ELSEVIER Theoretical Computer Science 160 (1996) 321–363

Theoretical Computer Science

Querying disjunctive databases through nonmonotonic logics [1]

Piero A. Bonatti [a,*], Thomas Eiter [b]

[a] *Dipartimento di Informatica, Università di Torino, Corso Svizzera 185, 10149 Torino, Italy*
[b] *Christian Doppler Lab for Expert Systems, Institut für Informationssysteme, TU Wien, Paniglgasse 16, A-1040 Wien, Austria*

Received August 1994; revised August 1995
Communicated by H. Katsuno

Abstract

Query languages for retrieving information from disjunctive databases are an interesting open area of research. In this paper we study the expressive power of major nonmonotonic formalisms – such as circumscription, default logic, autoepistemic logic and some logic programming languages – 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 expressiveness of those languages and interesting fragments thereof. For instance, we show that there exist simple queries that cannot be expressed by any preferential semantics (including the minimal model semantics and the various forms of circumscription), while they 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. The latter result is proved by means of a new technique, based on a counting argument. Then we prove that under the assumption that the database consists of clauses whose length is bounded by some constant, default logic and autoepistemic logic express *all* of the Σ_2^p-recognizable Boolean queries, while preference-based logics cannot. These results hold for brave reasoning; we obtain dual results for cautious reasoning. Our results appear to be interesting both in the area of database theory and in the area of knowledge representation.

* Corresponding author. E-mail: bonatti@di.unito.it. This author was supported by the Austrian Government through the Lise-Meitner Fellowship M0066-PHY. This work has been carried out while he was visiting the Institut für Informationssysteme, TU Wien.

[1] An extended abstract of this paper appeared in the Proceedings of the 5th International Conference on Database Theory (ICDT '95), January 1995, Prague, pp. 68–81, LNCS 893, Springer, 1995.

SSDI 0304-3975(95)00192-1

1. Introduction

Disjunctive databases provide one of the major techniques for representing incomplete information – a topic which has been receiving much attention in the areas of databases, logic programming and AI [14, 26, 11, 9]. Disjunctive databases can be regarded as sets of ground clauses [2] each of which contains at least one positive literal; so, for instance, incomplete information such as "John suffers from disease d_1 or d_2 or d_3" or "Nicky is either a man or a woman" can be naturally encoded as follows:

$$disease(John, d_1) \vee disease(John, d_2) \vee disease(John, d_3), \tag{1}$$

$$man(Nicky) \vee woman(Nicky). \tag{2}$$

Closely related approaches to the representation of incomplete information can be found in [10, 40], where databases can be unrestricted propositional or first-order theories. In general, disjunctive databases are better suited for implementation; their operational and computational aspects have been extensively investigated (cf. [26, 11]).

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. [38, 20, 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 [2, 6]. The study of query languages for disjunctive databases is much less established. Work on querying incomplete databases is related; cf. [14, 1] for relations with null values, and [39] for the similar proviso of missing unique names axioms in logical databases. In these works, disjunctive information involving distinct relations – as in (2) – cannot be directly encoded. More importantly, these works consider only monotonic queries (in [39] the answer to a query grows monotonically with the amount of information encoded into the database; in [14, 1] queries are monotonic functions over sets of possible worlds). However, many natural queries do not easily fit into these frameworks, e.g.

Find all the persons which suffer from a disease that has not yet been precisely identified.

To see the nonmonotonic nature of this query, note that in a database consisting solely of (1), John would be part of the answer, while in a database containing also the fact $disease(John, d_l)$, John would not be included in the answer. [3]

---

[2] Strictly speaking, disjunctive databases may contain variables, but each database is given the same meaning as its ground instantiation, so the declarative semantics is not affected. In general, the ground instantiation of a database can be exponentially larger than the nonground version.

[3] More precisely, [14, 1] consider not only what is necessarily true in the result of a query (which returns an incomplete database), but also what is *possibly true*; this is a first step toward nonmonotonicity. For instance, the above nonmonotonic query can be answered by computing a trivial query (the identity function), and then solving a necessity problem (is the person definitely affected by some disease?) plus a set of possibility problems (for all the diseases d_i that may affect the patient, is it possible that the patient does *not* suffer from d_i?).

Advanced query languages for relational databases are typically nonmonotonic, 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 [35, 8, 5]. The use of nonmonotonic formalisms as query languages is justified by their expressive power, and by their ability of defining nonmonotonic queries, such as the one illustrated above; see [5] for further motivating examples. Moreover, by studying the limitations of general nonmonotonic logics such as default logic (DL) and autoepistemic logic (AEL), one can immediately derive negative results for many (perhaps more realistic) query languages – such as Datalog and its extensions with negation, disjunction and nondeterministic constructs – which can be regarded as fragments of DL and AEL.

Informally, a query is defined by an expression (the "program") which is evaluated against the current database (the "data"). We point out that a similar situation is encountered in knowledge representation; a knowledge base can be regarded as a pair of subtheories that specify fixed background knowledge (the "program") and a varying specific situation (the "data"). In view of this analogy, knowledge representation languages are promising starting points for the definition of query languages and can themselves be interesting such languages. On the other hand, the analogy shows that expressiveness analysis is relevant to knowledge representation; it allows to determine which properties of the current situation can be recognised through the fixed background knowledge, and hence – indirectly – which behaviours can be defined through knowledge representation languages. Of course, incomplete information is central also in this framework; in realistic situations the knowledge about the current situation will most likely be incomplete.

In this paper we will study the expressive power of the major nonmonotonic formalisms used as query languages over disjunctive databases. We will pay particular attention to *positive* disjunctive databases – which are essentially sets of positive clauses – because they enjoy several nice properties that make them interesting from theoretical as well as practical points of view:

- First of all, a ground clause C is entailed by a positive database iff C is subsumed by some of the database's clauses. As a consequence, the answer to an arbitrary query in CNF can be computed in polynomial time (for unrestricted disjunctive databases, the same entailment problem would be coNP-hard). Deduction collapses to subsumption based *retrival*, therefore, CNF query processing can be implemented efficiently by means of well established database access techniques, combined with a specialized algorithm for subsumption checking (exploiting the fact that database clauses are ground).
- Despite their simplicity, positive databases can encode many partially specified facts which can be encountered in real situations. Note that (1) and (2) can be part of a positive database.

However, no such combinations of problems are considered in [14, 1]. Logically speaking, it seems more natural to regard this kind of reasoning as part of the query evaluation, and not as a post-processing of the result of a query.

- The modelling power of positive databases is minimal, in some sense; basically, they can encode facts (ground atoms) and null values with simple range restrictions (e.g. (1) encodes a null value whose range is restricted to $\{d_1, d_2, d_3\}$). Any nontrivial database system admitting incomplete information should have at least the same capabilities. (Strictly speaking, positive databases can also encode facts involving multiple relations, but this does not affect complexity nor expressiveness results.)

In the light of the above points, it is interesting to see how positive databases can be queried and, in particular, it would be interesting to find a language that captures exactly the queries computable in polynomial time. Our results do not depend on the restriction to positive clauses, so we shall consider also general disjunctive rules. 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. [11]), default logic [34] and autoepistemic logic [30] cannot express all of the PTIME-recognizable properties of positive disjunctive databases; this result is surprising, because in the framework of relational databases these logics can express every Σ_2^p-recognizable query. [4] More surprisingly, the same negative result holds for unrestricted preferential semantics, some of which are tremendously powerful (not even recursively enumerable). We obtain similar results for some of the major semantics of disjunctive logic programs, such as the stable model semantics and the perfect model semantics, cf. [11]. This kind of negative results is usually proved by means of persistency properties, cf. [22]; 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.

The paper is structured as follows. Section 2 states preliminaries on disjunctive databases, and gives a brief summary of the necessary concepts from nonmonotonic logics. In Section 3, we give an abstract, formal definition of query languages over disjunctive databases; in Section 4, we consider concrete such query languages defined from nonmonotonic logics, in particular from preferential semantics, default logic, and

[4] Recall that $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$ is the class of problems decidable in nondeterministic polynomial time with an oracle in NP.

autoepistemic logic. The subsequent Sections 5–7 are devoted to analyse the expressive power of those query languages under brave semantics. In Section 8, we deal with cautious semantics. The paper is concluded by Section 9, where we discuss the results and outline issues for future work. To improve the readability of the paper, the proofs of some results have been moved to the appendix.

2. Preliminaries

2.1. Disjunctive databases

A *relational scheme* over a domain *Dom* is a list $\overline{R} = R_1,\dots,R_z$ of relation (or predicate) symbols R_i of arity $a_i \geqslant 0$ ($i = 1,\dots,z$). We assume that *Dom*, which will not be explicitely 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 (possibly empty) conjunction of distinct first-order formulas of the form

$$A_1 \vee \dots \vee A_l \leftarrow B_1 \wedge \dots \wedge B_m, \qquad l > 0, m \geqslant 0, \tag{3}$$

where $A_1,\dots,A_l,B_1,\dots,B_m$ are ground atoms, built from the predicate symbols in $\overline{R}$ and from the constants in U.[5] Empty databases are represented by the empty conjunction, which is always true; the empty conjunction will be denoted by $\top$. The universe of D is denoted by $U(D)$. The set of disjunctive databases over $\overline{R}$ is denoted by $\mathscr{D}(\overline{R})$. The formulas (3) are called *disjunctive rules*. A disjunctive rule is *positive* if $m = 0$;[6] a disjunctive database (U,ϕ) is *positive* if ϕ is a conjunction of positive rules.

The *dimension* of a clause C, $dim(C)$, is the number of its disjuncts. For any integer $d \geqslant 1$, we denote by $\mathscr{D}(\overline{R})_{\leqslant d}$ the set of databases (U,ϕ) from $\mathscr{D}(\overline{R})$ such that for each clause C occurring in ϕ, $dim(C) \leqslant d$. Notice that $\mathscr{D}(\overline{R})_{\leqslant 1}$ corresponds in an obvious way to the class of relational databases over $\overline{R}$, i.e., the finite structures $\langle U, r_1^{a_1},\dots,r_z^{a_z}\rangle$ where $r_i^{a_i} \subseteq U^{a_i}$; the correspondence is unique modulo the ordering of the atoms in the formula ϕ of the database.

2.2. Normal disjunctive databases

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 $C_1 \wedge \dots \wedge C_n$ is *redundant* if it is subsumed by some C_j where $j \neq i$.

[5] Note that U suffices to specify the language of ϕ, for the set of predicate symbols is fixed by the relational scheme. Therefore, a disjunctive database is essentially a theory, i.e. a pair ⟨*Language*, *Axioms*⟩ (cf. [27]).

[6] In the literature on logic programming a positive clause is usually a rule that has no negative literals in the body. Here we regard rules as alternative representations for clauses, so we call a rule with $m = 0$ *positive* because it corresponds to a positive clause.

Definition 2.1. 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 2.2. *For all positive databases $D = (U, \phi)$:*

(a) *for all nontautological clauses C, $\phi \models C$ holds iff C is subsumed by some conjunct of ϕ 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 the ordering of clauses and disjuncts).*

2.3. Default logic

Default logic [34] is one of the best known formalizations of nonmonotonic reasoning (see [27, 29] for extensive studies of the subject). A *default* $\delta(\mathbf{x})$ is a rule of the form

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

(also written $(\alpha(\mathbf{x}) : \beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x})/\gamma(\mathbf{x}))$) such that $\alpha(\mathbf{x}), \beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x}), \gamma(\mathbf{x})$ are first-order formulas whose free variables are among those of $\mathbf{x} = x_1, \ldots, x_m$; $\alpha(\mathbf{x})$ is called the *prerequisite* of the default, $\beta_1(\mathbf{x}), \ldots, \beta_n(\mathbf{x})$ $(n \geqslant 0)$ are called *justifications* and $\gamma(\mathbf{x})$ is the *consequent*. When $n = 0$, then the empty conjunction $\top$ (truth) is implicitly assumed as the justification of the default. For convenience, 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.

A default theory T is a pair $\langle W, \Delta \rangle$ where W is a set of first-order formulas and Δ is a set of default rules; it is finite if both W and Δ are finite. In this paper, we focus on finite default theories. The default theory $T = \langle W, \Delta \rangle$ is *closed* iff all the defaults in Δ are closed, and *open* otherwise.[7]

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

$$E_0 = W, \text{ and for } i \geqslant 0$$

$$E_{i+1} = \mathrm{Th}(E_i) \cup \{ \gamma \mid (\alpha : \beta_1, \ldots, \beta_n/\gamma) \in \Delta,\ \alpha \in E_i,\ \forall j. \neg\beta_j \notin E, \}$$

[7] The semantics of open default theories involves technical complications that are not needed in this paper; the interested reader is referred to [27].

where $\mathrm{Th}(\cdot)$ denotes classical deductive closure. Then, E is an extension of T iff $E = \bigcup_{i=0}^{\infty} E_i$. Notice that E is deductively closed, and hence an infinite object.

Each extension of T can be constructed from its generating defaults. Let S be a set of formulas, and define

$$\mathrm{GD}(\Delta, S) = \{\, \delta \in \Delta \mid \delta = (\alpha : \beta_1, \ldots, \beta_n / \gamma),\ S \vdash \alpha,\ \forall j. S \nvdash \neg\beta_j \,\};$$

we call the defaults in $\mathrm{GD}(\Delta, S)$ the *generating defaults* of S w.r.t. Δ.[8] Furthermore, for every set of defaults Δ, denote by $\mathrm{CONS}(\Delta)$ the set of the consequents of the defaults in Δ. Then,

Lemma 2.3 (Reiter [34, Theorem 2.5]). *Let E be an extension of the closed default theory $T = \langle W, \Delta \rangle$. Then, $E = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E)))$.*

The converse is not true in general. However, from the above quasi-inductive characterization of extensions, one can easily obtain the following result, in the spirit of Marek and Truszczyński's characterizations based on well-orderings (cf. [29]).

Lemma 2.4. *Let $T = \langle W, \Delta \rangle$ be a finite closed default theory and let E be a set of sentences. Then, E is an extension of T iff*

(a) $E = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E)))$; *and*

(b) *there exists a strict partial order*[9] $\prec$ *on* $\mathrm{GD}(\Delta, E)$ *such that for every* $\delta = (\alpha : \beta_1, \ldots, \beta_n / \gamma) \in \mathrm{GD}(\Delta, E)$, *it holds that*

$$\alpha \in \mathrm{Th}(W \cup \mathrm{CONS}(\{\, \delta' \in \mathrm{GD}(\Delta, E) \mid \delta' \prec \delta \,\})).$$

Corollary 2.5 (cf. Theorems 3.80, 3.81 in Marek and Truszczyński [29, pp. 89, 91]). *Let $\langle W, \Delta \rangle$ be a closed prerequisite-free theory. Then, E is an extension of $\langle W, \Delta \rangle$ iff $E = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E)))$.*

In general, a default theory can have one, multiple or no extensions. Therefore, defining entailment of a formula ϕ from a default theory $\langle D, W \rangle$ is not straightforward. The standard variants are *brave entailment*, under which ϕ is entailed if ϕ belongs to some extension of T and *cautious entailment*, under which ϕ follows if ϕ belongs to every extension of T. From the computational side, credulous and skeptical reasoning have been extensively studied in the literature [21, 13, 36].

