JOURNAL OF COMPUTER AND SYSTEM SCIENCES 38, 290–325 (1989)

Minimalism, Justification and Non-monotonicity in Deductive Databases

NICOLE BIDOIT*

Laboratoire de Recherche en Informatique, U.A. 410 du CNRS. "al Khowarizmi," Université Paris-Sud, 91405 Orsay, Cedex, France

AND

RICHARD HULL†

Computer Science Department, University of Southern California, Los Angeles, California 90080-0782

Received October 26, 1986; revised September 1987

Three formalizations of the closed world assumption (CWA) which accomodate disjunctive information are compared. The semantic approach of Bossu and Siegel, here called "minimalism," is shown to be equivalent to the syntactic approach based on Reiter's "default logic," when a specific class of defaults corresponding to the CWA is used. Neither approach generalizes the "negation as failure" (NAF) inference rule of Clark. The three formalizations are synthesized to form "positivism," a new semantically defined formalization of the CWA. The expressive power of minimalism and positivism is compared, in both static and dynamic contexts. In the dynamic case the comparison shows that positivism and minimalism are "non-monotonic" in different ways. Finally, positivism and "stratification," an alternative formalization of the CWA which combines minimalism and NAF are briefly compared.

1. INTRODUCTION

Proposals for the theoretical foundations of databases have ranged along a spectrum from the essentially algebraic [Co] to various formalizations within the realm of first-order logic [GMN, R4]. It is widely accepted that when using the logic point of view, the closed world assumption (CWA) must be incororated at a fundamental level. For databases consisting of atomic facts [R4] or Horn clauses [Cl, R1], relatively simple proof-theoretic formalizations of the CWA have been obtained. However, it is desirable to utilize the expressive power of the logic

* Initial work by this author performed while supported by INRIA and visiting the University of Southern California.

† Work of this author supported in part by NSF Grants IST-83-06517 and IST-85-11541. Work by this author performed in part while visiting and supported by LRI, University of Paris XI at Orsay.

0022-0000/89 $3.00

approach for databases by permitting negation in the premises of clauses as in PROLOG [CIT], and also permitting general forms of disjunctive information [GMN]. Until now, formalizations of the CWA have captured only one or the other of these features. The purpose of this paper is twofold: first, we introduce and study "positivism," a model-theoretic formalization of the CWA which captures both features simultaneously; second, we develop techniques for comparing the expressive power of different formalizations of the CWA by comparing their non-monotonic behavior.

In the paper, we begin by examining two recent formalizations of the CWA for incomplete databases. The first [BS], here called "minimalism," is model-theoretic, and is defined using a partial ordering on the first-order models of a set of formulas. The second is proof-theoretic, and is a restricted case of the "default logic" proposed by Reiter [R2]. As shown in Section 4, these two approaches are equivalent in the context of deductive databases. Turning to the semantics of the CWA when clauses with negative premises are allowed, in Section 5 we introduce *justification*[1] a semantic analog of Clark's "negation as failure" (NAF) inference rule [Cl]. This is combined with minimalism to define positivism. An important result (Theorem 5.10) shows that the semantics of positivism coincides with that of NAF (more precisely, with the semantics of Clark's "completed" database).

A major result of this paper (Section 6) shows that in the context of simple insertions, the expressive power of the positivism approach (and that of NAF, if defined) is strictly greater than the expressive power of the minimalism approach. This result is especially interesting because in the static case, assuming domain closure, minimalism and positivism can simulate the other. Thus, in the context of deductive databases, positivism and NAF are distinguished from minimalism by their different non-monotonic behavior.

Positivism provides a completely nonprocedural semantics for deductive databases which use negation in premises of rules. In particular, as noted above it provides a model-theoretic semantics corresponding to the procedural semantics of NAF. A second fundamental motivation for introducing positivism is to permit the study of NAF and related semantics within a model-theoretic setting. Several results in the paper utilize this framework to provide new insights into the area. Perhaps the most interesting of these is that, speaking intuitively, in the absence of domain closure minimalism cannot simulate the "spirit" of NAF (Theorem 6.4).

Section 2 presents several examples which motivate and illustrate the different concepts of the paper. The formal discussion starts in Section 3 which provides a review of preliminary definitions. As noted above, Sections 4, 5, and 6 are devoted to the main results of the paper. Finally, Section 7 briefly compares positivism with "stratification" [ABW, P1], an alternative formalization of the CWA which combines minimalism and NAF.

[1] Essentially, the same notion is called *supported* in [ABW], where it was introduced independently and simultaneously.

2. Motivating Examples

This section provides informal illustrations of most of the major themes of the paper. First, a simple example illustrating how minimalism is used to combine disjunctive information with the CWA is presented. Next, an example illustrating the principle of NAF [Cl] is presented. The examples are then combined to illustrate how positivism provides a natural synthesis of these two approaches. Finally, the non-monotonic behavior of both minimalism and positivism are considered. The discussion here is informal, and several assumptions will be made implicitly throughout. A formal development parallelling the discussion of this section is presented in the remainder of the paper.

