In: Proceedings of the 5th International Conference on Database Theory (ICDT-95), Springer LNCS 893, pp. 68–81, 1995.

Querying Disjunctive Databases Through Nonmonotonic Logics

Piero A. Bonatti[1] and Thomas Eiter[2]

[1] Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy. email: bonatti@di.unito.it

[2] Christian Doppler Lab for Expert Systems, Institut für Informationssysteme, TU Wien, Paniglgasse 16, A-1040 Wien, Austria. email: eiter@vexpert.dbai.tuwien.ac.at

Abstract. In this paper we study the expressive power of major nonmonotonic formalisms – among them circumscription, default logic, and autoepistemic logic – used as query languages over disjunctive databases. For this aim, we define the semantics of query expressions formulated in different nonmonotonic logics. The expressive power of the languages that we consider has been explored in the context of relational databases. Here, we extend this study to disjunctive databases; as a result, we obtain a finer-grained characterization of the expressive capabilities of those languages and interesting fragments thereof.

In particular, we show that there exist queries that cannot be expressed by any preference-based semantics (including the minimal model semantics and the various forms of circumscription), but which can be expressed in default and autoepistemic logic. Secondly, we show that default logic, autoepistemic logic and some of their fragments express the same class of Boolean queries, which turns out to be a strict subclass of the Σ_2^p-recognizable Boolean queries. Then we prove that under the assumption that the database consists of clauses whose length is bound by some constant, default logic and autoepistemic logic express *all* of the Σ_2^p-recognizable Boolean queries, while preference-based logics do not.

1 Introduction

Disjunctive databases provide one possibility for representing incomplete information – a topic which has been receiving much attention in the areas of databases, logic programming and AI, cf. [12, 16, 9].

Query languages for retrieving information from disjunctive databases are an open area of research. In the framework of relational databases, many alternative query languages have been proposed; they have been extensively studied in the literature (cf. [25, 14, 6, 7, 2] for fundamentals and overviews) and, in particular, they have been compared to each other by characterizing their *expressiveness*, i.e., the class of queries that they can express. These relationships are quite well understood [7, 2]. The study of query languages for disjunctive databases, constituted by collections of ground clauses rather than facts, is much less established. Work on querying incomplete databases is related; cf. [12, 1] for relations with null values, and [26] for the similar proviso of missing unique names axioms in logical databases. However, such work is basically concerned with expressing disjunctive information of the type "$p(a)$ or $p(b)$" instead of "$p(a)$ or $q(b)$", i.e., disjunctions involving ground facts from the same relation, rather than ground facts from distinct relations.

Advanced query languages for relational databases are typically non-monotonic, i.e. after extending the database some facts may no longer be derivable. Indeed some query languages are built on well-known nonmonotonic formalisms such as logic programming languages, circumscription and default logic [24, 8, 5]. The use of nonmonotonic formalisms as query languages is justified by their expressive power; see [5] for motivating examples.

In this paper we study the expressive power of major non-monotonic formalisms used as query languages over disjunctive databases. We will pay particular attention to *positive* disjunctive databases which, roughly speaking, are sets of positive clauses. Such databases enjoy nice properties that make them suitable for implementations. For instance, deciding whether a clause is entailed by such a database can be done in polynomial time.

The main contributions of this paper are the following:

- We define the semantics of query expressions formulated in different nonmonotonic logics.
- We prove that the minimal model semantics (and, more generally, circumscription; cf. [10]), default logic [23] and autoepistemic logic [20] cannot express all the PTIME-recognizable properties of positive disjunctive databases; this result is surprising, because in the framework of relational databases these logics are able to express every Σ_2^p-recognizable query. More surprisingly, the same negative result holds for all model-preference based logics, some of which are tremendously powerful (not even recursively enumerable), and for the perfect and the stable models semantics of logic programs (cf. [10]). This kind of negative results is usually proved by means of persistency properties, cf. [15]; for default logic we use a new technique based on a counting argument.
- If we impose a constant bound on the length of database clauses, then default logic and autoepistemic logic can express every Σ_2^p-recognizable query, while model-preference based logics (including circumscription and the minimal model semantics) cannot express all of the PTIME-recognizable ones. It follows that default logic and autoepistemic logic are strictly more expressive than disjunctive datalog, in the framework of bounded databases. This result is somewhat unexpected: in the framework of relational databases, these formalisms have exactly the same capabilities.
- We identify some interesting subsets of default and autoepistemic logic and give a complete picture of their mutual relationships. For instance, we show that prerequisites do not increase the expressive power of default logic, and that inconsistent stable expansions do not increase the expressive power of

autoepistemic logic.

Due to space limitations, we cannot present in detail all concepts and proofs. Details and proofs of all results can be found in the full paper [3].

2 Preliminaries

A *relational scheme* over a domain Dom is a list $\overline{R} = R_1, \ldots, R_z$ of relation (or predicate) symbols R_i of arity $a_i \geq 0$ $(i = 1, \ldots, z)$. We assume that Dom, which will not be explicitly mentioned, is countably infinite. A *disjunctive database over* $\overline{R}$ is a pair $D = (U, \phi)$, where $U \subseteq$ Dom is a finite set of constants (universe) and ϕ is a conjunction of first order formulas

$$A_1 \vee \cdots \vee A_l \leftarrow B_1 \wedge \cdots \wedge B_m\,, \qquad l > 0\,,\ m \geq 0\,, \tag{1}$$

where $A_1, \ldots, A_l, B_1, \ldots, B_m$ are ground atoms, built from the predicate symbols in $\overline{R}$ and from the constants in U. The empty conjunction (denoted by $\top$) represents the empty database; the universe of D is denoted by $U(D)$. The set of disjunctive databases over $\overline{R}$ is denoted by $\mathcal{D}(\overline{R})$. A disjunctive rule of form (1) is *positive* if $m = 0$; a disjunctive database (U, ϕ) is *positive* if ϕ is a conjunction of positive rules.