2.4. Autoepistemic logic (AEL)

The language of autoepistemic logic [30] is a modal language with one modal operator L, to be read as "know" or "believe"; for an extensive treatment, cf. [28, 27].

[8] Reiter's notion of generating default [34] is defined only when S is an extension. Here we adopt the more general definition introduced by Marek and Trusczyński [29], which applies to arbitrary contexts S; intuitively, their notion of generating defaults captures all the defaults that are "applicable" in S.

[9] By strict partial order we mean an irreflexive and transitive binary relation.

The formulas where L does not occur are called *ordinary* or *objective*. The *kernel* of a set of formulas S, denoted by S_0, is the set of all the objective formulas of S. The formulas where predicate symbols occur only within the scope of L are called *subjective*. The formulas of the form $L\psi$ (where ψ is an arbitrarily complex formula) are called *autoepistemic atoms*.

An autoepistemic theory is a set of autoepistemic formulas. An autoepistemic theory is *closed* if all of its members are closed, and *open* otherwise.

Moore's original formulation tackles only the propositional case; extending it to full quantification involves subtle technical difficulties due to the behaviour of quantification through the modal operator; the first proposals in this direction are due to Konolige [24] and Levesque [25]. The complications of quantifying-in will not be tackled in this paper. We will restrict our attention to quantifier-free autoepistemic theories; they will be treated as sets of schemata to be instantiated in all possible ways. Thus, entailment needs to be defined only for closed quantifier-free theories, that can be regarded as propositional theories. Consequently, Moore's setting can be applied without modifications.

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:

$$S = \{\, \phi \mid T \cup LS \cup \neg L\overline{S} \vdash \phi \,\}$$

where

$$LS = \{\, L\psi \mid \psi \in S \,\} \quad \text{and} \quad \neg L\overline{S} = \{\, \neg L\psi \mid \psi \notin S \,\}.$$

Here $\vdash$ denotes classical derivability; no modal axioms are employed; autoepistemic atoms are treated as ordinary atoms. Accordingly, autoepistemic sentences are interpreted through *propositional interpretations*, which are mappings that assign classical truth values to ordinary atoms *and* autoepistemic atoms. Standard logical connectives are interpreted in the usual way. The notion of *model* is extended to propositional interpretations and autoepistemic sentences in the obvious way.

In general, an autoepistemic theory T may have one, multiple or no stable expansions, excepting objective theories, which always have a unique stable expansion denoted by $\mathrm{E}(T)$. Several forms of entailment can be defined (cf. entailment for default logic): *brave entailment*, under which ϕ is entailed if ϕ belongs to some stable expansion of T and *cautious entailment*, under which ϕ follows if ϕ belongs to every stable expansion of T.

2.5. Normal autoepistemic theories

An autoepistemic theory T is in *autoepistemic normal form* when it contains only sentences of the form

$$L\alpha_1 \wedge \cdots \wedge L\alpha_m \wedge \neg L\beta_1 \wedge \cdots \wedge \neg L\beta_n \rightarrow \phi \tag{4}$$

where $m, n \geqslant 0$ and $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_n, \phi$ are objective, cf. [23, 28]. Konolige proved that every closed autoepistemic theory is equivalent to a closed theory in autoepistemic normal form, in the sense that the two theories have the same stable expansions.

We can easily extend the *epistemic Gelfond–Lifschitz transformation* [3] to autoepistemic theories.

Definition 2.6. For all normal autoepistemic theories T and all sets of autoepistemic sentences S define (cf. Definition 11.3 of [29, p. 322])

$$\begin{aligned} T^S = \{\phi \mid\ & (L\alpha_1 \wedge \cdots \wedge L\alpha_m \wedge \neg L\beta_1 \wedge \cdots \wedge \neg L\beta_n \rightarrow \phi) \in T, \\ & \{\alpha_1, \ldots, \alpha_m\} \subseteq S, \\ & \{\beta_1, \ldots, \beta_n\} \cap S = \emptyset\} \end{aligned}$$

Note that $\bigwedge T$ and $\bigwedge T^S$ are given identical truth values in every model of $LS \cup \neg L\overline{S}$, and that $T^S = T^{S_0}$; thus, we get the key property of the above transformation.

Lemma 2.7. *For all normal autoepistemic theories T and all sets of autoepistemic sentences S*

$$T \cup LS \cup \neg L\overline{S} \equiv T^{S_0} \cup LS \cup \neg L\overline{S}$$

Through the epistemic Gelfond–Lifschitz transformation – which yields always an objective theory – we can determine the kernels of stable expansions in a nonmodal setting.[10] (Under different notation, this characterization appears in [28]; cf. also Section 11 in [29].)

Theorem 2.8. *For all normal autoepistemic theories T:*

(i) *If E is a stable expansion of T then $E_0 = \mathrm{Th}(T^{E_0})$ (where E_0 is the kernel of E)*

(ii) *Conversely, if $E = \mathrm{Th}(T^E)$ then $\mathrm{E}(E)$ (the unique stable expansion of E) is a stable expansion of T.*

Proof. (i) Assume that E is a stable expansion of T and let ψ be any objective sentence. We have $\psi \in E$ iff $T \cup LE \cup \neg L\overline{E} \vdash \psi$. Then, by Lemma 2.7, $\psi \in E$ iff $T^{E_0} \cup LE \cup \neg L\overline{E} \vdash \psi$. Moreover, the last entailment holds iff $T^{E_0} \vdash \psi$, because $LE \cup \neg L\overline{E}$ has no (autoepistemic) atoms in common with T^{E_0} and ψ. Conclude that $\psi \in E$ iff $\psi \in \mathrm{Th}(T^{E_0})$, which proves (i).

(ii) Assume that $E = \mathrm{Th}(T^E)$ and let $E' = \mathrm{E}(E)$. Then derive:

$$\begin{aligned} E' &\equiv E \cup LE' \cup \neg L\overline{E'} \\ &= \mathrm{Th}(T^E) \cup LE' \cup \neg L\overline{E'} \end{aligned}$$

[10] In the following by $\mathrm{Th}(X)$ we mean the theorems of X in first-order nonmodal logic; $\vdash$ will denote classical derivability in the modal language, instead.

$$\equiv T^E \cup LE' \cup \neg\overline{LE'}$$
$$\equiv T \cup LE' \cup \neg\overline{LE'} \quad \text{(by Lemma 2.7)}$$

Thus, by the definition of stable expansion, E' is a stable expansion of T. □

2.6. Preferential semantics and preference logics

Some of the most popular semantics for nonmonotonic reasoning, such as the minimal model semantics and the various forms of circumscription (see [27]), are based on preference relations over interpretations; different notions of preference lead to different preference logics. A preference relation $\prec$ is typically a strict partial order, i.e. an irreflexive and transitive relation; intuitively, $I \prec J$ means that the interpretation I is preferable to J. Usually, a sentence ϕ can be derived from a theory T if ϕ is true in all the $\prec$-minimal (i.e., most preferred) models of T. This definition corresponds to cautious reasoning; we will also consider brave reasoning, although it may seem less natural in the preferential framework. A sentence ϕ can be derived from T under brave reasoning if ϕ is true in at least one $\prec$-minimal model of T.

One of the most popular preference logics is *circumscription*, which is essentially a generalization of the minimal model semantics. The simplest form of circumscription is based on a preference relation $<^P$, where P is a set of predicates whose extension should be minimized. An interpretation I_1 is preferred to I_2 (i.e., $I_1 <^P I_2$) iff the following conditions hold:

1. I_1 and I_2 have the same domain;
2. they agree on the interpretation of constants and function symbols;
3. for all predicates $p \in P$, the extension of p in I_1 is smaller than or equal to the extension of p in I_2; moreover, for some $p \in P$, the extension of p in I_1 is strictly smaller than the extension of p in I_2.

Example 2.9. Let $P = \{p\}$, $T = \{p(a), \forall x.\neg p(x) \rightarrow q(x)\}$. The $<^P$-minimal models of T are those where $p(y)$ is false for all y excepting $y = a$. Therefore, through cautious reasoning, one can derive: $\forall x.x \neq a \rightarrow \neg p(x)$ and $\forall x.x \neq a \rightarrow q(x)$.

The $<^P$-minimal models of T whose domain is a singleton satisfy: $\forall x.p(x)$; therefore, this sentence can be derived by means of brave reasoning (but not through cautious reasoning). □

The interested reader is referred to [27] for more details.

3. Query languages for disjunctive databases

Query languages for relational databases are a well-studied topic in the theory of relational databases, cf. [38, 7, 20, 2]. Informally, a query is a function that maps each relational database onto another relational database, which is considered as the result of evaluating the query. Queries are specified by the expressions of query languages; expressions can be regarded as "programs" for computing queries. This setting 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 $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(\overline{S})$ 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 $IO(q)$; we will always assume that $\overline{R}$ and $\overline{S}$ are disjoint.

A *query language* is constituted of a set $\mathscr{L}$ of query expressions and a function μ such that for each expression $E \in \mathscr{L}$, $\mu(E)$ is a query (cf. [38]).

The queries which compute relations, i.e., $q(D) \in \mathscr{D}(\overline{S})_{\leqslant 1}$ 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 $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$ is denoted by $\mathbf{P}_q$. For every $D \in \mathscr{D}(\overline{R})$, we have $\mathbf{P}_q(D) = true$ iff $q(D) = (U(D), S)$ and $\mathbf{P}_q(D) = false$ iff $q(D) = (U(D), \top)$. We will denote by $\overline{\mathbf{P}}$ the property complementary to $\mathbf{P}$, i.e.,

$$\overline{\mathbf{P}}(D) = true \text{ iff } \mathbf{P}(D) = false$$

The complexity of a query language $\mathscr{L}$ is measured in terms of the resources needed by the Turing machines that compute the queries expressed by $\mathscr{L}$. More precisely, for all queries q expressible in $\mathscr{L}$, one considers the Turing machines that compute (an encoding of) $q(D)$ given any (encoded) database D as an input. This view of complexity is called data complexity [38].

We adopt an encoding $enc(\cdot)$ of disjunctive databases such that $enc(U, \phi)$ represents the relation symbols and the elements of U by numbers; the encoding lists the clauses of D according to some fixed ordering; U is described by its cardinality written down in unary notation.[11]

Let $\mathbf{C}$ be any complexity class (based on the Turing machine model). Following the approach in [38, 17], 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* $\mathbf{C}$)

[11] Equivalently w.r.t. to polynomial-time computability, we might represent U by enumerating its elements, cf. [17]. Note that the standard encoding of relational databases needs always an amount of space which is larger than the cardinality of U.

if, given D and a ground clause C, deciding whether $C \in q(D)$ is in **C**. **P** (resp. q) is called *hard* for **C**, if the above decision problem is hard for **C** under polynomial-time transformations.

Notice that in the general case, the size of $enc(q(D))$ can be exponential in the size of $enc(D)$; hence, a PTIME-recognizable query q may not be computable in polynomial time. To avoid this, a notion of complexity can be adopted under which efficiency is relativized to the size of the output (besides that of the input), and a computation is feasible if it is polynomial in the combined size of the input and the output, cf. [37, 19]. Our framework might be modified in a similar way.

In the case of queries with bounded output databases, i.e., where $q(D) \in \mathscr{D}(\overline{S})_{\leqslant d}$ for some constant d, and in particular for Boolean queries and queries computing relations, the size of $enc(q(D))$ is always polynomial in the size of $enc(D)$; moreover, if q is **C**-recognizable, then $enc(q(D))$ can be constructed in polynomial time with an oracle for **C**.

The *expressive power* of a query language $\mathscr{L}$, $\mathscr{E}(\mathscr{L})$, is the class of queries that are definable in $\mathscr{L}$, i.e., $\mathscr{E}(\mathscr{L}) = \{ q \mid \exists E \in \mathscr{L} : \mu(E) = q \}$.

A query language $\mathscr{L}$ is *well-balanced* if its expressiveness matches its complexity, i.e., it can precisely express the database properties (resp. queries) computable within the bounds of some complexity class **C**, cf. [18, 17, 2]; we say that $\mathscr{L}$ *captures* the class of **C**-recognizable database properties (resp. queries) in this case.

We remark that a **C**-complete $\mathscr{L}$ does not necessarily capture **C** (cf. below and [2]).

4. Nonmonotonic logics as query languages

In this section we define the semantics of nonmonotonic logics (in particular, default logic and AEL) as a query language for Boolean database properties over disjunctive databases; a generalization to relational-output queries is straightforward.

We take inspiration from DATALOG (cf. [20]). There, query expressions (which are finite collections of function-free Horn clauses) are “evaluated against a relational database”. This means that the query expression T (the “program”) is extended with the set of atoms D (the “data”) which are true in the relational database, and then the least Herbrand model M of the resulting logic program is computed. The answer to the query is “yes” if the output propositional letter P is true in M. Equivalently, the answer is “yes” if P is logically entailed by $T \cup D$.

We will extend this approach to disjunctive databases and nonmonotonic logics. Roughly speaking, the answer to the query expressed by T in a logic L evaluated against a disjunctive database $D = (U, \phi)$ is “yes” iff the output letter P is entailed in L by the union (or a similar combination) of T and ϕ.

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, which are obtained by substituting the elements of U for free variables in all possible ways.

4.1. Preference logics as query languages

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

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

Intuitively, $+$ expresses the combination of the query expressions and of the database. The contents of the database (i.e., ϕ) are added to the query program T through set-theoretic union, as in Datalog. Furthermore, we instantiate T with the elements of U in order to cover different forms of universal quantification; free variables are treated as universally quantified variables under substitutional quantification – which is adopted in several logic programming and knowledge representation languages [33] – while the variables which are universally quantified explicitly are interpreted as usual.

Definition 4.1. 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.[12] Then, T expresses the Boolean query $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$ under the brave preferential semantics iff for all $D \in \mathscr{D}(\overline{R})$,

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

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

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

Example 4.2. Consider the query: *Is John affected by some disease?* Under the cautious minimal model semantics, this query is expressed by the simple theory

$$T = \{ S \leftarrow \exists x.\, \mathit{disease}(\mathit{John}, x) \}.$$

In fact, S belongs to every minimal model of $T + D$ iff $D \models \exists x.\, \mathit{disease}(\mathit{John}, x)$.

Note that each database property $\mathbf{P}$ over $\mathscr{D}(\overline{R})$ corresponds to some Boolean query $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$ such that $\mathbf{P}_q = \mathbf{P}$. We will say that T expresses $\mathbf{P}$ under brave (resp. cautious) preferential semantics iff T expresses q under brave (resp. cautious) preferential semantics.