Throughout this section, we assume that the first-order languages considered have only finitely many constant symbols and no function symbols.

We begin with an example which is used to review the general approach taken in deductive databases and then to illustrate the particular approach of minimalism.

Example 2.1. *Minimalism.* Consider the database shown in Fig. 2.1. Here we show three one-column relations: STUD which holds the names of all students; UGRAD which holds names of undergraduate students; and GRAD which holds the names of graduate students. The inference rule

$$\sigma_1 = \mathrm{STUD}(x) \rightarrow \mathrm{UGRAD}(x) \vee \mathrm{GRAD}(x)$$

is also included in this database. Let Σ_1 denote the set of first-order predicate calculus sentences corresponding to the atomic facts in this database, along with the universally quantified sentence $\forall x(\sigma_1)$.

STUD	GRAD	UGRAD
Babek	Dong	Babek
Dong	Mary	
Mary	Toto	
Toto		
Zanja		

$\sigma_1 = \mathrm{STUD}(x) \rightarrow \mathrm{UGRAD}(x) \vee \mathrm{GRAD}(x)$

Fig. 2.1. Database illustrating minialism.

In the formal approach taken by deductive databases, a sentence σ is true in the database of Example 2.1 if it is implied (in some specially defined sense) by Σ_1. For example, in a *model-theoretic* approach, a set $\mathcal{M}^{xxx}(\Sigma_1)$ of *models* is associated with Σ_1, and σ is defined to be true in the database if each model in $\mathcal{M}^{xxx}(\Sigma_1)$ satisfies σ. In a *proof-theoretic* approach, on the other hand, a set of first-order theories $\mathcal{T\!h}^{xxx}(\Sigma_1)$ is associated with Σ_1, and σ is viewed as true if Ω proves σ for each theory $\Omega \in \mathcal{T}\!h^{xxx}(\Sigma_1)$ (or equivalently, if **I** saisfies σ for each model **I** of some $\Omega \in \mathcal{T}\!h^{xxx}(\Sigma_1)$).

Continuing with the general discussion of deductive databases, we recall that it is common to assume axioms which express fundamental properties of (relational) databases in the context of logic [R3]. These typically include a domain closure (DC) axiom, which in this case might be:

$$\forall x(x \simeq \text{Babek} \vee \cdots \vee x \simeq \text{Zanja}),$$

expressing that each object of the real world is identifiable by a constant symbol; and also a unique name (UN) axiom [R3], for example:

$$(\text{Babek} \not\simeq \text{Dong}) \wedge (\text{Babek} \not\simeq \text{Mary}) \wedge \cdots$$
$$\cdots (\text{Mary} \not\simeq \text{Zanja}) \wedge (\text{Toto} \not\simeq \text{Zanja}),$$

expressing that two distinct constant symbols identify two distinct objects of the real world. We assume DC and UN for the remainder of the informal discussion in this section.

Speaking in terms of the model theoretic approach, each model of the database Σ_1 of Example 2.1 will (essentially) have exactly five elements in its universe of discourse. It is thus natural to blur the distinction between the constants appearing in Σ_1 and the corresponding elements in first order models of Σ_1. (The notion of *Herbrand models* provides a formal counterpart of this.) Indeed, under these assumptions the set Σ_1 has 48 models in conventional first-order predicate calculus. This includes several models in which one or more of the students is in both the GRAD and UGRAD relations.

We turn now to the approach of *minimalism* due to [BS] (see also [Mi, Mc1]). Under this approach, the family of all first order models of Σ_1 is partially ordered by a relation $\leqslant$, which can be informally defined as follows: models **I** and **I**$'$ satisfy $\mathbf{I} \leqslant \mathbf{I}'$ if each positive atomic fact true in **I** is also true in **I**$'$. Speaking roughly, $\mathbf{I} \leqslant \mathbf{I}'$ if **I** contains fewer true facts than **I**$'$. The intuition behind the CWA is now captured by focussing on the set $\mathcal{M}^{\min}(\Sigma_1)$ of models which are minimal with respect to $\leqslant$. Restricting one's attention to minimal models of Σ_1 has the effect of reducing as much as possible the number of true facts contained in a potential state of the database. For example, it is easily seen that in each model of $\mathcal{M}^{\min}(\Sigma_1)$, each student is a member of GRAD or UGRAD but not both. From the included atomic facts, it is clear that in each minimal model GRAD(Dong), GRAD(Mary), GRAD(Toto), and UGRAD(Babek) are true. It follows that (under the assumptions of the preceding paragraph) there are only two minimal models of Σ_1: in one GRAD(Zanja) is true, and in the other UGRAD(Zanja) is true. As a result, we see that, for example,

$$\Sigma_1 \models^{\min} \text{GRAD(Zanja)} \vee \text{UGRAD(Zanja)};$$

$$\Sigma_1 \not\models^{\min} \text{GRAD(Zanja)}; \text{ and}$$

$$\Sigma_1 \models^{\min} \forall x(\neg\text{GRAD(Zanja)} \vee \neg\text{UGRAD(Zanja)}).$$