For any integer $d \geq 1$, we denote by $\mathcal{D}(\overline{R})_{\leq d}$ the set of databases (U, ϕ) from $\mathcal{D}(\overline{R})$ such that for each clause C occurring in ϕ, the *dimension* of C, which is the number of its disjuncts, is not larger than d. Notice that $\mathcal{D}(\overline{R})_{\leq 1}$ corresponds in an obvious way to the relational databases over $\overline{R}$, i.e., the finite structures $\langle U, r_1^{a_1}, \ldots, r_z^{a_z} \rangle$ where $r_i^{a_i} \subseteq U^{a_i}$.

A clause C *subsumes* a clause C' iff every disjunct of C is also a disjunct of C'. A disjunct of a clause C is *redundant* if it equals a distinct disjunct of C. A conjunct C_i of a CNF sentence $C_1 \wedge \cdots \wedge C_n$ is *redundant* if it is subsumed by some C_j where $j \neq i$.

A sentence ϕ is in *normal form* iff ϕ is in CNF, it contains no redundant conjuncts, and its conjuncts contain no redundant literals; a database $D = (U, \phi)$ is in normal form iff ϕ is.
Clearly, by removing redundant literals and clauses, every CNF-sentence ϕ (resp. disjunctive database D) can be transformed in quadratic time into a logically equivalent normal sentence (resp. disjunctive database), that will be denoted by ϕ^* (resp. D^*); this sentence (resp. database) is unique modulo the ordering of clauses and their disjuncts. Positive databases enjoy the following properties:

Proposition 1. *For all positive databases $D = (U, \phi)$: (a) for all non-tautological clauses C, $\phi \models C \iff C$ is subsumed by some conjunct of $\phi \iff C$ is a prime clause of ϕ. (b) for all positive databases $D_1 = (U, \phi_1)$, ϕ_1 is logically equivalent to ϕ iff $\phi^* = \phi_1^*$ (up to clause and literal ordering).*

2.1 Default Logic

A default theory ([23]; cf. [17, 19] for a background) $T = \langle W, \Delta \rangle$ is a pair of a set W of first-order formulas and a set Δ of default rules

$$\frac{\alpha(\mathbf{x}) : \beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x})}{\gamma(\mathbf{x})}$$

such that the *prerequisite* $\alpha(\mathbf{x})$, the *justifications* $\beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x})$ and the *consequent* $\gamma(\mathbf{x})$ are first-order formulas whose free variables are among those of $\mathbf{x} = x_1, \ldots, x_m$. When $n = 0$, then $\top$ (truth) is implicitly assumed as the justification of the default. We will omit writing $\alpha(\mathbf{x})$ if the prerequisite is $\top$; such defaults will be called *prerequisite-free*. A default is *closed* if it has no free variables, and *open* otherwise; the same notion is defined for default theories in the obvious way.

The semantics of a closed default theory $\langle W, \Delta \rangle$ is based on *extensions*. Formally, an extension E can be characterized through a quasi-inductive construction as follows. Let E be a set of first-order formulas. Define

$$\begin{aligned} E_0 &= W, \qquad \text{and for } i \geq 0 \\ E_{i+1} &= \mathrm{Th}(E_i) \cup \{\, \gamma \mid (\alpha : \beta_1, \ldots, \beta_n \,/\, \gamma) \in \Delta,\ \alpha \in E_i,\ \forall j.\, \neg\beta_j \notin E \,\} \end{aligned}$$

where $\mathrm{Th}(\cdot)$ denotes classical deductive closure. Then, E is an extension of T iff $E = \bigcup_{i=0}^{\infty} E_i$. Each extension E can be constructed from its *generating defaults*. For a set S of formulas and a set Δ of defaults, let

$$\mathrm{GD}(\Delta, S) = \{\, \delta \in \Delta \mid \delta = (\alpha : \beta_1, \ldots, \beta_n \,/\, \gamma),\ S \vdash \alpha,\ \forall j.\, S \not\vdash \neg\beta_j \,\}$$

be the generating defaults of S w.r.t. Δ. Furthermore, denote by $\mathrm{CONS}(\Delta)$ the set of the consequents of the defaults in Δ.

Lemma 2. [23, Theorem 2.5] *Let E be an extension of the closed default theory $T = \langle W, \Delta \rangle$. Then, $E = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E)))$.*

3 Query Languages for Disjunctive Databases

Query languages for relational databases are a well-studied topic in the theory of relational databases, cf. [25, 6, 14, 2]. The notion of a query (which maps each relational database onto another relational database) can be naturally extended to disjunctive databases.