4.2. Default logic as a query language

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 $\overline{R}$. The restriction to a function

[12] We adopt here the notion of *preferential semantics* which has been introduced in Shoham's book *Reasoning About Change*, MIT Press, 1988. Note that the preference relation must be fixed in advance (and independent of the syntactical form of T). Intermediate approaches – such as the stable semantics of disjunctive logic programs [32] and the perfect model semantics [31] – do not fit into this framework.

and quantifier-free first-order language is rather common in database theory (cf. the query language Datalog). Here we need it to achieve decidability.

By $[\Delta]_U$ we denote the set of all ground instances of defaults from Δ obtained by replacing free variables with elements of U in all possible ways. Finally, for all disjunctive databases $D = (U, \phi)$, define

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

The intuition underlying this definition is essentially the same as in the previous section. The only difference is due to the fact that a default theory is not a set of formulae but a pair of sets, and hence ϕ has to be added to the appropriate element of the pair.

Definition 4.3. Let $T = \langle W, \Delta \rangle$ be a default theory. Then, T *expresses* the Boolean query $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$ under the *brave* semantics iff for all databases $D \in \mathscr{D}(R)$,

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

T expresses the Boolean query $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$ over $\mathscr{D}(\overline{R})$ under the *cautious* semantics iff for all databases $D \in \mathscr{D}(R)$,

$\mathbf{P}_q(D) = \mathit{true}$ iff S belongs to every extension of $T + D$.

As we did for preferential semantics, we might say that a default theory T expresses a database property $\mathbf{P}$ iff T expresses the corresponding Boolean query q. However, it will be convenient–and probably more appealing–to adopt the following definition.

Definition 4.4. T expresses $\mathbf{P}$ over $\mathscr{D}(\overline{R})$ under the *brave* semantics iff for all databases $D \in \mathscr{D}(R)$,

$\mathbf{P}(D) = \mathit{true}$ iff $T + D$ has an extension.

This definition is simpler to handle, and it does not change the expressive power of brave default reasoning. In fact, given a theory T that expresses a Boolean query $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$, we can obtain a theory T' that expresses $\mathbf{P}_q$ under Definition 4.4 by extending T with the default $(: \neg S / \bot)$, where $\bot$ is any unsatisfiable sentence.

Conversely, given a theory T' (in which S does not occur) that expresses $\mathbf{P}$ under Definition 4.4, we can obtain a theory T that expresses the Boolean query q by extending T' with the simple axiom S.

Example 4.5. Consider the query:

Is there a patient whose disease has not yet been precisely identified and belongs to the set $\{d_1, \ldots, d_n\}$?

This query can be expressed by means of a default theory $T = \langle \emptyset, \{\delta\} \rangle$, consisting of a single default,

$$\delta = \frac{\mathit{disease}(x, d_1) \vee \cdots \vee \mathit{disease}(x, d_n) : \neg\mathit{disease}(x, d_1), \cdots, \neg\mathit{disease}(x, d_n)}{S} .$$

The prerequisite of δ checks that the patient suffers from one of the diseases $d_1, \ldots, d_n$. The justifications verify that no such disease has been unequivocally diagnosed, i.e., that none of the facts $disease(x, d_i)$ is entailed by the database. For this particular T, the theory $T + D$ has always a unique extension, therefore brave and cautious semantics coincide. For each database D, the answer to the above query is "yes" iff the extension of $T + D$ contains S (note that S can be derived only by means of δ, due to the assumption that the input and output relational schemes of a query have no symbols in common). The database property corresponding to the above query is

> *There is a patient whose disease has not yet been precisely identified and belongs to the set* $\{ d_1, \ldots, d_n \}$.

This property is captured by the default theory $T' = \langle \emptyset, \{ \delta, \delta' \} \rangle$, where δ is the above default and $\delta' = (: \neg S / \bot)$. A database D has the above property iff $T' + D$ has an extension (which happens iff δ is applicable and its conclusion, S, blocks δ').

4.3. Autoepistemic logic as a query language

In a similar way, we define the use of AEL as a query language.

Definition 4.6. Let T be an autoepistemic theory, whose underlying first-order language is function-free and quantifier-free and includes the predicates from $\overline{R}$. For every disjunctive database $D = (U, \phi)$, define

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

The autoepistemic theory T *expresses* the Boolean query $q : \mathscr{D}(\overline{R}) \to \mathscr{D}(S)$ under the *brave* semantics iff for all databases $D \in \mathscr{D}(R)$,

$$\mathbf{P}_q(D) = \mathit{true} \text{ iff } S \text{ belongs to at least one stable expansion of } T + D.$$

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

$$\mathbf{P}_q(D) = \mathit{true} \text{ iff } S \text{ belongs to every stable expansion of } T + D.$$

We will adopt the following definition for the properties expressed through the brave semantics (cf. previous paragraph):

T expresses $\mathbf{P}$ over $\mathscr{D}(\overline{R})$ under the *brave* semantics iff for all databases $D \in \mathscr{D}(R)$,

$$\mathbf{P}(D) = \mathit{true} \text{ iff } T + D \text{ has a stable expansion.}$$

Example 4.7. Consider the query illustrated in Example 4.5. The same query can be expressed in AEL by means of the theory T consisting of the unique axiom

$$L(disease(x, d_1) \vee \cdots \vee disease(x, d_n)) \wedge \neg L\, disease(x, d_1) \wedge \cdots \wedge \neg L\, disease(x, d_1) \to S.$$

Note the similarity between this axiom and the default δ in Example 4.5. In this particular case, $T+D$ has always a unique stable expansion, therefore brave and cautious semantics coincide. The corresponding database property can be expressed by extending T with the axiom $\neg LS \to S$. The new axiom plays the same role as δ' in Example 4.5; when S cannot be derived by means of the first axiom, the second one generates an instability which causes $T+D$ to have no stable expansions.

Remark 4.8. Both in default and autoepistemic logic, brave and cautious semantics are complementary, in the sense that given a theory T which expresses a property $\mathbf{P}$ under the brave semantics, we can always find a theory T' which expresses $\overline{\mathbf{P}}$ under the cautious semantics, and vice versa (cf. Section 8).

5. Expressive power of preferential semantics

It has been recently shown that the brave version of parallel circumscription captures the Σ_2^p-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 preferential semantics have the same limitations.* For this purpose, consider the following property.

Definition 5.1. 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 $U(D)^a$: $\phi \models R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)$ and $\phi \not\models R(\mathbf{c}_i)$ $(i = 1, \ldots, n)$.

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 $R(c, v_1) \vee \cdots \vee R(c, v_n)$. From this point of view, a database (U, ϕ) has the property $\mathbf{P}_R$ iff D contains a proper null value.

Theorem 5.2. *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 5.3. *Let $\prec$ be any binary relation over the set of interpretations, and for all sets of sentences S, let* MM(S) *denote the set of $\prec$-minimal models of S. For all sets of sentences $S, \{\sigma_1, \ldots, \sigma_n\}$ we have*

$$\mathrm{MM}(S \cup \{\sigma_1 \vee \cdots \vee \sigma_n\}) \subseteq \mathrm{MM}(S \cup \{\sigma_1\}) \cup \cdots \cup \mathrm{MM}(S \cup \{\sigma_n\}).$$

Proof. Suppose not, and let M be some model in $\mathrm{MM}(S \cup \{\sigma_1 \vee \cdots \vee \sigma_n\}) \setminus [\mathrm{MM}(S \cup \{\sigma_1\}) \cup \cdots \cup \mathrm{MM}(S \cup \{\sigma_n\})]$; we will derive a contradiction. To satisfy $S \cup \{\sigma_1 \vee \cdots \vee \sigma_n\}$,

M must be a model of $S \cup \{\sigma_i\}$, for some $i \in \{1,\dots,n\}$. By assumption, $M \notin \mathrm{MM}(S \cup \{\sigma_i\})$, therefore there must be a model M' of $S \cup \{\sigma_i\}$ such that $M' \prec M$. But M' is also a model of $S \cup \{\sigma_1 \vee \dots \vee \sigma_n\}$, therefore M cannot be a minimal model of $S \cup \{\sigma_1 \vee \dots \vee \sigma_n\}$; a contradiction. □

Proof of Theorem 5.2. 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 5.3, 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. □

A similar result can be obtained for a somewhat related property of theoretical interest:

Definition 5.4. For all $D = (U, \phi)$ define $\mathbf{P}_{\mathrm{MM}}(D) = \mathit{true}$ iff ϕ has at least two distinct minimal models.

In other words, D has the property $\mathbf{P}_{\mathrm{MM}}$ iff D is not equivalent to a relational database

Theorem 5.5. *The property $\mathbf{P}_{\mathrm{MM}}$ cannot be expressed by any first-order theory under any brave preferential semantics.*

Proof. Similar to the proof of Theorem 5.2. □

Note that $\mathbf{P}_R(D)$ and $\mathbf{P}_{\mathrm{MM}}(D)$ can be computed in polynomial time when D is positive (hint: use Proposition 2.2(i) to design an algorithm which solves the problem in quadratic time). As a consequence:

Corollary 5.6. *There exist properties of positive databases that are computable in polynomial time, but cannot be expressed under any brave preferential semantics.*

The same result holds for cautious preferential semantics. This can be easily derived from the database property complementary to $\mathbf{P}_R$, i.e., the property of having no null values.

In particular, we have that none of the various forms of circumscription (including parallel circumscription, possibly with fixed and varying predicates, and prioritized circumscription) can express all the polynomial-time recognizable properties of positive disjunctive databases.

It is interesting to note that some preferential semantics are highly complex; the preference relation may encode lots of information, thereby causing brave reasoning to be hard for arbitrary complexity classes; some theories are not even semi-decidable; but despite their tremendous power, these formalisms cannot express simple properties such as $\mathbf{P}_R$ and $\mathbf{P}_{\mathrm{MM}}$.

Remark 5.7. The negative results for preferential semantics can be immediately extended to first- and second-order logic. In fact, (i) monotonic logics are captured by the empty preference relation, and (ii) the proofs of Theorems 5.2 and 5.5 go through whenever the language contains classical disjunction (i.e., a connective that satisfies the usual truth-recursive rules).

The above results can be extended also to the *stable model semantics* (see [32]) – which is 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.

Theorem 5.8. *The property* $\mathbf{P}_R$ *cannot be expressed by any disjunctive logic program under the brave stable semantics.*

The proof can be found in Appendix A.

Remark 5.9. The same negative result holds for disjunctive logic programs under the *perfect model semantics* [31] (which is, like stable model semantics, not a preferential semantics according to the concept that we follow here; the proof of the negative result is analogous to the proof for stable model semantics) and for *extended disjunctive logic programs*, under their standard answer set semantics (see [12]). The latter can be easily embedded into (nonextended) disjunctive logic programs under the stable semantics, by replacing each negative literal $\neg A$ with a new atom A^*. Thus, Theorem 5.8 holds for extended disjunctive logic programs, too.

Similar results have already been proved for other important semantics of disjunctive logic programs, cf. [4].

6. Expressive power of default logic

Default logic can express the property $\mathbf{P}_R$, which cannot be expressed by any preference logic.

The default theory $T_R = \langle W_R, \Delta_R \rangle$, whose formulas and defaults are illustrated in Fig. 1, is suitable for this purpose. The predicates C and S are new and do not occur among the database predicates $\overline{R}$. Intuitively, predicate C describes a clause $R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)$; $C(\mathbf{x})$ means that $R(\mathbf{x})$ occurs in the clause. The clause is "guessed" by the defaults (D1)–(D2); the default (D3) assures that the clause is truly disjunctive, i.e., it contains no $R(\mathbf{c}_i)$ such that $\phi \models R(\mathbf{c}_i)$. The formula (F) and the default (D4)

$$\text{(F)}\; C(\mathbf{x}) \wedge R(\mathbf{x}) \rightarrow S$$

$$\text{(D1)}\ \frac{:C(\mathbf{x})}{C(\mathbf{x})} \qquad \text{(D2)}\ \frac{:\neg C(\mathbf{x})}{\neg C(\mathbf{x})} \qquad \text{(D3)}\ \frac{C(\mathbf{x}) \wedge R(\mathbf{x}):}{\perp} \qquad \text{(D4)}\ \frac{:\neg S}{\perp}$$

Fig. 1. Formula (F) and defaults (D1)–(D4) of the default theory T_R.

check that the clause is implied by ϕ, by assuring that the propositional variable S is derivable. [13]

Proposition 6.1. *For every database $D = (U, \phi)$ over $\overline{R}$, the default theory $T_R + D$ has an extension iff $\mathbf{P}_R(D) = true$.*

Proof. ($\Rightarrow$) Let E be any extension of $T_R + D$. Clearly, E must be consistent. Consider any clause $R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)$ such that $\{\mathbf{c}_1, \ldots, \mathbf{c}_n\} = \{\mathbf{d} \mid C(\mathbf{d}) \in E\}$. From (D3) we conclude that $R(\mathbf{c}_i) \notin E$, and hence $\phi \not\models R(\mathbf{c}_i)$, for all $i = 1, \ldots, n$. From (D4) we conclude that $S \in E$; hence, by Lemma 2.3

$$[W_R]_U \cup \{\phi\} \cup \mathrm{CONS}(\mathrm{GD}([\Delta_R]_U, E)) \models S.$$

From the formulas of $[W_R]_U$ (cf. (F)), we conclude that $\phi \models R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)$. Thus, $\mathbf{P}(D) = true$.

($\Leftarrow$) If $\mathbf{P}_R(D) = true$, then there exist $\mathbf{c}_1, \ldots, \mathbf{c}_n$ such that $\phi \models R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)$ and $\phi \not\models R(\mathbf{c}_i)$, for all $i = 1, \ldots, n$. Define

$$E = [W_R]_U \cup \{\phi\} \cup \{C(\mathbf{c}_1), \ldots, C(\mathbf{c}_n)\} \cup \{\neg C(\mathbf{d}) \mid \mathbf{d} \notin \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}\}.$$

Notice that $E \models S$. Hence, it is easy to see that $\mathrm{Th}(E)$ is an extension of $T_R + D$. $\square$

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

Proof. As shown in [13], deciding whether a propositional default theory has an extension is a Σ_2^p-complete problem. For any fixed default theory T, the size of $T + D$ grows polynomially with the size of D. Hence, the result follows. $\square$

[13] The reader may have noted that T_R is the translation of an extended logic program P into default logic. Since T_R expresses $\mathbf{P}_R$, this fact seems to contradict Remark 5.9, which states that $\mathbf{P}_R$ cannot be expressed by any extended disjunctive program. This apparent contradiction can be explained by noting that in extended programs, disjunction (denoted by '|') is not interpreted classically (cf. [12]); the clauses of the input database should be interpreted as if they had the form $A_1 \mid \cdots \mid A_k$. This causes T_R and the corresponding logic program P to behave in totally different ways. For example, let $D = (\{a,b\}, R(a) \vee R(b))$. Clearly, D has the property $\mathbf{P}_R$, and $T_R + D$ has an extension (containing, among other things, $C(a), C(b)$ and S). On the contrary, the corresponding extended program $P \cup \{R(a) \mid R(b)\}$ has no answer sets.

