 3.1. □

Remark: It is not essential that a variable may occur in the head and negated in the body of the same clause to derive this result.

As immediate from Theorem 3.1,

Theorem 5.10 *Literal inference from a positive DNDB DB under DSM is Π_2^P-hard.*

Theorem 5.11 *Formula inference from a DNDB DB under DSM is in Π_2^P.*

Proof. (Sketch) A guess for $M \in \mathrm{DSM}(DB)$ such that $M \not\models F$ can be verified with an NP oracle in polynomial time (DB^M is efficiently computable). □

In [20] also the *Partial Disjunctive Stable Model Semantics* (PDSM) is introduced, which extends the *Well-Founded Semantics* of van Gelder, Ross, and Schlipf [29]. It is defined like DSM but using partial (i.e. 3-valued) models, in which true, false, and undefined variables have respective truth values $1, 0$, and 0.5. We do not define partial stable models here, and only outline in the sequel how the results in Tables 1 and 2 are derived. The interested reader is referred to [8] for the details.

Przymusinski shows that PDSM coincides with DSM if DB is positive; thus the Π_2^P lower bound on literal inference of DSM carries over to PDSM.

On the other hand, under 3-valued truth valuations the generalized Gelfond-Lifschitz transformation DB^I is still efficiently computable, and the truth value of a formula is efficiently computable. Furthermore, deciding partial model minimality in this framework is still in coNP. Consequently, a Π_2^P upper bound on formula inference for PDSM holds, as well as a Σ_2^P upper bound on model existence in the general case.

Since by adding the clause

$$\leftarrow x \wedge \neg x$$

for every $x \in V$, partial stable models can be forced to be total models, the Σ_2^P lower bound on deciding model existence for DSM carries over also to PDSM. However, integrity clauses are not crucial for proving Σ_2^P-hardness of PDSM-model existence. It can be shown that this lower bound holds if no integrity clauses are present [8].

Summarizing, we obtain the same complexity results for PDSM as for DSM.

References

[1] K. Apt, H. Blair, and A. Walker. Towards a Theory of Declarative Knowledge. In Minker [17], pages 89–148.

[2] N. Bidoit and C. Froidevaux. Negation by default and unstratifiable programs. *Theoretical Computer Science*, 78:85–112, 1991.

[3] M. Cadoli and M. Lenzerini. The Complexity of Closed World Reasoning and Circumscription. In *Proceedings AAAI-90*, pages 550–555, 1990.

[4] M. Cadoli and M. Schaerf. A Survey on Complexity Results for Non-monotonic Logics. Technical report, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", 1992. Journal of Logic Programming, to appear.

[5] E. Chan. A Possible Worlds Semantics for Disjunctive Databases. *IEEE Transactions on Data and Knowledge Engineering*, 1991. To appear.

[6] A. Chandra and D. Harel. Horn Clause Queries and Generalizations. *Journal of Logic Programming*, 2:1–15, 1985.

[7] T. Eiter and G. Gottlob. Propositional Circumscription and Extended Closed World Reasoning are $\mathbf{\Pi}_2^P$-complete. Technical Report CD-TR 91/20, Christian Doppler Laboratory for Expert Systems, Vienna University of Technology, Austria, May 1991. To appear in *Theoretical Computer Science*.

[8] T. Eiter and G. Gottlob. On the Computational Cost of Disjunctive Logic Programming: Propositional Case. Manuscript, March 1993.

[9] J. Fernández and J. Minker. Semantics of Disjunctive Deductive Databases. In *Proceedings ICDT-92*, pages 21–50, Berlin, October 1992.

[10] M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In *Proceedings Fifth Logic Programming Symposium*, pages 1070–1080, Cambridge Mass., 1988. MIT Press.

[11] M. Gelfond and H. Przymusinska. Negation as Failure: Careful Closure Procedure. *Artificial Intelligence*, 30:273–287, 1986.

[12] M. Gelfond, H. Przymusinska, and T. Przymusinski. On the Relationship Between Circumscription and Negation as Failure. *Artificial Intelligence*, 38:75–94, 1989.

[13] D. S. Johnson. A Catalog of Complexity Classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A, chapter 2. Elsevier Science Publishers B.V. (North-Holland), 1990.

[14] V. Lifschitz. Computing Circumscription. In *Proceedings IJCAI-85*, pages 121–127, 1985.

[15] W. Marek and M. Truszczyński. Autoepistemic Logic. *Journal of the ACM*, 38(3):588–619, 1991.

[16] J. Minker. On Indefinite Data Bases and the Closed World Assumption. In *Proceedings of the 6^{th} Conference on Automated Deduction (CADE)*, pages 292–308, 1982.

[17] J. Minker, editor. *Foundations of Deductive Databases and Logic Programming.* Morgan Kaufman, Washington DC, 1988.

[18] C. H. Papadimitriou and M. Yannakakis. The Complexity of Facets (And Some Facets of Complexity). *Journal of Computer and System Sciences*, 28:244–259, 1984.

[19] T. Przymusinski. On the Declarative and Procedural Semantics of Stratified Deductive Databases. In Minker [17], pages 193–216.

[20] T. Przymusinski. Stable Semantics for Disjunctive Programs. *New Generation Computing*, 9:401–424, 1991.

[21] A. Rajasekar, J. Lobo, and J. Minker. Weak Generalized Closed World Assumption. *Journal of Automated Reasoning*, 5:293–307, 1989.

[22] R. Reiter. On Closed-World Databases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 55–76. Plenum Press, New York, 1978.

[23] K. Ross and R. Topor. Inferring Negative Information From Disjunctive Databases. *Journal of Automated Reasoning*, 4(2):397–424, 1988.

[24] C. Sakama. Possible Model Semantics for Disjunctive Databases. In *Proceedings 1st Intl. Conf. DOOD-89*, pages 337–351, Kyoto Japan, 1989.

[25] M. Schaerf. Logic Programming and Autoepistemic Logics: New Relations and Complexity Results. Submitted, December 1992.

[26] M. Schaerf. Negation and Minimality in Non-Horn Databases. In *Proceedings PODS-93*, 1993.

[27] J. Schlipf. A Survey of Complexity and Undecidability Results in Logic Programming. In H. Blair, W. Marek, A. Nerode, and J. Remmel, editors, *Informal Proceedings of the Workshop on Structural Complexity and Recursion-Theoretic Methods in Logic Programming*, pages 93–102, Washington DC, November 13 1992. Cornell University, Mathematical Sciences Institute.

[28] A. van Gelder. Negation as Failure Using Tight Derivations for General Logic Programs. In Minker [17], pages 1149–1176.

[29] A. van Gelder, K. Ross, and J. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.

[30] A. Yahya and L. Henschen. Deduction in Non-Horn Databases. *Journal of Automated Reasoning*, 1(2):141–160, 1985.