Let $\overline{R}$ and $\overline{S}$ be relational schemes. A *database mapping* or *query* is a recursive function $q : \mathcal{D}(\overline{R}) \to \mathcal{D}(\overline{S})$ such that $U(q(D)) = U(D)$; moreover, we require q to be invariant under isomorphism (i.e., *generic* in the sense of [2]) and under logical consequence, i.e., $q(D) = q(D')$ if D and D' are logically equivalent. The latter condition insures that queries are not sensitive to the syntactic representation of disjunctive information. The pair $(\overline{R}, \overline{S})$ is called the *input/output scheme of* q and is denoted by $IO(q)$; it is assumed that $\overline{R}$ and $\overline{S}$ are disjoint.

The queries which compute relations, i.e. such that $q(D) \in \mathcal{D}(\overline{R})_{\leq 1}$ for all D, are of natural interest; they deduce only atomic facts from the database. An important special case of such relational output-queries are those where $\overline{S}$ consists of a single 0-ary relation symbol P, i.e., a propositional letter. Such *Boolean* queries model yes/no queries on disjunctive databases.

Boolean queries naturally correspond to *database properties*. For a given relational scheme $\overline{R}$, a database property is a predicate $\mathbf{P}$ which associates with every database D over $\overline{R}$ a truth value $\mathbf{P}(D)$ from $\{true,\ false\}$ and which is closed under isomorphism and logical equivalence. The database property corresponding to a Boolean query $q : \mathcal{D}(\overline{R}) \to \mathcal{D}(S)$ is denoted by $\mathbf{P}_q$; for all $D \in \mathcal{D}(\overline{R})$, we have $\mathbf{P}_q(D) = true$ iff $q(D) = (U(D), S)$ and $\mathbf{P}_q(D) = false$ iff $q(D) = (U(D), \top)$.

Let $\mathbf{C}$ be any complexity class (based on the Turing machine model). Following the approach in [25, 13], we say that a database property is $\mathbf{C}$-*recognizable* (or, *in* $\mathbf{C}$) if, given D, deciding whether $\mathbf{P}(D) = true$ is in $\mathbf{C}$; a query q is $\mathbf{C}$-*recognizable* (or, *in* $\mathbf{C}$) if, given D and a ground clause C, deciding whether $C \in q(D)$ is in $\mathbf{C}$.[3]

4 Nonmonotonic Logics as Query Languages

In this section we define the semantics of preference-based logics and default logic as query languages for expressing Boolean properties of disjunctive databases; a generalization to relational-output queries is straightforward.

We follow the deductive view of datalog and other logical query languages. Roughly speaking, the answer to the query expressed by a theory T in a logic L evaluated against a disjunctive database $D = (U, \phi)$ is "yes" iff a designated output letter P is entailed in L by the union of T and ϕ, where T is instantiated over D.

Notation. For all sets of first-order formulas T and any set U, we will denote by $[T]_U$ the set of closed instances of the formulas of T, obtained by binding their free variables to elements of U in all possible ways.

We first define the semantics of first-order query expressions T under preferential semantics. For all disjunctive databases $D = (U, \phi)$ define

$$T + D = [T]_U \cup \{\phi\}$$

Definition 3. Let T be a set of first-order formulas on a relational vocabulary including $\overline{R}$ and a propositional letter S not occurring in $\overline{R}$, and let $\prec$ be a strict partial order (preference relation) on the models of T. Then, T expresses under the brave preferential semantics the Boolean query $q : \mathcal{D}(\overline{R}) \to \mathcal{D}(S)$ iff for all $D \in \mathcal{D}(\overline{R})$,

$\mathbf{P}_q(D) = true$ iff S is true in at least one $\prec$-minimal model of $T + D$.

[3] We assume a natural encoding in which U is described by its cardinality written down in unary notation. Equivalently w.r.t. to polynomial time computability, we could represent U by an enumeration of its elements, cf. [13].

Similarly, T expresses under the cautious preferential semantics the Boolean query $q : \mathcal{D}(\overline{R}) \to \mathcal{D}(S)$ iff for all $D \in \mathcal{D}(\overline{R})$,

$$\mathbf{P}_q(D) = \mathit{true} \text{ iff } S \text{ is true in all } \prec\text{-minimal models of } T + D\,.$$

Note that the standard minimal model semantics for logic programs is a preference based logic where $\prec = \subseteq$. Similarly, monotonic logics are special cases of preference-based logics, where $\prec$ is empty.

Next we define the semantics of a default logic query. Let $T = \langle W, \Delta \rangle$ be a default theory, whose underlying first-order language is function-free and quantifier-free and includes the predicates from $\overline{R}$.[4] By $[\Delta]_U$ we denote the set of all ground instances of defaults from Δ obtained by binding their free variables to elements of U in all possible ways. Finally, for all disjunctive databases $D = (U, \phi)$ define

$$T + D = \langle\, \{\phi\} \cup [W]_U,\, [\Delta]_U \,\rangle\,.$$

Definition 4. Let $T = \langle W, \Delta \rangle$ be a default theory. Then, T *expresses* the Boolean query $q : \mathcal{D}(\overline{R}) \to \mathcal{D}(S)$ under the *brave* semantics iff for all databases $D \in \mathcal{D}(R)$,

$$\mathbf{P}_q(D) = \mathit{true} \text{ iff } T + D \text{ has an extension which contains } S\,.$$

The definition of the cautious semantics is *mutatis mutandis.* Now each database property $\mathbf{P}$ over $\mathcal{D}(\overline{R})$ under brave (as well as cautious) default semantics might be expressed by means of a suitable Boolean query $q : \mathcal{D}(\overline{R}) \to \mathcal{D}(S)$ such that $\mathbf{P}_q = \mathbf{P}$. However, it will be convenient – and probably more appealing – to adopt the following definition, which is simpler to handle and yields the same notion of expressive power:

Definition 5. T expresses $\mathbf{P}$ over $\mathcal{D}(\overline{R})$ under the *brave* semantics iff for all databases $D \in \mathcal{D}(R)$, $\mathbf{P}(D) = \mathit{true}$ iff $T + D$ has an extension.