Moreover, there exist database properties defined in default logic which are Σ_2^p-hard to compute (cf. Section 6.1). 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 6.3. *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.

If $\Delta = \{ \delta_1(\mathbf{x}), \ldots, \delta_n(\mathbf{x}) \}$, where each $\delta_i(\mathbf{x})$ has the form

$$\delta_i(\mathbf{x}) = \frac{A_i(\mathbf{x}) : B_{i,1}(\mathbf{x}) \cdots B_{i,k(i)}(\mathbf{x})}{C_i(\mathbf{x})}$$

then the corresponding set of defaults $\mathrm{pf}(\Delta)$ is illustrated in Fig. 2. Intuitively, $\mathrm{pf}(\Delta)$ "guesses" a set of generating defaults and a strict partial order, then it checks the conditions of Lemma 2.4.

The predicates gen_i, $prc_{i,j}$, con_i, con_{GD}, $derA_i$ and $der\overline{B}_{i,l}$ must be new predicates, with no occurrences in T and $\overline{R}$. Intuitively, $gen_i(\mathbf{x})$ means that $\delta_i(\mathbf{x})$ is a generating default. The facts of the form $\delta_i(\mathbf{x}) \prec \delta_j(\mathbf{y})$ are encoded by $prc_{i,j}(\mathbf{x},\mathbf{y})$.

The theory $\mathrm{pf}(T)$ simulates the defaults of T without deriving their consequents; (PF 1)–(PF 2) guess the generating defaults of an extension of T while (PF 3)–(PF 4) guess a strict partial order over such defaults ((PF 5)–(PF 6) ensure that the predicates $prc_{i,j}$ actually represent a strict partial order); then (PF 13)–(PF 15) verify that the guessed defaults generate an extension of T by checking the conditions of Lemma 2.4. Roughly speaking, $con_i(\mathbf{x})$ stands for the conjunction of the consequents of the defaults that should be applied before $\delta_i(\mathbf{x})$, i.e.,

$$\bigwedge \mathrm{CONS}(\{ \delta_j(\mathbf{y}) \mid \delta_j(\mathbf{y}) \prec \delta_i(\mathbf{x}) \})$$

Actually, $con_i(\mathbf{x})$ is not equivalent to the above conjunction; the extensions of $\mathrm{pf}(T)$ contain only

$$con_i(\mathbf{x}) \rightarrow \bigwedge \mathrm{CONS}(\{ \delta_j(\mathbf{y}) \mid \delta_j(\mathbf{y}) \prec \delta_i(\mathbf{x}) \})$$

which is sufficient for our purposes; the above implication is enforced by (PF 7). The atom con_{GD} corresponds in a similar way to

$$\bigwedge \mathrm{CONS}(\{ \delta_j(\mathbf{y}) \mid \delta_j(\mathbf{y}) \text{ is a generating default} \}).$$

The correspondence is enforced by (PF 8).

Finally, for all defaults $\delta_i(\mathbf{x})$, the atom $derA_i(\mathbf{x})$ means that $A_i(\mathbf{x})$ can be derived from the guessed extension; similarly, $der\overline{B}_{i,j}(\mathbf{x})$ means that $\neg B_{i,j}(\mathbf{x})$ can be derived. The intended meaning of $derA_i(\mathbf{x})$ and $der\overline{B}_{i,j}(\mathbf{x})$ is enforced by (PF 9) and (PF 11),

$$\text{(PF 1)}\ \frac{: gen_i(\mathbf{x})}{gen_i(\mathbf{x})} \qquad \text{(PF 2)}\ \frac{: \neg gen_i(\mathbf{x})}{\neg gen_i(\mathbf{x})}$$

$$\text{(PF 3)}\ \frac{: gen_i(\mathbf{x}), gen_j(\mathbf{y}), prc_{i,j}(\mathbf{x},\mathbf{y})}{prc_{i,j}(\mathbf{x},\mathbf{y})} \qquad \text{(PF 4)}\ \frac{: \neg prc_{i,j}(\mathbf{x},\mathbf{y})}{\neg prc_{i,j}(\mathbf{x},\mathbf{y})}$$

$$\text{(PF 5)}\ \frac{: prc_{i,j}(\mathbf{x},\mathbf{y}), prc_{j,k}(\mathbf{y},\mathbf{z}), \neg prc_{i,z}(\mathbf{x},\mathbf{z})}{\bot} \qquad \text{(PF 6)}\ \frac{: prc_{i,i}(\mathbf{x},\mathbf{x})}{\bot}$$

$$\text{(PF 7)}\ \frac{: prc_{i,j}(\mathbf{x},\mathbf{y})}{con_j(\mathbf{y}) \to C_i(\mathbf{x})} \qquad \text{(PF 8)}\ \frac{: gen_i(\mathbf{x})}{con_{GD} \to C_i(\mathbf{x})}$$

$$\text{(PF 9)}\ \frac{: \neg(con_{GD} \to A_i(\mathbf{x}))}{\neg derA_i(\mathbf{x})} \qquad \text{(PF 10)}\ \frac{: derA_i(\mathbf{x})}{derA_i(\mathbf{x})}$$

$$\text{(PF 11)}\ \frac{: \neg(con_{GD} \to \neg B_{i,l}(\mathbf{x}))}{\neg der\overline{B}_{i,l}(\mathbf{x})} \qquad \text{(PF 12)}\ \frac{: der\overline{B}_{i,l}(\mathbf{x})}{der\overline{B}_{i,l}(\mathbf{x})}$$

$$\text{(PF 13)}\ \frac{: gen_i(\mathbf{x}), \neg(con_i(\mathbf{x}) \to A_i(\mathbf{x}))}{\bot}$$

$$\text{(PF 14)}\ \frac{: gen_i(\mathbf{x}), der\overline{B}_{i,l}(\mathbf{x})}{\bot} \quad \text{where } 0 < l \le k(i)$$

$$\text{(PF 15)}\ \frac{: \neg gen_i(\mathbf{x}), derA_i(\mathbf{x}), \neg der\overline{B}_{i,1}(\mathbf{x}), \ldots, \neg der\overline{B}_{i,k(i)}(\mathbf{x})}{\bot}$$

Fig. 2. The set of defaults pf(T).

respectively; (PF 10) and (PF 12) complete their definitions. The intuitive behaviour of the new predicates is formalized by the following lemma.

Lemma 6.4. *Let E' be a consistent extension of* pf$(T) + D$, *where* $T = \langle W, \Delta\rangle$ *and* $D = (U, \phi)$. *Let W_D be an abbreviation for* $[W]_U \cup \{\phi\}$ *and let G be an abbreviation for* $\{\delta_i(\mathbf{c}) \mid gen_i(\mathbf{c}) \in E'\}$. *Finally, define* $\delta_i(\mathbf{c}) \prec \delta_j(\mathbf{d})$ *iff* $prc_{i,j}(\mathbf{c},\mathbf{d}) \in E'$. *We have:*

(i) $(con_i(\mathbf{c}) \to A_i(\mathbf{c})) \in E' \iff W_D \cup \mathrm{CONS}(\{\delta \mid \delta \prec \delta_i(\mathbf{c})\}) \vdash A_i(\mathbf{c})$,
(ii) $(con_{GD} \to A_i(\mathbf{c})) \in E' \iff W_D \cup \mathrm{CONS}(G) \vdash A_i(\mathbf{c})$,
(iii) $\neg derA_i(\mathbf{c}) \in E' \iff W_D \cup \mathrm{CONS}(G) \nvdash A_i(\mathbf{c})$,
(iv) $der\overline{B}_{i,l}(\mathbf{c}) \in E' \iff W_D \cup \mathrm{CONS}(G) \vdash \neg B_{i,l}(\mathbf{c})$.

Proof. (i) Note that con_i occurs only in the consequent of (PF 7); therefore the sentences of $W_D \cup \mathrm{CONS}(\mathrm{GD}(\mathrm{pf}(\Delta), E'))$ which contain con_i are exactly those of the form: $con_i(\mathbf{c}) \to C_j(\mathbf{d})$ where $\neg prc_{j,i}(\mathbf{d},\mathbf{c}) \notin E'$ (and $C_j(\mathbf{d})$ is the consequent of $\delta_j(\mathbf{d})$). Then standard interpolation arguments can be used to derive

$$\begin{aligned} & W_D \cup \mathrm{CONS}(\mathrm{GD}(\mathrm{pf}(\Delta), E')) \vdash con_i(\mathbf{c}) \rightarrow A_i(\mathbf{c}) \\ \iff & W_D \cup \mathrm{CONS}(\mathrm{GD}(\mathrm{pf}(\Delta), E')) \cup \{ con_i(\mathbf{c}) \} \vdash A_i(\mathbf{c}) \\ \iff & W_D \cup \mathrm{CONS}(\{ \delta_j(\mathbf{d}) \mid \neg prc_{j,i}(\mathbf{d}, \mathbf{c}) \notin E' \}) \vdash A_i(\mathbf{c}) \end{aligned}$$

Then (i) follows by noting that $E' = \mathrm{Th}(W_D \cup \mathrm{GD}(\mathrm{pf}(\Delta), E'))$ and that, due to (PF 4), $\neg prc_{j,i}(\mathbf{d}, \mathbf{c}) \notin E' \iff prc_{j,i}(\mathbf{d}, \mathbf{c}) \in E' \iff \delta_j(\mathbf{d}) \prec \delta_i(\mathbf{c})$.

(ii) Similar.

(iii) Note that $\neg derA_i(\mathbf{c})$ can be derived only through (PF 9); it follows that $\neg derA_i(\mathbf{c}) \in E' \iff (con_{GD} \rightarrow A_i(\mathbf{c})) \notin E'$; from this fact and (ii) we get (iii).

(iv) Similar to the proof of (iii); instead of (ii) use

$$\text{(ii}'\text{)} \quad (con_{GD} \rightarrow \neg B_{i,l}(\mathbf{c})) \in E' \iff W_D \cup \mathrm{CONS}(G) \vdash \neg B_{i,l}(\mathbf{c})$$

which can be proved in a similar way. □

It is not difficult to see that for all D, $T + D$ has an extension iff $\mathrm{pf}(T) + D$ does; this proves Theorem 6.3. A formal proof can be found in Appendix B.

Next we will show that prerequisite-free default theories cannot express all the polynomial-time recognizable properties of positive databases. Consider the following property.

Definition 6.5. For all positive databases $D = (U, \phi)$ from $\mathscr{D}(\overline{R})$, $\mathbf{P}_{\mathrm{ev}}(D) = true$ iff the dimensions of the clauses in the normalized sentence ϕ^* are all even.

Theorem 6.6. $\mathbf{P}_{\mathrm{ev}}$ *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 $\mathbf{P}_{\mathrm{ev}}$ so 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 $\mathbf{P}_{\mathrm{ev,g}}$.

Basically, the proof of Theorem 6.6 goes as follows: we exploit two interpolation properties of DL (Lemmas 6.7 and 6.8) 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 6.9); then we show that no fixed default theory can provide so many defaults. The proofs of the first two lemmas, which state some properties of default logic in general, can be found in the appendix.

Lemma 6.7. *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 6.8. *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 6.9. *Let $D_1 = (U,\phi_1)$ and $D_2 = (U,\phi_2)$ be nonequivalent 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. Suppose not, i.e., suppose that for some extensions E_1 and E_2 of $T + D_1$ and $T + D_2$ (respectively) we have $\mathrm{GD}([\Delta]_U,E_1) = \mathrm{GD}([\Delta]_U,E_2)$.

Let ϕ be the normal sentence equivalent to $\phi_1 \vee \phi_2$, and let $D = (U,\phi)$. By Lemma 6.8 we have that the default theory $T + D$ has an extension E such that $\mathrm{GD}([\Delta]_U,E) = \mathrm{GD}([\Delta]_U,E_1)$. Since T expresses $\mathbf{P}_{\mathrm{ev}}$, it follows that $\mathbf{P}_{\mathrm{ev}}(D) = true$.

Since ϕ_1 and ϕ_2 are not equivalent we have either $\phi \not\models \phi_1$ or $\phi \not\models \phi_2$; without loss of generality we may assume that $\phi \not\models \phi_1$. As a consequence, ϕ contains a normal clause C which is strictly subsumed by some normal clause C_1 of ϕ_1^*. Since both D and D_1 have the property $\mathbf{P}_{\mathrm{ev}}$, it follows that C has dimension n, where n is some even integer, and C_1 has dimension n', where n' is even and $n' \leqslant n-2$; therefore, we can find an intermediate normal clause C' whose dimension is odd (namely, $n-1$) and such that $C_1 \models C'$ and $C' \models C$. Now obtain D' from D by replacing C with C' in ϕ. Obviously $\mathbf{P}_{\mathrm{ev}}(D') \neq true$. However, Lemma 6.7 can be applied to $T + D_1$ and $T + D$ to derive that $T + D'$ has an extension; this contradicts the assumption that T expresses $\mathbf{P}_{\mathrm{ev}}$. $\square$

Proof of Theorem 6.6. 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.

Let U be any universe with cardinality $n = 4i$ with i an 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 $\mathbf{P}_{\mathrm{ev}}$. To prove the theorem it suffices to show that the following inequalities hold for some polynomial in n, $p(n)$:

$$N \leqslant 2^{p(n)}, \tag{5}$$

$$N \geqslant 2^{2^{n/2}}. \tag{6}$$

In fact, (5) and (6) imply $2^{2^{n/2}} \leqslant 2^{p(n)}$ but for sufficiently large universes we have $2^{2^{n/2}} > 2^{p(n)}$, hence a contradiction.

First we prove (5). By assumption, T expresses $\mathbf{P}_{\mathrm{ev}}$, 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 6.9 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 $N \leqslant 2^l$. 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 $N \leqslant 2^{p(n)}$, which proves (5).

We are left to prove (6). The number of normal clauses of dimension $2j$ is

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

therefore, the number of databases of S that contain only clauses of dimension $2j$ is at least $2^{2^{n/2}}$. Disequation (6) follows immediately. □

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

Corollary 6.10. *There exist PTIME-recognizable properties of positive databases that cannot be expressed through brave reasoning in prerequisite-free default logic.*

Remark 6.11. The proof of Theorem 6.6 cannot be immediately extended to general default theories. It is easy to see that Lemmas 6.7 and 6.8 are not valid if T_1 and T_2 are not prerequisite-free, and it seems very hard to find alternative interpolation properties.

6.1. Bounded databases

In this section we will restrict our attention to bounded databases, i.e., the databases in $\mathscr{D}(\overline{R})_{\leqslant d}$, for a fixed constant d. On this class of databases, default logic exactly captures the class of Σ_2^p-recognizable database properties.

Theorem 6.12. *For every Σ_2^p-recognizable property* $\mathbf{P}$ *defined over $\mathscr{D}(\overline{R})_{\leqslant d}$, 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 $\mathscr{D}(\overline{R})_{\leqslant d}$ 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 $D \in \mathscr{D}(\overline{R}')_{\leqslant 1}$, 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 6.13. *Let* $\mathbf{P}$ *be a Σ_2^p-recognizable database property over $\mathscr{D}(\overline{R})_{\leqslant 1}$. Then, there exists a set of defaults Δ such that for every $D = (U, \phi) \in \mathscr{D}(\overline{R})_{\leqslant 1}$, 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.*[14]

Thus, intuitively, T can be constructed from 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 $\mathbf{P}$.

Proof of Theorem 6.12. To specify the details we need some notation. A *clause pattern* is a formula

$$\pm R_{i_1}(\mathbf{x}_1) \vee \cdots \vee \pm R_{i_h}(\mathbf{x}_h) \tag{7}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_h$ form a vector of pairwise distinct variables, and $\pm R_{ij}(x_j)$ stands for either $R_{i_j}(\mathbf{x}_j)$ or $\neg R_{i_j}(\mathbf{x}_j)$, where R_{i_j} is from $\overline{R}$. If $h \leqslant d$ we say that the pattern is *d-bounded*. For a fixed d, the set of d-bounded clause patterns is finite up to variable renaming; in the following, let $CP_d(\overline{R}) = \{C_1, \ldots, C_N\}$ be a fixed minimal and complete set of d-bounded clause patterns (i.e. each d-bounded pattern C is a variant of exactly one C_i in $CP_d(\overline{R})$).

Every disjunctive database $D = (U, \phi) \in \mathscr{D}(\overline{R})_{\leqslant d}$ is logically equivalent to the clause set

$$\mathscr{C}_D = \{\, C_i\theta \mid C_i \in CP_d(\overline{R}), D \models C_i\theta, \text{ and } \theta \text{ is a ground substitution over } U \,\}.$$

We use this set $\mathscr{C}_D$ to encode D into a relational database $D_{\mathrm{r}} = \langle U_{\mathrm{r}}, \phi_{\mathrm{r}} \rangle$ (called *relational representation of D*) as follows. Introduce for each pattern $C_i \in CP_d(\overline{R})$ of the form (7) a new predicate P_i of arity $a_{i_1} + \cdots + a_{i_h}$. Then,

- let $U_{\mathrm{r}} = U(D)$;
- let ϕ_{r} be a conjunction of all ground atoms $P_i\theta$ such that $D \models C_i\theta$ for some ground substitution θ, for all $C_i \in CP_d(\overline{R})$.

(Notice that ϕ_{r} is unique modulo the ordering of atoms.) For each database property $\mathbf{P}$ defined over $\mathscr{D}(\overline{R})_{\leqslant d}$, we can define a corresponding property $\mathbf{P}_{\mathrm{r}}$ defined over the relational representations of $\mathscr{D}(\overline{R})_{\leqslant d}$ by $\mathbf{P}_{\mathrm{r}}(D_{\mathrm{r}}) = \mathbf{P}(D)$. Note that $\mathbf{P}_{\mathrm{r}}$ is well-defined, because different positive databases may have the same relational representation only if they are logically equivalent, and $\mathbf{P}$ is invariant under logical equivalence by definition.

The sets of defaults Δ_1 and Δ_2 of T_1 and T_2, respectively, are as follows. Δ_1 consists of the defaults

$$\frac{C_i(\mathbf{x}_1 \cdots \mathbf{x}_h) :}{P_i(\mathbf{x}_1 \cdots \mathbf{x}_h)}, \quad \frac{: \neg C_i(\mathbf{x}_1 \cdots \mathbf{x}_h)}{\neg P_i(\mathbf{x}_1 \cdots \mathbf{x}_h)}$$

obtained from all clause patterns $C_i \in CP_d$. Δ_2 is the set of defaults which express the Σ_2^p-recognizable database property $\mathbf{P}_{\mathrm{r}}$ (Δ_2 exists by Lemma 6.13); we may assume without loss of generality that the only predicate symbols occurring both in T_1 and in T_2 are the predicates P_i that encode the clause patterns C_i. Define $T = \langle \emptyset, \Delta_1 \cup \Delta_2 \rangle$. By standard interpolation properties, the following claim can be easily verified.

[14] COMP(ϕ) yields the completion of ϕ under the closed-world assumption.

Claim. *For every $D = (U, \phi) \in \mathscr{D}(\overline{R})_{\leqslant d}$, $T + D$ has an extension iff* $\mathbf{P}(D) = true$.

Proof. ($\Leftarrow$) If $\mathbf{P}(D) = true$ then $\mathbf{P}_r(D_r) = true$ and hence $\langle \mathrm{COMP}(D_r), [\Delta_2]_U \rangle$ has an extension E. Define $E' = \mathrm{Th}(E \cup \{[\phi]_U\})$. Then, E' is an extension of $T + D$.

($\Rightarrow$) Assume that E is an extension of $T + D = \langle \{[\phi]_U\}, [\Delta_1 \cup \Delta_2]_U \rangle$, and let E' be the restriction of E to all formulas in which no predicate from $\overline{R}$ occurs. Clearly, E is consistent. Hence, it follows that for every P_i and ground substitution θ, $P_i\theta \in E$ iff $D \models C_i\theta$ and $\neg P_i\theta \in E$ iff $D \not\models C_i\theta$; consequently, $\mathrm{COMP}(D_r) \subseteq E'$. It follows that E' is an extension of $\langle \mathrm{COMP}(D_r), [\Delta_2]_U \rangle$, which means that $\mathbf{P}_r(D_r) = true$. Consequently, $\mathbf{P}(D) = true$. This proves the claim.

Consequently, the default theory T defines the property $\mathbf{P}$ over $\mathscr{D}(\overline{R})_{\leqslant d}$. This proves the theorem. $\square$

A similar result will be derived for autoepistemic logic. On the contrary, model-preference based logics cannot express all the Σ_2^p-recognizable properties over bounded databases.

Theorem 6.14. *For all $d > 1$, the properties $\mathbf{P}_{\mathrm{MM}}$ and $\mathbf{P}_R$ restricted to $\mathscr{D}(\overline{R})_{\leqslant d}$ cannot be expressed by any first-order theory under any brave preferential semantics.*

Proof. Identical to the proofs of Theorems 5.2 and 5.5. $\square$

As a consequence, as far as bounded databases are concerned, default logic is strictly more expressive than any preference logic whose brave semantics is in Σ_2^p. In the next section we will show that autoepistemic logic has exactly the same expressive power as default logic.

7. Expressive power of AEL

In this section we analyse the expressive power of autoepistemic logic used as a query language over disjunctive databases. We show that AEL can express exactly the same queries as default logic. This result is established by showing that AEL has the same expressive power as some of its fragments, including what we call prerequisite-free AEL, which naturally corresponds to prerequisite-free default logic. Thus, from the results of the previous section, it follows that AEL and default logic have the same expressive power.

7.1. Prerequisite-free AEL

Intuitively, an autoepistemic normal formula of the form

$$L\alpha_1 \wedge \cdots \wedge L\alpha_m \wedge \neg L\beta_1 \wedge \cdots \wedge \neg L\beta_n \rightarrow \phi$$

(cf. (4)) can be regarded as a default $(\alpha_1 \wedge \cdots \wedge \alpha_m : \neg\beta_1 \cdots \neg\beta_n/\phi)$. Accordingly, we say that a normal autoepistemic theory T is *prerequisite-free* if all formulas in T have the form (4) with $m = 0$.

A semantical correspondence between prerequisite-free DL and prerequisite-free AEL can be established through Konolige's translation of default theories into AEL [23].

Definition 7.1. For all defaults $\delta = (\alpha : \beta_1 \cdots \beta_n/\gamma)$ define

$$\mathrm{tr}(\delta) = L\alpha \wedge \neg L\neg\beta_1 \wedge \cdots \neg L\neg\beta_n \rightarrow \gamma.$$

For all default theories $T = \langle W, \Delta\rangle$ define $\mathrm{tr}(T) = W \cup \{\,\mathrm{tr}(\delta) \mid \delta \in \Delta\,\}$.

Notice that for each prerequisite-free default theory T, $\mathrm{tr}(T)$ is a prerequisite-free AEL theory; conversely, for every prerequisite-free AEL theory T, there exists a prerequisite-free default theory T' such that $\mathrm{tr}(T') = T$. It is well-known that, in general, the extensions of T do not correspond exactly to the stable expansions of $\mathrm{tr}(T)$. But when T is prerequisite-free, we have a one-to-one correspondence.

Lemma 7.2. (cf. also Theorems 12.19, 12.20 in Marek and Truszczyński [29, pp. 369, 371]) *For all prerequisite-free default theories $T = \langle W, \Delta\rangle$:*

(i) *If E is an extension of T then* $\mathrm{E}(E)$ *(the unique stable expansion of E) is a stable expansion of* $\mathrm{tr}(T)$.

(ii) *Conversely, if E is a stable expansion of* $\mathrm{tr}(T)$ *then E_0 (the kernel of E) is an extension of T.*

Proof. Point (i) is a special case of a well-known property of tr (which holds for unrestricted default theories). To prove (ii) let E be any stable expansion of $\mathrm{tr}(T)$. By Theorem 2.8 we have $E_0 = \mathrm{Th}(\mathrm{tr}(T)^{E_0})$. Note that $\mathrm{tr}(T)^{E_0} = W^{E_0} \cup \{\,\mathrm{tr}(\delta) \mid \delta \in \Delta\,\}^{E_0} = W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E_0))$. From the above equalities we get $E_0 = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E_0)))$, which implies, by Corollary 2.5, that E_0 is an extension of T. $\square$

Thus, we easily obtain the following result on the expressiveness of prerequisite-free AEL.

Theorem 7.3. *Prerequisite-free default theories and prerequisite-free normal autoepistemic theories express the same class of properties over disjunctive databases or subsets thereof.*

Proof. It suffices to show that for all default theories T, $\mathrm{tr}(T)$ and T express the same property.

Note that for all default theories T and all disjunctive databases $D = (U, \phi)$, $\mathrm{tr}(T + D) = \mathrm{tr}(T) + D$. By Lemma 7.2 we have that $\mathrm{tr}(T + D)$ has a stable expansion iff $T + D$ has an extension. It follows that $\mathrm{tr}(T) + D$ has a stable expansion iff $T + D$ has an extension; this implies that $\mathrm{tr}(T)$ and T express the same property. $\square$

7.2. Expressiveness of AEL

Our first step in analysing the expressive power of AEL concerns the role of inconsistent stable expansions. We show that excluding autoepistemic theories T such that $T+D$ can have an inconsistent stable expansion, does not diminish the expressive power of AEL. This observation will be used later for showing that AEL can express the same Boolean queries as prerequisite-free AEL, and hence the same Boolean queries as prerequisite-free default logic. Formally, we introduce the following notion.

Definition 7.4. We say that an autoepistemic theory T is *everywhere consistent* if for all databases D and for all stable expansions E of $T+D$, E is consistent. The class of everywhere consistent autoepistemic theories is referred to as *everywhere consistent AEL*.

We show that every normal autoepistemic theory can be rewritten to an everywhere consistent normal autoepistemic theory, such that the existence of a stable expansion is preserved. The following lemma is useful for this purpose.

Lemma 7.5. *Let T be a theory in which f does not occur, and let $S \subseteq S' \subseteq T$. Define $T' = (T \setminus S) \cup \{\psi \vee f \mid \psi \in S'\}$. Then, for every formula ϕ in which f does not occur,*

$$T \models \phi \quad \textit{iff} \quad T' \models \phi \vee f.$$

Proof. If $T \models \phi$, then clearly $T' \models \phi \vee f$. If $T \not\models \phi$, there exists a model M such that $M \models T$ and $M \not\models \phi$. Let $M' = M \setminus \{f\}$. Then, $M' \not\models \phi$, and hence also $M' \not\models \phi \vee f$. On the other hand, $M' \models T'$. It follows that $T' \not\models \phi \vee f$. This proves the result. □

Lemma 7.6. *Let T be a normal AEL theory, i.e., all formulas are of the form*

$$\neg L\alpha_1 \wedge \cdots \wedge \neg L\alpha_k \wedge L\beta_1 \wedge \cdots \wedge L\beta_\ell \rightarrow \phi,$$

cf. (4), *and let f be a propositional symbol which occurs neither in T nor in $\overline{R}$. Let $f(T)$ be the theory obtained from T by replacing each formula by*

$$\neg L(\alpha_1 \vee f) \wedge \cdots \wedge \neg L(\alpha_k \vee f) \wedge L(\beta_1 \vee f) \wedge \cdots \wedge L(\beta_\ell \vee f) \rightarrow \phi \vee f.$$

Then, for every database D, it holds that $T+D$ has a stable expansion iff $f(T)+D$ has a stable expansion.

Proof. ($\Leftarrow$) Let E be the kernel of a stable expansion of $f(T)+D$. Define

$$T' = \{\phi \mid \phi \vee f \in E, f \text{ does not occur in } \phi\}.$$

We show that $\mathrm{Th}(T')$ is the kernel of a stable expansion of $T+D$. Thus, we have to show that $\mathrm{Th}(T') = \mathrm{Th}((T+D)^{\mathrm{Th}(T')})$.

Let ϕ be an arbitrary formula in which f does not occur. Then,

$$\begin{aligned} \phi \in T' &\iff \phi \vee f \in E && \text{(by definition)} \\ &\iff \phi \vee f \in \mathrm{Th}((f(T)+D)^{E}) && (E \text{ is kernel}) \\ &\iff \phi \in \mathrm{Th}((T+D)^{\mathrm{Th}(T')}) && \text{(by Lemma 7.5)} \end{aligned}$$

Since f does not occur in any formula from T' and $(T+D)^{\mathrm{Th}(T')}$, it follows that $T' \subseteq \mathrm{Th}((T+D)^{\mathrm{Th}(T')})$ and $(T+D)^{\mathrm{Th}(T')} \subseteq T'$. Consequently, $\mathrm{Th}(T') = \mathrm{Th}((T+D)^{\mathrm{Th}(T)})$. This proves the "$\Leftarrow$" part.