5 Expressive Power

It has been recently shown that the brave version of parallel circumscription captures the Σ^p_2-recognizable database properties of relational databases (provided that the language contains equality) [8]. We are going to prove that this result cannot be generalized to disjunctive databases. Actually, we will prove a stronger result, namely, that *all preference-based semantics have the same limitation.* For this purpose, consider the following property:

Definition 6. Let R be an a-ary predicate symbol in $\overline{R}$. For all $D = (U, \phi)$ define $\mathbf{P}_R(D) = \mathit{true}$ iff for some tuples $\mathbf{c}_1, \ldots, \mathbf{c}_n$ in $U(D)^a$, $\phi \models R(\mathbf{c}_1) \vee \ldots \vee R(\mathbf{c}_n)$ and $\phi \not\models R(\mathbf{c}_i) \quad (i = 1, \ldots, n)$.

[4] The restriction to a function and quantifier-free first-order language is common in database theory.

To give an intuitive meaning to the above property, we remind that certain null values, that represent unknown attributes belonging to some fixed range $\{ v_1, \ldots, v_n \}$, can be specified by disjunctions of the form $R(c, v_1) \vee \ldots \vee R(c, v_n)$. From this point of view, a database (U, ϕ) has the property $\mathbf{P}_R$ iff D contains a proper null value.

Theorem 7. *The property $\mathbf{P}_R$ cannot be expressed by any first order theory under any brave preferential semantics.*

The proof of this theorem is based on the following lemma:

Lemma 8. *Let $\prec$ be any binary relation over the set of interpretations, and for all sets of sentences S, let $\mathrm{MM}(S)$ denote the set of $\prec$-minimal models of S.*[5] *For all sets of sentences S and $\{ \sigma_1, \ldots, \sigma_n \}$ we have:* $\mathrm{MM}(S \cup \{ \sigma_1 \vee \ldots \vee \sigma_n \}) \subseteq \mathrm{MM}(S \cup \{ \sigma_1 \}) \cup \ldots \cup \mathrm{MM}(S \cup \{ \sigma_n \})$.

Proof. (of Theorem 7) Suppose that some theory T expresses $\mathbf{P}_R$ under some preferential semantics based on a preference relation $\prec$, i.e. for all databases $D = (U, \phi)$, $\mathbf{P}_R(D) = \mathit{true}$ iff the output proposition P is true in some $\prec$-minimal model of $[T]_U \cup \{ \phi \}$. Then, for any given universe U and for any distinct tuples of U-constants $\mathbf{c}$ and $\mathbf{d}$, $[T]_U \cup \{ R(\mathbf{c}) \vee R(\mathbf{d}) \}$ must have a minimal model M where P is true. By Lemma 8, M must be a minimal model of either $[T]_U \cup \{ R(\mathbf{c}) \}$ or $[T]_U \cup \{ R(\mathbf{d}) \}$; but neither $(U, R(\mathbf{c}))$ nor $(U, R(\mathbf{d}))$ have the property $\mathbf{P}_R$, therefore T does not express $\mathbf{P}_R$; a contradiction. $\square$

A similar result can be obtained for the property $\mathbf{P}_{\mathrm{MM}}$ defined as follows: $\mathbf{P}_{\mathrm{MM}}(D) = \mathit{true}$ iff $D = (U, \phi)$ and ϕ has at least two distinct minimal models (i.e. D is truly disjunctive and it is not equivalent to any relational database).

Both $\mathbf{P}_R$ and $\mathbf{P}_{\mathrm{MM}}$ can be computed on positive databases in polynomial time. Thus, there exist properties of positive databases that are computable in polynomial time, but cannot be expressed under any brave preferential semantics. The same holds for cautious preferential semantics. (This can be easily derived from the database property complementary to $\mathbf{P}_R$.)

Remark. The above results can be extended also to the *stable model semantics* [21] – one of the major semantics for disjunctive logic programs – although it is not really a preferential semantics, but rather an hybrid between a preferential semantics and the semantics of default logic. By means of Lemma 8 it is easy to prove the following proposition:

> Each stable model of $T + D$, where $D = (U,\ R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n))$ is also a stable model of $T + D'$, for some database $D' = (U,\ R(\mathbf{c}_i)) \quad (1 \leq i \leq n)$.

It follows that no disjunctive logic program expresses $\mathbf{P}_R$ or $\mathbf{P}_{\mathrm{MM}}$ under the stable model semantics (the proof is similar to that of Theorem 7; apply the

[5] By $\prec$-minimal models we mean those models M such that for no model N, $N \prec M$.

above proposition instead of Lemma 8). Moreover, through similar techniques, one can extend the proof to the *perfect model semantics* [22] (the above property of stable models applies to perfect models, too.) Similar results have already been proved for other important semantics of disjunctive logic programs, cf. [4].

5.1 Expressive Power of Default Logic

Default logic can express the property $\mathbf{P}_R$. The default theory $T_R = \langle W_R, \Delta_R \rangle$, whose formulas and defaults are as follows, is suitable for this purpose:

$$\text{(F)}\; C(\mathbf{x}) \wedge R(\mathbf{x}) \rightarrow S \qquad \text{(D1)}\; \frac{: C(\mathbf{x})}{C(\mathbf{x})} \qquad \text{(D2)}\; \frac{: \neg C(\mathbf{x})}{\neg C(\mathbf{x})}$$

$$\text{(D3)}\; \frac{C(\mathbf{x}) \wedge R(\mathbf{x}) :}{\bot} \qquad \text{(D4)}\; \frac{: \neg S}{\bot}$$

The predicates C and S are new and do not occur among the database predicates $\overline{R}$. Intuitively, the extension of C describes a clause $R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)$, where $C(\mathbf{x})$ means that $R(\mathbf{x})$ occurs in the clause. The clause is selected by the defaults (D1)-(D2); the default (D3) assures that the clause is truly disjunctive, i.e., does not contain some $R(\mathbf{c}_i)$ such that $\phi \models R(\mathbf{c}_i)$. The formula (F) and the default (D4) check that the clause is implied by ϕ, by assuring that the propositional variable S is derivable.

Proposition 9. *For every database $D = (U, \phi)$ over $\overline{R}$, the default theory $T_R + D$ has an extension iff $\mathbf{P}_R(D) = true$.*

Now the question is: what is the class of properties of disjunctive databases that can be expressed through default logic? From well-known results on the complexity of propositional default logic, we obtain easily the following upper bound for the expressive power of default logic.

Theorem 10. *The database properties expressible in default logic are Σ_2^p-recognizable.*

Proof. As shown in [11], deciding whether a propositional default theory has an extension is a Σ_2^p-complete problem. For every default theory T and database D, the size of $T + D$ is polynomial in the size of T plus the size of D. Hence, the result follows. □

Moreover, there exist database properties defined in default logic which are Σ_2^p-hard to compute (cf. Lemma 19 below). Thus, default logic seems to be powerful; the question is whether it is a well-balanced query language, i.e., can default logic express *all* the Σ_2^p-recognizable properties of disjunctive databases?

The answer is no. We will prove this result in two steps; first we will show that prerequisite-free default logic has the same expressive power as unrestricted default logic; then we will show that prerequisite-free default logic cannot express a simple property of positive databases.

Theorem 11. *For all default theories $T = \langle W, \Delta \rangle$ there exists a prerequisite-free default theory* $\mathrm{pf}(T) = \langle W, \mathrm{pf}(\Delta) \rangle$ *that expresses the same property as T.*

The proof – which is omitted here – describes $\mathrm{pf}(\Delta)$, which roughly speaking simulates the defaults of T without deriving their consequents.[6] Now consider the following property:

Definition 12. For all positive databases $D = (U, \phi)$ from $\mathcal{D}(\overline{R})$, $\mathbf{P}_{\mathrm{ev}}(D) =$ *true* iff the dimensions of the clauses in ϕ^* are all even.

Theorem 13. $\mathbf{P}_{\mathrm{ev}}$ *is not expressed by any prerequisite-free default theory, under brave reasoning.*

It follows easily from this theorem that there exists a Σ_2^p-recognizable database property over general disjunctive databases which cannot be expressed in default logic: Simply extend the definition of $\mathbf{P}_{\mathrm{ev}}$ such that $\mathbf{P}_{\mathrm{ev}}(D) = \mathbf{P}_{\mathrm{ev}}(D')$ if D is logically equivalent to a positive database D', and $\mathbf{P}_{\mathrm{ev}}(D) =$ *false* otherwise; call this property $\mathbf{P}_{\mathrm{ev,p}}$.

Basically, the proof of Theorem 13 goes as follows: we exploit two interpolation properties of DL (Lemmas 14 and 15) to prove that, in order to distinguish the databases with property $\mathbf{P}_{\mathrm{ev}}$ from the ones that do not have this property, we need numerous defaults (Lemma 16); then we show that no fixed default theory can provide so many defaults.

Lemma 14. *Let $T_1 = \langle W_1, \Delta \rangle$ and $T_2 = \langle W_2, \Delta \rangle$ be closed prerequisite-free default theories such that, for some W, $W_1 \models W$ and $W \models W_2$. If T_1 and T_2 have extensions E_1 and E_2 (respectively) such that* $\mathrm{GD}(\Delta, E_2) = \mathrm{GD}(\Delta, E_1)$ *then also the theory $T = \langle W, \Delta \rangle$ has an extension E such that* $\mathrm{GD}(\Delta, E) = \mathrm{GD}(\Delta, E_1)$.

Lemma 15. *Let $T_1 = \langle \{ \phi_1 \}, \Delta \rangle$ and $T_2 = \langle \{ \phi_2 \}, \Delta \rangle$ be prerequisite-free closed default theories. If T_1 has an extension E_1 and T_2 has an extension E_2 such that* $\mathrm{GD}(\Delta, E_2) = \mathrm{GD}(\Delta, E_1)$ *then also the theory $T = \langle \{ \phi_1 \vee \phi_2 \}, \Delta \rangle$ has an extension E such that* $\mathrm{GD}(\Delta, E) = \mathrm{GD}(\Delta, E_1)$.