($\Rightarrow$) Let E be the kernel of some stable expansion of $T+D$. Define

$$T' = \{\, \phi \vee f \mid \phi \in E, f \textit{ does not occur in } \phi \,\} \cup D.$$

We show that $\mathrm{Th}(T')$ is the kernel of a stable expansion of $f(T)+D$. Thus, we have to show that $\mathrm{Th}(T') = \mathrm{Th}((f(T)+D)^{\mathrm{Th}(T')})$. Let ϕ be an arbitrary formula in which f does not occur. Then,

$$\begin{aligned} \phi \vee f \in T' &\iff \phi \in E && \text{(by definition)} \\ &\iff \phi \in \mathrm{Th}((T+D)^{E}) && (E \text{ is kernel}) \\ &\iff \phi \vee f \in \mathrm{Th}((f(T)+D)^{\mathrm{Th}(T')}) && \text{(by Lemma 7.5)} \end{aligned}$$

Clearly, we have $D \subseteq T'$ and $D \subseteq (f(T)+D)^{\mathrm{Th}(T')}$. Notice that each formula ψ from T' and $(f(T)+D)^{\mathrm{Th}(T')}$ such that $\psi \notin D$ is of the form $\phi \vee f$ where f does not occur in ϕ. Consequently, we obtain that

$$T' \subseteq \mathrm{Th}((f(T)+D)^{\mathrm{Th}(T')})$$

and

$$(f(T)+D)^{\mathrm{Th}(T')} \subseteq T'.$$

It follows that

$$\mathrm{Th}(T') = \mathrm{Th}((f(T)+D)^{\mathrm{Th}(T')}).$$

This concludes the part ($\Rightarrow$) of the proof. The result follows. □

Through the above lemma, one can easily prove that everywhere consistent AEL can express the same properties as unrestricted AEL (cf. Theorem 7.8 below).

Our next goal is to show that all Boolean queries that can be expressed in everywhere consistent AEL can also be expressed by prerequisite-free normal autoepistemic theories. Thus, by combining this result with the previous one, we obtain that unrestricted AEL expresses the same properties as prerequisite-free normal AEL. The key result is the following lemma.

Lemma 7.7. *If a property* **P** *of unbounded databases can be expressed by an everywhere consistent autoepistemic theory T under the brave semantics, then* **P** *can also be expressed by a prerequisite-free normal autoepistemic theory T' under the brave semantics.*

The intuition underlying the proof is very simple: for all conjuncts $L\alpha$ in the body of a normal formula like (4) we introduce a new predicate symbol p_α whose intended meaning is "α is not derivable"; then we replace $L\alpha$ with the equivalent formula $\neg L p_\alpha$. Each new predicate is defined by a single rule

$$\neg L\alpha \rightarrow p_\alpha$$

The formal proof can be found in Appendix D. Finally, we arrive at the main result of this section.

Theorem 7.8. *The following languages express under the brave semantics the same class of properties over unbounded disjunctive databases.*

(i) *autoepistemic logic (AEL)*
(ii) *everywhere consistent AEL*
(iii) *prerequisite-free AEL*

Proof. It suffices to show that expressibility of a property in one of the above languages entails expressibility in the others.

(i) ⇒ (ii): This can be concluded from Lemma 7.6: It is easy to see (cf. Theorem 2.8) that, for every normal autoepistemic theory T, the theory $f(T)$ is everywhere consistent. Since normal autoepistemic theories express the same as unrestricted autoepistemic theories, the claim follows.

(ii) ⇒ (iii): This follows from Lemma 7.7.

(iii) ⇒ (i): Trivial. □

Through the above theorem and Theorem 7.3 we can immediately extend the results on the expressive power of default logic to autoepistemic logic.

Corollary 7.9. *There exist PTIME-recognizable properties of positive databases that cannot be expressed by any autoepistemic theory under the brave semantics. For example, the property* $\mathbf{P}_{\mathrm{ev}}$ *cannot be expressed by any autoepistemic theory under the brave semantics.*

Corollary 7.10. *For all Σ_2^p-recognizable properties* **P** *defined over $\mathscr{D}_{\leqslant d}(\overline{R})$, there exists a prerequisite-free autoepistemic theory T that captures* **P** *under brave reasoning.*

Proof. By Theorems 6.12 and 6.3, every Σ_2^p-recognizable property **P** defined over $\mathscr{D}_{\leqslant d}(\overline{R})$ can be expressed by a prerequisite-free default theory T. It follows, by Theorem 7.3, that the prerequisite-free autoepistemic theory $\mathrm{tr}(T)$ expresses the same property as T, that is, $\mathrm{tr}(T)$ expresses **P**. □

8. Expressiveness of cautious semantics

The limitations of cautious semantics are similar to the limitations of brave semantics. For preferential semantics we have the following result.

Theorem 8.1. *The properties* $\overline{\mathbf{P}_R}$ *and* $\overline{\mathbf{P}_{\mathrm{MM}}}$ *cannot be expressed by any first-order theory under any brave preferential semantics.*

Proof. Analogous to the proof of Theorem 5.2. □

Corollary 8.2. *There exist properties of positive databases that are computable in polynomial time, but cannot be expressed under any cautious preferential semantics.*

The cautious version of default and autoepistemic logic is complementary to the brave version in the following sense.

Lemma 8.3. *For all default (resp. autoepistemic) theories T that express a property $\mathbf{P}$ under the brave semantics there exists a default (resp. autoepistemic) theory T' that expresses $\overline{\mathbf{P}}$ under the cautious semantics. Conversely, if T expresses $\mathbf{P}$ under the cautious semantics, then there exists T' that expresses $\overline{\mathbf{P}}$ under the brave semantics.*

Proof (*sketch*). First let T be an autoepistemic theory that expresses $\mathbf{P}$ under the brave semantics. There must be an everywhere-consistent theory T'' that expresses $\mathbf{P}$ under the brave semantics. Obtain T' by extending T'' with the axiom $\neg Lp \rightarrow p'$, where p is the output letter of T'' and p' is a new propositional letter that plays the role of output letter for T'. Clearly, for all D, the stable expansions of $T'' + D$ and $T' + D$ are in one-to-one correspondence; they agree on all formulas that do not contain p' and they are all consistent, because T'' is everywhere consistent. Moreover, a stable expansion of $T' + D$ contains p' iff p' is derivable through $\neg Lp \rightarrow p'$ (by consistency) iff the corresponding expansion of $T'' + D$ does not contain p. It follows immediately that T' expresses $\overline{\mathbf{P}}$ under the cautious semantics.

Conversely, if T expresses $\mathbf{P}$ under the cautious semantics, then one can show that there exists an everywhere consistent T'' that expresses $\mathbf{P}$ under the cautious semantics (same technique as Lemma 7.6). Obtain T' from T'' as before; T' expresses $\overline{\mathbf{P}}$ under the brave semantics. This completes the proof for autoepistemic theories.

To prove the theorem for default theories, transform T into a prerequisite-free default theory T'' that expresses $\mathbf{P}$ (under the brave semantics, this is possible by Theorem 6.3; a similar result for cautious semantics can be derived with similar techniques); then exploit the part of the lemma concerning AEL through the correspondence between prerequisite-free DL and AEL (Lemma 7.2). □

From this lemma and the results of the previous sections we immediately obtain an analysis of the expressiveness of cautious DL and AEL:

1. $\overline{\mathbf{P}_{\mathrm{ev}}}$ can be expressed neither in DL nor in AEL under the cautious semantics. Consequently, there exist PTIME-recognizable properties of positive databases that cannot be expressed through cautious DL and AEL.

2. Cautious DL and AEL capture exactly the Π_2^p-recognizable properties of bounded disjunctive databases.

3. The cautious versions of DL, prerequisite-free DL, AEL, prerequisite-free AEL and everywhere consistent AEL express exactly the same class of queries over disjunctive databases, which is a strict subclass of all the Π_2^p-recognizable queries.

9. Discussion and conclusions

The main results of this paper are summarized in Figs. 3 and 4. They concern brave semantics. 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 trivial relations; the other arrows illustrate the results of the previous sections. $\mathbf{P}_R$, $\mathbf{P}_{\mathrm{MM}}$ and $\mathbf{P}_{\mathrm{ev,g}}$ are the simple disjunctive database properties introduced in the Definitions 5.1, 5.4, and the extension of Definition 6.5.

We have shown that the brave versions of default logic (DL) and autoepistemic logic (AEL) express the same class of queries. This class is a strict subset of the queries recognizable in Σ_2^p, although it contains some Σ_2^p-hard queries.

Moreover, we have explored some relevant subsets of DL and AEL, namely prerequisite-free DL and AEL, and everywhere-consistent AEL. Prerequisite-free DL had already received some attention, cf. [29]. It was already known that every propositional default theory can be transformed into an equivalent prerequisite-free one (M. Truszczyński, personal communication); however, we had to prove this result in a more general setting, because the transformation must preserve the meaning of $T+D$ for *all* D (the pre-existing techniques yield different translations for different D). We have carried over the notion of prerequisite-free theory to AEL, and we have introduced everywhere-consistent AEL. Our results show that all of those fragments have exactly the same expressive power. The proofs are constructive; they allow to transform effectively each query expressed in any of those fragments into an equivalent query expressed in any of the other fragments. (It is interesting to note that each transformation can be carried out in polynomial time.)

A few comments on expression complexity and combined complexity [38] are needed. We recall that the expression (resp. combined) complexity is the complexity of evaluating a varying query over a fixed (resp. varying) database. It is easy to see from the results on query recognizability (where the query is fixed) that $\mathrm{NEXP}^{\mathrm{NP}}$ (the class of problems decidable in nondeterministic exponential time with an oracle in NP) is an upper bound for expression and combined complexity. On the other hand, it was reported in [5] that – according to expression as well as combined complexity – default

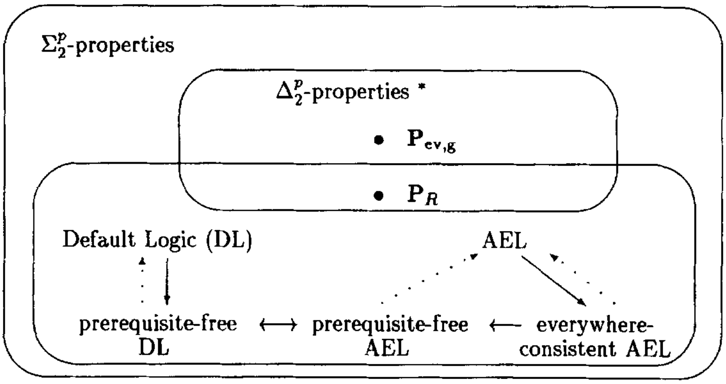

Fig. 3. Expressiveness for unbounded databases.

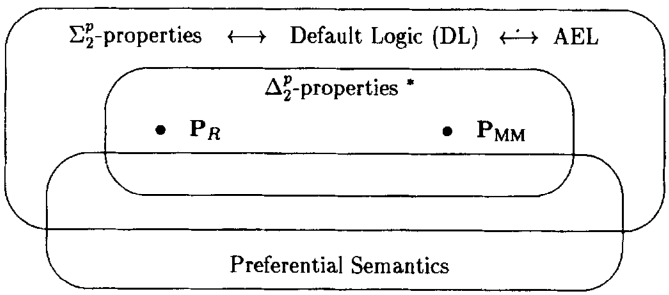

Fig. 4. Expressiveness for bounded databases (* over positive disjunctive databases, $\mathbf{P}_R$, $\mathbf{P}_{MM}$, and $\mathbf{P}_{ev,g}$ are polynomial-time recognizable.)

logic over relational databases is $NEXP^{NP}$-complete. Thus, by means of our polynomial transformations, it follows that all query languages in Fig. 3 have $NEXP^{NP}$-complete expression and combined complexity.

If we restrict our attention to bounded disjunctive databases (i.e., disjunctive databases whose clauses contain at most d literals, for some constant d), then DL and AEL can express all the Σ_2^p-recognizable properties. On the contrary, preferential semantics (including the minimal model semantics and the various forms of circumscription) cannot express all of those properties. In particular, the properties $\mathbf{P}_R$ and $\mathbf{P}_{MM}$ cannot be expressed by *any* preferential semantics, no matter how complex they can be (even highly noncomputable). Note that $\mathbf{P}_R$ can be expressed very easily in DL and AEL.

From the above results one can easily derive dual results for the cautious semantics. A picture of these results can be obtained by replacing Σ_2^p, $\mathbf{P}_R$, $\mathbf{P}_{MM}$ and $\mathbf{P}_{ev,g}$ with Π_2^p, $\overline{\mathbf{P}_R}$, $\overline{\mathbf{P}_{MM}}$ and $\overline{\mathbf{P}_{ev,g}}$ (respectively) in Figs. 3 and 4.

Summarizing, in general, none of the languages that have been considered in this paper is well-balanced (i.e., their expressiveness does not match their complexity). Over bounded databases, however, DL and AEL are well-balanced, while preference logics are not.

The generality of nonmonotonic logics makes it possible to extend our negative results to many extensions of logic programming. It is well-known that normal logic programs (under the stable semantics) can be regarded as subsets of AEL and/or DL. We can think of many possible extensions of the logic programming paradigm, intermediate between normal programs and full AEL or DL. Some of them have already been considered, e.g. the epistemic semantics introduced in [4], and the nondeterministic choice constructs analysed in [16]. We immediately know that none of these extensions makes it possible to express queries such as $\mathbf{P}_{\mathrm{ev,g}}$. It remains to be seen which of these extensions is well-balanced over bounded databases.

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, DATALOG under the minimal model semantics and DATALOG under the stable model semantics express exactly the same class of queries (cf. the results of this paper and [8]), while in the area of bounded disjunctive databases DL and AEL are strictly more expressive than DATALOG, which expresses $\mathbf{P}_R$ (or its complement in the cautious version) neither under the minimal model semantics nor under the stable model semantics.

The limitations of DL and AEL seem to arise essentially because the number of disjunctive databases grows tremendously fast with the size of the universe, while the instantiation of a query in DL or AEL grows at a much slower rate. (There are more than $2^{2^{|U|/2}}$ disjunctive databases with universe U, while the instantiation of a fixed theory in DL or AEL is polynomial in $|U|$.) When the database dimension is bounded, then the number of nonequivalent databases grows only polynomially in $|U|$, and in that case DL and AEL are well-balanced. It may be interesting to study how different notions of bounds affect the property of being well-balanced.

In order to overcome the limitations of DL and AEL, one can look for extensions based on the following two ideas, which have been proposed for similar purposes in the context of relational databases (cf. [2]): (1) Allow queries to introduce new constants; and (2) allow the use of predicates with nonfixed arities. However, such extensions remain to be explored.

Besides issues on nonmonotonic logics, there are some interesting issues concerning queries over disjunctive databases in general. In the framework of relational databases, second-order logic expresses all the PH-recognizable properties; moreover, there is a nice correspondence between the levels of the polynomial hierarchy and fragments of the logic. Finding a formalism which plays the same role for disjunctive databases seems an interesting direction for further research. Our results on preferential semantics show that second-order logic is not suitable, just as any other monotonic logic.

Finally, there exists an alternative view of databases, related to finite model theory. Relational databases and incomplete databases can be regarded as first-order interpretations and sets thereof, respectively [20, 14]. Equivalently, incomplete databases can be regarded as finite S5 Kripke structures. Under this view, the database property expressed by a logical formula ϕ is the characteristic function of the *generalized spectrum* of ϕ, that is, the set of finite models of ϕ. Our negative results allow to find properties of incomplete databases that cannot be expressed through S5, due to the relations between S5 and autoepistemic logic. The generalized spectra of modal sentences may deserve further attention.