Lemma 16. *Let $D_1 = (U, \phi_1)$ and $D_2 = (U, \phi_2)$ be non-equivalent positive databases with property $\mathbf{P}_{\mathrm{ev}}$, and let $T = \langle W, \Delta \rangle$ be a prerequisite-free theory that expresses $\mathbf{P}_{\mathrm{ev}}$. For all extensions E_1 and E_2 of $T + D_1$ and $T + D_2$ (respectively) we have* $\mathrm{GD}([\Delta]_U, E_1) \neq \mathrm{GD}([\Delta]_U, E_2)$.

Proof. (of Theorem 13) Suppose that the theorem is not valid, i.e. assume that there exists a prerequisite-free theory $T = \langle W, \Delta \rangle$ that expresses $\mathbf{P}_{\mathrm{ev}}$; we will derive a contradiction.

[6] It was already known that every default theory can be translated into an equivalent prerequisite-free theory, but it was not known how to translate Δ into a single theory $\mathrm{pf}(\Delta)$, preserving the meaning of Δ for *all* possible input databases D.

Let U be any universe with cardinality $n = 4i$ with i integer. Let m be the cardinality of the corresponding Herbrand base; note that $m = 4j$ for some integer j.

Let $S = \{ D_1, \ldots, D_N \}$ be the set of normal positive databases with universe U that have the property $\mathbf{P}_{\text{ev}}$. To prove the theorem it suffices to show that the following inequalities hold for some polynomial $p(n)$ in n:

$$N \leq 2^{p(n)} \tag{2}$$

$$N \geq 2^{2^{n/2}} \tag{3}$$

In fact, (2) and (3) imply $2^{2^{n/2}} \leq 2^{p(n)}$ but for sufficiently large universes we have $2^{2^{n/2}} > 2^{p(n)}$, hence a contradiction.

First we prove (2). By assumption, T expresses $\mathbf{P}_{\text{ev}}$, therefore, for all $D_h \in S$, $T + D_h$ has an extension E_h. Note that distinct normal databases are not equivalent; therefore, by applying Lemma 16 to all pairs of databases of S we derive that each extension E_h corresponds to a distinct set of generating defaults $\mathrm{GD}([\Delta]_U, E_h)$. The number of possible sets of generating defaults is bounded by 2^l, where l is the cardinality of $[\Delta]_U$, therefore $N \leq 2^l$. Moreover, l can be expressed as a polynomial in n, say $p(n)$, where the exponents are determined by the number of free variables in the defaults of Δ. It follows that $N \leq 2^{p(n)}$, which proves (2).

We are left to prove (3). The number of normal clauses of dimension $2j$ is:

$$\binom{m}{2j} = \binom{m}{m/2} \geq 2^{m/2} \geq 2^{n/2}$$

therefore, the number of databases of S that contain only clauses of dimension $2j$ is at least $2^{2^{n/2}}$. Disequation (3) follows immediately. □

Since $\mathbf{P}_{\text{ev}}$ can be computed on databases in polynomial time (because ϕ^* can), we conclude that:

Corollary 17. *There exist PTIME-recognizable properties of positive databases that cannot be expressed through brave reasoning in prerequisite-free default logic.*

However, for the class of bounded databases, i.e., the databases in $\mathcal{D}(\overline{R})_{\leq d}$, for a fixed constant d, we have the following.

Theorem 18. *For every Σ_2^p-recognizable property $\mathbf{P}$ defined over $\mathcal{D}(\overline{R})_{\leq d}$, for d a constant, there exists a default theory T that defines $\mathbf{P}$ under brave reasoning.*

This result can be intuitively grasped through two facts. First, each database from $\mathcal{D}(\overline{R})_{\leq d}$ can be represented in a relational database over some scheme $\overline{R}'$. Second, all Σ_2^p-recognizable queries over relational databases can be expressed in default logic [5]. Recall that we can think of a relational database over $\overline{R}'$ as a database $D \in \mathcal{D}(\overline{R}')_{\leq 1}$, where the clauses in D (which are facts) correspond to the tuples stored in the database relations. The following lemma rephrases the main result in [5].

Lemma 19. *Let* $\mathbf{P}$ *be a* Σ_2^p*-recognizable database property over* $\mathcal{D}(\overline{R})_{\leq 1}$ *. Then, there exists a set of defaults* Δ *such that for every* $D = (U, \phi) \in \mathcal{D}(\overline{R})_{\leq 1}$ *, the default theory* $\langle\, \mathrm{COMP}(D),\, [\Delta]_U \,\rangle$ *has an extension iff* $\mathbf{P}(D) = true$*, where* $\mathrm{COMP}(D)$ *consists of all ground atoms occurring in* ϕ *and the negations of all ground atoms over* U *not occurring in* ϕ *.*[7]

Thus, intuitively, T can be composed of two subtheories $T_1 = \langle\, \emptyset, \Delta_1 \rangle$ and $T_2 = \langle\, \emptyset, \Delta_2 \rangle$, such that T_1 transforms the input (disjunctive) database D into a relational database D', and T_2 tests whether D' has the property $\mathbf{P}'$ corresponding to $\mathbf{P}$.

It is interesting to see that model-preference based logics cannot express all the Σ_2^p-recognizable properties over bounded databases.