Acknowledgements

The authors would like to thank Georg Gottlob for numerous discussions and 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. Moreover, they are grateful for useful review comments that helped to improve this paper.

Appendix A. Proof of Theorem 5.8

Theorem 5.8. *The property* $\mathbf{P}_R$ *cannot be expressed by any disjunctive logic program under the brave stable semantics.*

Proof. Let $D = (U, R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n))$, and P be any disjunctive logic program. We claim that:

Each stable model of $P + D$ is also a stable model of $P + D'$, where $D' = (U, R(\mathbf{c}_i))$, for some $i = 1, \ldots, n$.

To prove this claim, let M be a stable model of $P + D$, that is, $M \in \mathrm{MM}((P+D)^M)$ (where $(.)^M$ is the Gelfond–Lifschitz transformation w.r.t. M; see [32]). Note that $(P+D)^M = ([P]_U)^M \cup \{R(\mathbf{c}_1) \vee \cdots \vee R(\mathbf{c}_n)\}$; by applying Lemma 5.3 to the right hand side we get

$$M \in \mathrm{MM}(([P]_U)^M \cup \{R(\mathbf{c}_i)\}) \tag{$*$}$$

for some $i = 1, \ldots, n$. Now let $D' = (U, R(\mathbf{c}_i))$ and note that $([P]_U)^M \cup \{R(\mathbf{c}_i)\} = (P + D')^M$. It follows, by $(*)$, that M is a stable model of $P + D'$. This proves the claim.

Now proving that $\mathbf{P}_R$ cannot be expressed is easy. Assume that there exists a program P which expresses $\mathbf{P}_R$ under the brave stable semantics, and take D as defined above. Clearly D has the property $\mathbf{P}_R$, therefore $P + D$ should have a stable model M. By the above claim, M would also be a stable model of $P + D'$ (as defined above). But D' doesn't have the property $\mathbf{P}_R$, and hence P does not express $\mathbf{P}_R$. A contradiction. □

Appendix B. Proof of Theorem 6.3

Theorem 6.3. *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.

Proof. We have to prove that for all $D = (U, \phi)$, $T + D$ has an extension iff $\mathrm{pf}(T) + D$ has some. Some notation will be needed: we will abbreviate $[W]_U \cup \{\phi\}$ to W_D; thus $T + D = \langle W_D, [\Delta]_U \rangle$ and $\mathrm{pf}(T) + D = \langle W_D, [\mathrm{pf}(\Delta)]_U \rangle$; let $\Delta = \{\, \delta_1(\mathbf{x}), \ldots, \delta_n(\mathbf{x}) \,\}$ and

$$\delta_i(\mathbf{x}) = \frac{A_i(\mathbf{x}) : B_{i,1}(\mathbf{x}) \cdots B_{i,k(i)}(\mathbf{x})}{C_i(\mathbf{x})}$$

($\Longrightarrow$) Assume that $T + D$ has an extension E. If E is inconsistent then W_D must be inconsistent and hence E must also be an extension of $\mathrm{pf}(T) + D$.

Next assume that E is consistent. Let $\prec$ be a strict partial order over $\mathrm{GD}([\Delta]_U, E)$ which satisfies the condition (b) of Lemma 2.4. Define

$$\begin{aligned}
E' = W_D &\cup \{\, gen_i(\mathbf{c}) \mid \delta_i(\mathbf{c}) \in \mathrm{GD}([\Delta]_U, E) \,\} \\
&\cup \{\, \neg gen_i(\mathbf{c}) \mid \delta_i(\mathbf{c}) \notin \mathrm{GD}([\Delta]_U, E) \,\} \\
&\cup \{\, prc_{i,j}(\mathbf{c}, \mathbf{d}) \mid \delta_i(\mathbf{c}) \prec \delta_j(\mathbf{d}) \,\} \\
&\cup \{\, \neg prc_{i,j}(\mathbf{c}, \mathbf{d}) \mid \delta_i(\mathbf{c}) \not\prec \delta_j(\mathbf{d}) \,\} \\
&\cup \{\, con_i(\mathbf{c}) \rightarrow C_j(\mathbf{d}) \mid \delta_j(\mathbf{d}) \prec \delta_i(\mathbf{c}) \,\} \\
&\cup \{\, con_{GD} \rightarrow C_j(\mathbf{d}) \mid \delta_j(\mathbf{d}) \in \mathrm{GD}([\Delta]_U, E) \,\} \\
&\cup \{\, derA_i(\mathbf{c}) \mid A_i(\mathbf{c}) \in E \,\} \\
&\cup \{\, \neg derA_i(\mathbf{c}) \mid A_i(\mathbf{c}) \notin E \,\} \\
&\cup \{\, der\overline{B}_{i,l}(\mathbf{c}) \mid \neg B_{i,l}(\mathbf{c}) \in E \,\} \\
&\cup \{\, \neg der\overline{B}_{i,l}(\mathbf{c}) \mid \neg B_{i,l}(\mathbf{c}) \notin E \,\}
\end{aligned}$$

Note that E' is consistent. Through the deduction theorem and standard interpolation arguments we derive

$$\begin{aligned}
& E' \vdash con_i(\mathbf{c}) \rightarrow A_i(\mathbf{c}) \iff \\
& \iff E' \cup \{\, con_i(\mathbf{c}) \,\} \vdash A_i(\mathbf{c}) \\
& \iff W_D \cup \mathrm{CONS}(\{\, \delta_j(\mathbf{d}) \mid \delta_j(\mathbf{d}) \prec \delta_i(\mathbf{c}) \,\}) \vdash A_i(\mathbf{c}) \\
& \iff A_i(\mathbf{c}) \in E
\end{aligned}$$

In a similar way we obtain

$$\begin{aligned}
& E' \vdash con_{GD} \rightarrow A_i(\mathbf{c}) \iff \\
& \iff W_D \cup \mathrm{CONS}(\{\, \delta_j(\mathbf{d}) \mid \delta_i(\mathbf{c}) \in \mathrm{GD}([\Delta]_U, E) \,\}) \vdash A_i(\mathbf{c}) \\
& \iff A_i(\mathbf{c}) \in E \\
& E' \vdash con_{GD} \rightarrow \neg B_{i,l}(\mathbf{c}) \iff \\
& \iff W_D \cup \mathrm{CONS}(\{\, \delta_j(\mathbf{d}) \mid \delta_i(\mathbf{c}) \in \mathrm{GD}([\Delta]_U, E) \,\}) \vdash \neg B_{i,l}(\mathbf{c}) \\
& \iff \neg B_{i,l}(\mathbf{c}) \in E
\end{aligned}$$

Now the reader may easily verify that

$$E' = W_D \cup \mathrm{CONS}(\mathrm{GD}([\mathrm{pf}(\Delta)]_U, \mathrm{Th}(E')))$$

From this equation and Corollary 2.5 it follows that $\mathrm{Th}(E')$ is an extension of $\mathrm{pf}(T)+D$.

($\Longleftarrow$) Assume that $\mathrm{pf}(T)+D$ has an extension E'. If E' is inconsistent then E' must also be an extension of $T+D$.

Next assume that E' is consistent. Consequently, no instance of (PF 5), (PF 6) and (PF 13)–(PF 15) can be generating for E'. Define

$$G = \{\, \delta_i(\mathbf{c}) \mid gen_i(\mathbf{c}) \in E' \,\}$$
$$E = W_D \cup \mathrm{CONS}(G)$$
$$\delta_i(\mathbf{c}) \prec \delta_j(\mathbf{d}) \equiv prc_{i,j}(\mathbf{c},\mathbf{d}) \in E'$$

(note that $\prec$ must be a strict partial order, otherwise some instance of (PF 5) or (PF 6) would be generating for E'). We will prove that $\mathrm{Th}(E)$ is an extension of $T+D$. By Lemma 2.4, it suffices to show that

(a) $\mathrm{Th}(E) = \mathrm{Th}(W_D \cup \mathrm{CONS}(\mathrm{GD}([\Delta]_U, \mathrm{Th}(E)))$;

(b) for all $\delta_i(\mathbf{c}) \in \mathrm{GD}([\Delta]_U, \mathrm{Th}(E))$,

$$W_D \cup \mathrm{CONS}(\{\, \delta_j(\mathbf{d}) \mid \delta_j(\mathbf{d}) \prec \delta_i(\mathbf{c}) \,\}) \vdash A_i(\mathbf{c})$$

To prove (a), we first show that

$$G = \mathrm{GD}([\Delta]_U, \mathrm{Th}(E)) \tag{C.1}$$

Since (PF 13) cannot be generating for E' we have

$$\begin{aligned}
&\delta_i(\mathbf{c}) \in G \\
&\quad \Longrightarrow (con_i(\mathbf{c}) \rightarrow A_i(\mathbf{c})) \in E' \\
&\quad \Longrightarrow W_D \cup \mathrm{CONS}(\{\, \delta_j(\mathbf{d}) \mid \delta_j(\mathbf{d}) \prec \delta_i(\mathbf{c}) \,\}) \vdash A_i(\mathbf{c}) \quad &&\text{(Lemma 6.4(i))} \\
&\quad \Longrightarrow W_D \cup \mathrm{CONS}(G) \vdash A_i(\mathbf{c}) &&\text{(monotonicity)} \\
&\quad \Longrightarrow \mathrm{Th}(E) \vdash A_i(\mathbf{c}) &&\text{(by def. of } E\text{)}
\end{aligned}$$

Moreover, since (PF 14) cannot be generating for E' we have for all $l = 1, \ldots, k(i)$:

$$\begin{aligned}
&\delta_i(\mathbf{c}) \in G \\
&\quad \Longrightarrow \neg der\overline{B}_{i,l}(\mathbf{c}) \in E' \\
&\quad \Longrightarrow der\overline{B}_{i,l}(\mathbf{c}) \notin E' \\
&\quad \Longrightarrow W_D \cup \mathrm{CONS}(G) \nvdash \neg B_{i,l}(\mathbf{c}) \quad \text{(Lemma 6.4(iv))} \\
&\quad \Longrightarrow \mathrm{Th}(E) \nvdash \neg B_{i,l}(\mathbf{c})
\end{aligned}$$

It follows that $\delta_i(\mathbf{c}) \in G \Longrightarrow \delta_i(\mathbf{c}) \in \mathrm{GD}([\Delta]_U, \mathrm{Th}(E))$. Conversely, since (PF 15) cannot be generating for E' we have

$$\begin{aligned} &\delta_i(\mathbf{c}) \notin G \\ &\Longrightarrow \neg gen_i(\mathbf{c}) \in E' && \text{(through (PF 2))} \\ &\Longrightarrow \neg derA_i(\mathbf{c}) \in E' \text{ or } \exists l \leqslant k(i).\ der\overline{B}_{i,l}(\mathbf{c}) \in E' && \text{((PF 15) is not gen.)} \\ &\Longrightarrow \mathrm{Th}(E) \nvdash A_i(\mathbf{c}) \text{ or } \exists l \leqslant k(i).\ \mathrm{Th}(E) \vdash \neg B_{i,l}(\mathbf{c}) && \text{(Lemma 6.4(iii),(iv))} \\ &\Longrightarrow \delta_i(\mathbf{c}) \notin \mathrm{GD}([\Delta]_U, \mathrm{Th}(E)) \end{aligned}$$

This completes the proof of (C.1). From (C.1) and the definition of E we immediately obtain (a).

We are left to prove (b). We have

$$\begin{aligned} &\delta_i(\mathbf{c}) \in \mathrm{GD}([\Delta]_U, \mathrm{Th}(E)) \\ &\Longrightarrow gen_i(\mathbf{c}) \in E' && \text{(by (C.1) and def. of } G) \\ &\Longrightarrow (con_i(\mathbf{c}) \rightarrow A_i(\mathbf{c})) \in E' && \text{((PF 13) is not generating)} \end{aligned}$$

From this fact and Lemma 6.4(i) we immediately get (b). This completes the proof. □

Appendix C. Proof of Lemmas 6.7 and 6.8

Lemma 6.7. *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 $\mathrm{GD}(\Delta, E) = \mathrm{GD}(\Delta, E_1)$.*

Proof. Assume that E_1 and E_2 satisfy the hypotheses. By Proposition 2.5, we have:

$$E_1 = \mathrm{Th}(W_1 \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E_1))) \tag{C.2}$$

$$\begin{aligned} E_2 &= \mathrm{Th}(W_2 \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E_2))) \\ &= \mathrm{Th}(W_2 \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E_1))) \end{aligned} \tag{C.3}$$

Define $E = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E_1)))$. To prove the lemma it suffices to show that $\mathrm{GD}(\Delta, E) = \mathrm{GD}(\Delta, E_1)$, for then we get $E = \mathrm{Th}(W \cup \mathrm{CONS}(\mathrm{GD}(\Delta, E)))$, which implies that E is an extension of T (by Corollary 2.5).

From (C.2), (C.3) and the hypothesis derive $E_1 \supseteq E \supseteq E_2$. As a consequence,

$$\mathrm{GD}(\Delta, E_1) \subseteq \mathrm{GD}(\Delta, E) \subseteq \mathrm{GD}(\Delta, E_2)$$

Since by hypothesis $\mathrm{GD}(\Delta, E_2) = \mathrm{GD}(\Delta, E_1)$, it follows that $\mathrm{GD}(\Delta, E) = \mathrm{GD}(\Delta, E_1)$. □

Lemma 6.8. *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* $\mathrm{GD}(\Delta,E) = \mathrm{GD}(\Delta,E_1)$.

Proof. Assume that E_1 and E_2 satisfy the hypotheses and define $E = \mathrm{Th}(\{\phi_1 \vee \phi_2\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E_1)))$. It suffices to show that $\mathrm{GD}(\Delta,E) = \mathrm{GD}(\Delta,E_1)$ because this implies $E = \mathrm{Th}(\{\phi_1 \vee \phi_2\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E)))$, which proves that E is an extension of T (by Corollary 2.5).

First note that $E \subseteq \mathrm{Th}(\{\phi_1\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E_1))) = E_1$, therefore $\mathrm{GD}(\Delta,E) \supseteq \mathrm{GD}(\Delta,E_1)$.

We are left to show that $\mathrm{GD}(\Delta,E) \subseteq \mathrm{GD}(\Delta,E_1)$. For this purpose, let $(:\alpha/\beta)$ be any prerequisite-free default such that $(:\alpha/\beta) \notin \mathrm{GD}(\Delta,E_1)$; we have to show that $(:\alpha/\beta) \notin \mathrm{GD}(\Delta,E)$.

Since $\mathrm{GD}(\Delta,E_2) = \mathrm{GD}(\Delta,E_1)$ we have both $(:\alpha/\beta) \notin \mathrm{GD}(\Delta,E_1)$ and $(:\alpha/\beta) \notin \mathrm{GD}(\Delta,E_2)$; this means that

$$\{\phi_1\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E_1)) \vdash \neg\alpha \tag{C.4}$$

$$\{\phi_2\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E_2)) \vdash \neg\alpha \tag{C.5}$$

and hence, by the above equality

$$\{\phi_2\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E_1)) \vdash \neg\alpha \tag{C.6}$$

From (C.4) and (C.6) we get $\{\phi_1 \vee \phi_2\} \cup \mathrm{CONS}(\mathrm{GD}(\Delta,E_1)) \vdash \neg\alpha$, hence $(:\alpha/\beta) \notin \mathrm{GD}(\Delta,E)$, which completes the proof. □

Appendix D. Proof of Lemma 7.7

Lemma 7.7. *If a property* **P** *of unbounded databases can be expressed by an everywhere consistent autoepistemic theory T under the brave semantics, then* **P** *can also be expressed by a prerequisite-free normal autoepistemic theory T' under the brave semantics.*

Proof. We may assume without loss of generality that T is in autoepistemic normal form. Construct T' from T as follows. Replace each formula

$$\rho = L\alpha_1(\mathbf{x}_1) \wedge \cdots \wedge L\alpha_m(\mathbf{x}_m) \wedge \neg L\beta_1(\mathbf{y}_1) \wedge \cdots \wedge \neg L\beta_n(\mathbf{y}_n) \rightarrow \phi$$

of T with the formulas

$$\begin{aligned}
\rho_1 &= \neg L p_{\alpha_1}(\mathbf{x}_1) \wedge \cdots \wedge \neg L p_{\alpha_m}(\mathbf{x}_m) \wedge \neg L\beta_1(\mathbf{y}_1) \wedge \cdots \wedge \neg L\beta_n(\mathbf{y}_n) \rightarrow \phi \\
\rho_2 &= \neg L\alpha_1(\mathbf{x}_1) \rightarrow p_{\alpha_1}(\mathbf{x}_1) \\
&\ \ \vdots \\
\rho_{m+1} &= \neg L\alpha_m(\mathbf{x}_m) \rightarrow p_{\alpha_m}(\mathbf{x}_m)
\end{aligned}$$

where each formula α_i is associated with a distinct predicate p_{α_i} that occurs neither in T nor in $\overline{R}$.[15] Intuitively, $p_{\alpha_i}(\mathbf{c})$ should be derivable iff the corresponding sentence $\alpha_i(\mathbf{c})$ is not. Let T_1 be the set of all transformed rules ρ_1 and let T_2 be the set of all new rules $\rho_2, \ldots, \rho_{m+1}$, for all $\rho \in T$. Define $T' = T_1 \cup T_2$.

First we show that the transformed theory "recognizes" all the databases that have the property $\mathbf{P}$, that is, for all databases $D = (U, \phi)$ such that the theory $T + D = [T]_U \cup \{\phi\}$ has a stable expansion, we shall prove that the corresponding theory $T' + D = [T']_U \cup \{\phi\}$ has a stable expansion, too. For this purpose, assume that $T + D$ has a stable expansion and let E be its kernel. By Theorem 2.8, $E = \mathrm{Th}((T+D)^E)$. Define

$$E' = \mathrm{Th}((T+D)^E \cup \{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\})$$

We claim that E' must be consistent.

> In fact, by hypothesis, T is everywhere consistent, which implies that E must be consistent – and so must be $(T+D)^E$. Since the predicates p_α do not occur in $(T+D)^E$, every model of $(T+D)^E$ can be extended with the set $\{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\}$ without changing the truth-value of $(T+D)^E$; obviously, the resulting interpretation is a model of E'.

Since E' is consistent and the languages of $(T+D)^E$ and $\{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\}$ are disjoint, we derive from the definition of E' that

$$p_\alpha(\mathbf{c}) \in E' \iff p_\alpha(\mathbf{c}) \in \{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\} \iff \alpha(\mathbf{c}) \notin E.$$

Similarly, for all formulas $L\alpha(\mathbf{x})$ and $\neg L\beta(\mathbf{x})$ that occur in T:

$$\alpha(\mathbf{c}) \in E' \iff \alpha(\mathbf{c}) \in \mathrm{Th}((T+D)^E) \iff \alpha(\mathbf{c}) \in E,$$

$$\beta(\mathbf{c}) \in E' \iff \beta(\mathbf{c}) \in \mathrm{Th}((T+D)^E) \iff \beta(\mathbf{c}) \in E.$$

From the above observations, it follows that if the head of some instantiated rule $\rho\theta \in [T]_U$ must be put in $[T]_U^E$, then the head of the corresponding rule $\rho_1\theta \in [T_1]_U$ must be put in $[T_1]_U^{E'}$ and vice versa; therefore, $[T]_U^E = [T_1]_U^{E'}$.

By similar considerations we derive also that $\{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\} = [T_2]_U^{E'}$. Then we get

$$\begin{aligned}
E' &= \mathrm{Th}((T+D)^E \cup \{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\}) \\
&= \mathrm{Th}(([T]_U \cup \{\phi\})^E \cup \{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\}) \\
&= \mathrm{Th}([T]_U^E \cup \{\phi\} \cup \{\, p_\alpha(\mathbf{c}) \mid \alpha(\mathbf{c}) \notin E \,\}) \\
&= \mathrm{Th}([T_1]_U^{E'} \cup \{\phi\} \cup [T_2]_U^{E'}) \\
&= \mathrm{Th}([T']_U^{E'} \cup \{\phi\})
\end{aligned}$$

[15] We assume that the language contains countably many predicate symbols.

$$= \mathrm{Th}(([T']_U \cup \{\phi\})^{E'})$$

$$= \mathrm{Th}((T' + D)^{E'})$$

It follows that $\mathrm{E}(E')$ is a stable expansion of $(T' + D)$.

We are left to show that the transformation is sound, that is, for all databases $D = (U, \phi)$ such that $(T' + D)$ has a stable expansion, the theory $(T + D)$ has a stable expansion, too. For this purpose, assume that $(T' + D)$ has a stable expansion and let E' be its kernel. We claim that E' is consistent.

Suppose not. Then $(T' + D)$ has an inconsistent stable expansion. By Lemma 7.2, the corresponding default theory, $\langle W, \Delta\rangle = \mathrm{tr}^{-1}(T' + D)$, has an inconsistent extension. Then, by a well-known property of default theories, W must be inconsistent and hence $(T' + D)_0$, which equals W, must be inconsistent, too. But $(T' + D)_0 = (T + D)_0$ (because the transformation $T \rightsquigarrow T'$ does not affect ordinary formulas); it follows that $(T + D)$ has an inconsistent stable expansion, which contradicts the hypothesis that T is everywhere consistent.

From Theorem 2.8 and the definition of T', we have

$$E' = \mathrm{Th}((T' + D)^{E'}) = \mathrm{Th}([T_1]_U^{E'} \cup [T_2]_U^{E'} \cup \{\phi\})$$

From the consistency of E' and from the fact that the predicates p_α occur in $[T_2]_U^{E'}$ but not in $[T_1]_U^{E'}$ nor in ϕ, it is not difficult to prove that

$$p_\alpha(\mathbf{c}) \in E' \iff p_\alpha(\mathbf{c}) \in [T_2]_U^{E'} \iff \alpha(\mathbf{c}) \notin E',$$

$$\alpha(\mathbf{c}) \in E' \iff \alpha(\mathbf{c}) \in \mathrm{Th}([T_1]_U^{E'} \cup \{\phi\}),$$

$$\beta(\mathbf{c}) \in E' \iff \beta(\mathbf{c}) \in \mathrm{Th}([T_1]_U^{E'} \cup \{\phi\})$$

for all formulas $L\alpha$ and $\neg L\beta$ that occur in T. From the first of the above observations, it follows that the epistemic Gelfond–Lifschitz transformation of any rule $\rho_1\theta \in [T_1]_U$ and of the corresponding rule $\rho\theta \in [T]_U$ yield the same result, and hence

$$[T_1]_U^{E'} = [T]_U^{E'}.$$

From this fact and from the remaining observations we get

$$\alpha(\mathbf{c}) \in E' \iff \alpha(\mathbf{c}) \in \mathrm{Th}([T]_U^{E'} \cup \{\phi\}) = \mathrm{Th}((T + D)^{E'}),$$

$$\beta(\mathbf{c}) \in E' \iff \beta(\mathbf{c}) \in \mathrm{Th}([T]_U^{E'} \cup \{\phi\}) = \mathrm{Th}((T + D)^{E'}).$$

Now define $E = \mathrm{Th}((T + D)^{E'})$. By the above two facts we have

$$\alpha(\mathbf{c}) \in E' \iff \alpha(\mathbf{c}) \in E,$$

$$\beta(\mathbf{c}) \in E' \iff \beta(\mathbf{c}) \in E,$$

consequently, the epistemic Gelfond–Lifschitz transformation of $T + D$ w.r.t. E' and E yields the same results, i.e., $(T + D)^{E'} = (T + D)^E$. It follows immediately that

$E = \mathrm{Th}((T+D)^E)$, and hence $\mathrm{E}(E)$ is a stable expansion of $(T+D)$. This completes the proof. $\square$

References

[1] S. Abiteboul, P. Kanellakis and G. Grahne, On the representation and querying of sets of possible worlds, *Theoret. Comput. Sci.* **78** (1991) 159–187.

[2] S. Abiteboul and V. Vianu, Expressive power of query languages, in: J. Ullman, ed., *Theoretical Studies in Computer Science* (Academic Press, New York, 1992). Festschrift in Honour of Seymour Ginsburg's 64th Birthday.

[3] P. Bonatti, Autoepistemic logic programming, *J. Automat. Reasoning* **13** (1994) 35–67.

[4] P. Bonatti, Autoepistemic logics as a unifying framework for the semantics of logic programs. *J. Logic Programming* **22** (1995) 91–149.

[5] M. Cadoli, T. Eiter and G. Gottlob, Using default logic as a query language, in: *Proc. 4th Internat. Conf. on Principles of Knowledge Representation and Reasoning* (*KR-94*) (1994) 99–108.

[6] A.K. Chandra, Theory of database queries, in: *Proc. PODS-88* (1988).

[7] A. Chandra and D. Harel, Horn clause queries and generalizations, *Journal of Logic Programming* **2** (1985) 1–15.

[8] T. Eiter, G. Gottlob and H. Mannila, Adding disjunction to datalog, in: *Proc. 13th ACM SIGACT SIGMOD-SIGART Symp. on Principles of Database Systems* (*PODS-94*) (1994) 267–278.

[9] D.W. Etherington, *Reasoning with Incomplete Information* (Morgan Kaufmann, Los Altos, 1988).

[10] R. Fagin and J.D. Ullman, On the semantics of updates in databases, in: *Proc. of PODS-83* (1983) 352–365.

[11] J. Fernández and J. Minker, Semantics of disjunctive deductive databases, in: *Proc. ICDT-92*, Berlin (1992) 21–50.

[12] M. Gelfond and V. Lifschitz, Classical negation in logic programs and disjunctive databases, *New Generation Comput.* **9** (1991) 365–385.

[13] G. Gottlob, Complexity results for nonmonotonic logics, *J. Logic Comput.*, **2** (1992) 397–425.

[14] G. Grahne, *The Problem of Incomplete Information in Relational Databases*, Lecture Notes in Computer Science, Vol. 554 (Springer, Berlin, 1991).

[15] G. Grahne, Updates and counterfactuals, in: *Proc. 2nd Internat. Conf. on Principles of Knowledge Representation and Reasoning (KR-91)*, (1991) 269–276.

[16] S. Greco, D. Saccà and C. Zaniolo, DATALOG queries with stratified negation and choice, in: *Proc. 5th Internat. Conf. on Database Theory* (*ICDT'95*), Lecture Notes in Computer Science, Vol. 893 (Springer, Berlin, 1995) 82–96.

[17] Y. Gurevich, Logic and the challenge of computer science, in: E. Börger, ed., *Trends in Theoretical Computer Science* Ch. 1 (Computer Science Press, Rockville, MD, 1988).

[18] N. Immerman, Languages that capture complexity classes, *SIAM J. Comput.* **16** (1987) 760–778.

[19] D.S. Johnson, M. Yannakakis and C.H. Papadimitriou, On generating all maximal independent sets, *Inform. Processing Lett.* **27** (1988) 119–123.

[20] P. Kanellakis, Elements of relational database theory, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, Vol. B, Ch. 17 (Elsevier, Amsterdam, 1990).

[21] H. Kautz and B. Selman, Hard problems for simple default logics, *Artif. Intelligence* **49** (1991) 243–279.

[22] P. Kolaitis and M. Vardi, On the expressive power of datalog: tools and a case study, in: *Proc. PODS-90* (1990) 61–71.

[23] K. Konolige, On the relationship between default and autoepistemic logic, *Artif. Intelligence* **35** (1988) 343–382, + Errata, same journal, **41** (1989/90) 115.

[24] K. Konolige, On the relation between autoepistemic logic and circumscription, in: *Proc. IJCAI-89* (1989).

[25] H. Levesque, All i know: A study in autoepistemic logic, *Artif. Intelligence* **42** (1990) 263–309.

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

[27] W. Lukasiewicz, *Non-Monotonic Reasoning* (Ellis Horwood, Chichester, England, 1990).

[28] W. Marek and M. Truszczyński, Autoepistemic logic, *J. ACM* **38** (1991) 588–619.

[29] W. Marek and M. Truszczyński, *Nonmonotonic Logics – Context-Dependent Reasoning* (Springer, Berlin, 1993).

[30] R. Moore, Semantical considerations on nonmonotonic logics, *Artif. Intelligence* **25** (1985) 75–94.

[31] T. Przymusinski, On the declarative and procedural semantics of stratified deductive databases, in: J. Minker, ed., *Foundations of Deductive Databases and Logic Programming* (Morgan Kaufman, Washington, DC, 1988) 193–216.

[32] T. Przymusinski, Stable semantics for disjunctive programs, *New Generation Comput.* **9** (1991) 401–424.

[33] T. Przymusinski, Three-valued nonmonotonic formalisms and semantics of logic programming, *Artif. Intelligence* **49** (1991) 309–344.

[34] R. Reiter, A logic for default reasoning, *Artif. Intelligence* **13** (1980) 81–132.

[35] J. Schlipf, The expressive powers of logic programming semantics, Tech. Report CIS-TR-90-3, Computer Science Department, University of Cincinnati, 1990. Preliminary version in *Proc. PODS-90* (1990) 196–204. To appear in *J. Comput. System Sci.*

[36] J. Stillman, The complexity of propositional default logic, in: *Proc. AAAI-92* (1992) 794–799.

[37] L. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8** (1979) 410–421.

[38] M. Vardi, Complexity of relational query languages, in: *Proc. 14th STOC* (1982) 137–146.

[39] M. Vardi, Querying logical databases, *J. Comput. and System Sci.* **33** (1986) 142–160.

[40] M. Winslett, *Updating Logical Databases* (Cambridge Univ. Press, Cambridge, 1990).