Theorem 20. *For all* $d > 1$ *, the properties* $\mathbf{P}_{\mathrm{MM}}$ *and* $\mathbf{P}_R$ *restricted to* $\mathcal{D}(\overline{R})_{\leq d}$ *cannot be expressed by any first order theory under any brave preferential semantics.*

As a consequence, as far as bounded databases are concerned, default logic is strictly more expressive than any model-preference based logic whose brave semantics is in Σ_2^p.

6 Conclusion

The main results of the paper for brave semantics are summarized in Fig. 1 and Fig. 2. An arrow from A to B means that every property which can be expressed in A can also be expressed in B. Dotted arrows correspond to obvious relations; the other arrows illustrate results of the paper. $\mathbf{P}_R$, $\mathbf{P}_{\mathrm{MM}}$ and $\mathbf{P}_{\mathrm{ev,p}}$ are the simple disjunctive database properties introduced in Sect. 5. A picture of the results for the dual cautious semantics can be obtained by replacing Σ_2^p , $\mathbf{P}_R$, $\mathbf{P}_{\mathrm{MM}}$ and $\mathbf{P}_{\mathrm{ev,p}}$ with Π_2^p and the respective complementary properties.

The figures also include results on autoepistemic logic (AEL) [20] as a query language; its semantics is defined in a similar way as the semantics of default logic. The language of AEL is a modal language with one modal operator L , to be read as "know" or "believe"; for an extensive treatment, cf. [18, 17]. An autoepistemic theory is a set of autoepistemic formulas, and called *closed* if all of its members are closed, and *open* otherwise. The semantics of closed autoepistemic theories is based on *stable expansions*, which are the counterparts of default extensions. A set of sentences S is a stable expansion of an autoepistemic theory T iff it satisfies the following fixpoint equation:

$$S = \{\, \phi \mid T \cup \{\, L\psi \mid \psi \in S \,\} \cup \{\, \neg L\psi \mid \psi \notin S \,\} \vdash \phi \,\},$$

where $\vdash$ denotes classical derivability. Prerequisite-free AEL is the fragment where all formulas in T are of the form

$$\neg\, L\beta_1 \wedge \cdots \wedge \neg\, L\beta_n \rightarrow \phi\,, \qquad n \geq 0\,, \tag{4}$$

[7] $\mathrm{COMP}(\phi)$ yields the completion of ϕ under the closed-world assumption.

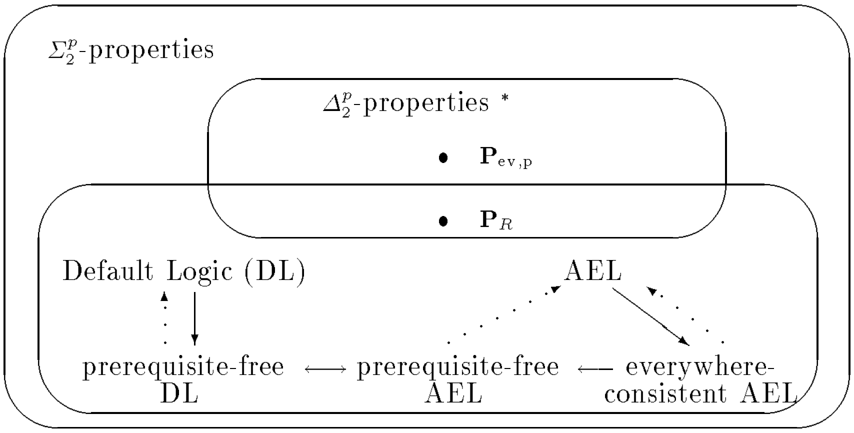

Fig. 1. Expressiveness for unbounded databases

* Over positive disjunctive databases, $\mathbf{P}_R$, $\mathbf{P}_{\mathrm{MM}}$, and $\mathbf{P}_{\mathrm{ev,p}}$ are polynomial-time recognizable.

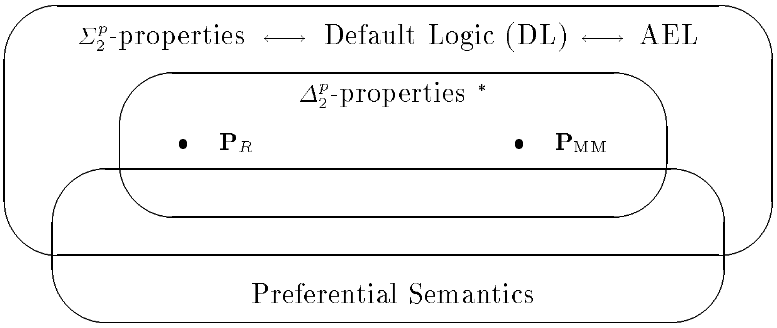

Fig. 2. Expressiveness for bounded databases

* Over positive disjunctive databases, $\mathbf{P}_R$, $\mathbf{P}_{\mathrm{MM}}$, and $\mathbf{P}_{\mathrm{ev,p}}$ are polynomial-time recognizable.

where $\beta_1, \ldots, \beta_n$ and ϕ are objective (L does not occur); everywhere consistent AEL is the subclass of AEL theories T such that for each database D, the instantiation $T + D$ (defined similarly as for default theories) has only consistent expansions.

The proofs for the results in Figs. 1 and 2 are constructive; they allow to effectively transform a query expressed in any of those fragments into an equivalent query expressed in any other of the fragments. Moreover, the transformations are polynomial.

The study of expressiveness w.r.t. generalizations of relational databases leads to deeper understanding of logical formalisms, because it provides a finer-grained characterization of their expressive capabilities. For instance, in the area of relational databases, DL, AEL and DATALOG under the minimal model semantics express exactly the same class of queries (cf. [8]). However, in the area of bounded disjunctive databases, DL and AEL are strictly more expressive than DATALOG using minimal models, which cannot express $\mathbf{P}_R$ (or its complement in the cautious version); moreover, no (brave or cautious) preferential-based semantics for DATALOG (or any other first-order query) is capable of this. Similarly, perfect and stable model semantics do not express all the queries in PTIME.

Acknowledgements

The authors would like to thank Georg Gottlob for numerous discussions and for his important suggestions, which had a strong influence on this work, Mirek Truszczyński for his valuable hints and comments, and Victor Vianu for providing references to related work. This paper has been written while the first author was visiting the Institut für Informationssysteme, TU Wien, supported by the Austrian FWF through the Lise-Meitner Fellowship M0066-PHY.

References

1. Abiteboul, S., Kanellakis, P., Grahne, G.: On the Representation and Querying of Sets of Possible Worlds. Theoretical Computer Science **78** (1991) 159–187
2. Abiteboul, S., Vianu, V.: Expressive Power of Query Languages. In Ullman, J. (ed.) Theoretical Studies in Computer Science. Academic Press, 1992
3. Bonatti, P. A., Eiter, T.: Querying Disjunctive Databases Through Nonmonotonic Logics. Technical Report CD-TR 94/70, CD-Laboratory for Expert Systems, TU Vienna, Austria, 1994. Forthcoming. Preliminary version available from the authors.
4. Bonatti, P. A.: Autoepistemic Logics as a Unifying Framework for the Semantics of Logic Programs. Journal of Logic Programming (to appear). Preliminary version in Apt, K. (ed.) *Proc. of the Joint International Conference and Symposium on Logic Programming*, MIT Press, 1992
5. Cadoli, M., Eiter, T., Gottlob, G.: Using Default Logic as a Query Language. In Proc. Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR-94), 99–108, 1994
6. Chandra, A., Harel, D.: Horn Clause Queries and Generalizations. Journal of Logic Programming **2** (1985) 1–15
7. Chandra A. K.: Theory of Database Queries. In Proc. PODS-88, 1988
8. Eiter, T., Gottlob, G., Mannila, H.: Adding Disjunction to Datalog. In Proc. Thirteenth ACM SIGACT SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-94), 267–278, May 1994
9. Etherington, D. W.: Reasoning with Incomplete Information. Morgan Kaufmann Publishers, Inc., Los Altos, 1988
10. Fernández, J., Minker, J.: Semantics of Disjunctive Deductive Databases. In Proc. 4th Intl. Conference on Database Theory (ICDT-92), LNCS 646, 21–50, Springer-Verlag, 1992

11. Gottlob G.: Complexity Results for Nonmonotonic Logics. Journal of Logic and Computation **2** (1992) 397–425
12. Grahne G.: Updates and Counterfactuals. In Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning (KR-91), 269–276, 1991
13. Gurevich Y.: Logic and the Challenge of Computer Science. In E. Börger, editor, Trends in Theoretical Computer Science, Chapter 1. Computer Science Press, 1988
14. Kanellakis, P.: Elements of Relational Database Theory. In J. van Leeuwen (ed.) Handbook of Theoretical Computer Science, Volume B, Chapter 17. Elsevier Science Publishers B.V. (North-Holland), 1990
15. Kolaitis, P., Vardi, M.: On the Expressive Power of Datalog: Tools and a Case Study. In Proceedings PODS-90, 61–71, 1990
16. Lobo, J., Minker, J., Rajasekar, A.: Foundations of Disjunctive Logic Programming. MIT Press, Cambridge, MA, 1992
17. Lukasiewicz, W.: Non-Monotonic Reasoning. Ellis Horwood Limited, Chichester, England, 1990
18. Marek, W., Truszczyński, M.: Autoepistemic Logic. Journal of the ACM **38** (1991) 588–619
19. Marek, W., Truszczyński, M.: Nonmonotonic Logics – Context-Dependent Reasoning. Springer-Verlag, 1993
20. Moore R.: Semantical Considerations on Nonmonotonic Logics. Artificial Intelligence **25** (1985) 75–94
21. Przymusinski, T.: Stable Semantics for Disjunctive Programs. New Generation Computing **9** (1991) 401–424
22. Przymusinski, T.: On the Declarative and Procedural Semantics of Stratified Deductive Databases. in Minker, J. (ed.) Foundations of Deductive Databases and Logic Programming, 193–216. Morgan Kaufman, 1988
23. Reiter, R.: A Logic for Default Reasoning. Artificial Intelligence **13** (1980) 81–132
24. Schlipf, J.: The Expressive Powers of Logic Programming Semantics. Technical Report CIS-TR-90-3, Computer Science Department, University of Cincinnati, 1990. Preliminary version in Proc. PODS-90, 196–204. To appear in the Journal of Computer and System Sciences
25. Vardi, M.: Complexity of relational query languages. In Proceedings 14th STOC, 137–146, 1982
26. Vardi, M.: Querying Logical Databases. Journal of Computer and System Sciences **32** (1986) 142–160

This article was processed using the LaTeX macro package with LLNCS style