As noted in the Introduction, a second formalization of the CWA which permits disjunctive information will be discussed in this paper. This second approach is proof theoretic and is based on the default logic of Reiter [R2]. As shown in Section 4, this approach is equivalent to minimalism when relativized to deductive databases. Although not discussed in detail here, a number of approaches [Mi, Li1, GPP] have been proposed in the literature to formalize the CWA in presence of disjunction. Whereas the GCWA [Mi] just makes use of minimal models, a direct correspondence between the ECWA [GPP] and circumscription [Li1] has been demonstrated.

A significant limitation of minimalism is that it does not interpret negative premises in clauses according to the principle of NAF. To illustrate this, we present an example of NAF and then show why minimalism fails to capture the intuition behind it.

EXAMPLE 2.2. *Negation as failure.* In Fig. 2.2 a database focussing on graduate students is shown. As suggested by the inference rule

$$\sigma_2 = \mathrm{GRAD}(x) \wedge \neg \mathrm{TA}(x) \rightarrow \mathrm{ADV\text{-}GRAD}(x),$$

each graduate student who is not a teaching assistant is an advanced graduate student (and thus, for example, supported by a research grant). As in Example 2.1, let Σ_2 be the set of first-order predicate calculus sentences corresponding to the atomic facts in this database, along with $\forall x(\sigma_2)$.

GRAD	TA	ADV-GRAD
Dong	Mary	Mary
Mary	Toto	
Toto		

$$\sigma_2 = \mathrm{GRAD}(x) \wedge \neg\, \mathrm{TA}(x) \rightarrow \mathrm{ADV\text{-}GRAD}(x).$$

FIG. 2.2. Database illustrating negation as failure.

The approach of NAF is essentially proof-theoretic. Although the truth of a sentence σ in a database Σ is defined in terms of an algorithm (see Section 5), Clark has shown [Cl] that the NAF rule is sound in the following sense: if σ has been infered from Σ using the NAF algorithm, then σ is a logical consequence of a specific theory Σ' associated with Σ. Although not discussed in detail here, it is important for the reader to know that (in general) the NAF rule is not complete with respect to Clark's completed theory. This problem is one among others presented and discussed in detail in [L1, S1, S2].

Under NAF, a positive atomic formula is assumed false unless there is a specific reason forcing it to be true. Thus, in the database Σ_2, GRAD(Dong) is true under NAF because it is in Σ_2. On the other hand, TA(Dong) is false under NAF because it is not in Σ_2, and it cannot be deduced from any rule in Σ_2. (Indeed, TA(x) does not appear in the consequent of any rule of Σ_2.) Finally, in virtue of σ_2, we see that under NAF, ADV-GRAD(Dong) is true. In contrast, because TA(Toto) is true, ADV-GRAD(Toto) is not true. (Note that nothing is violated by the presence of both TA(Mary) and ADV-GRAD(Mary) in Σ_2.)

As is apparent from the above discussion, under NAF he rule σ_2 plays a dual role. First, it guarantees for each graduate student x that either TA(x) or ADV-GRAD(x). Furthermore, it provides a priority on these two possibilities: unless otherwise specified a given graduate student is assumed to be an advanced-graduate. This ability to express priorities is a fundamental capability of the NAF approach which minimalism does not provide. Speaking loosely, this is because under first-order logic, and hence under minimalism, rule σ_2 is equivalent to both of the following clauses (considered separately or together).

$$\mathrm{GRAD}(x) \wedge \neg\, \mathrm{ADV\text{-}GRAD}(x) \rightarrow \mathrm{TA}(x)$$

$$\mathrm{GRAD}(x) \rightarrow \mathrm{TA}(x) \vee \mathrm{ADV\text{-}GRAD}(x);$$

In particular, because minimalism is quite close to first-order logic, it interprets a negative premise $\neg P(\mathbf{x})$ as if $P(\mathbf{x})$ was included as a disjunct in the consequent.

It should be noted that NAF can be used to handle simple cases of exceptions. For example, under NAF the database containing

$$\mathrm{BIRD}(x) \wedge \neg\, \mathrm{EXC}(x) \rightarrow \mathrm{FLY}(x)$$

and

$$\mathrm{PENQUIN}(x) \rightarrow \mathrm{EXC}(x)$$

expresses the fact that all birds fly, except penquins.

The purpose of *positivism*, introduced in this paper, is to combine the advantages of minialism, which permits disjunctive consequents of rules, with NAF, which provides a mechanism for expressing simple priorities on these disjuncts. This is accomplished by augmenting the model-theoretic approach of minimalism with the new notion of *justification*, a model-theoretic generalization of NAF (Theorem 5.4). These notions are now illustrated.

EXAMPLE 2.3. *Positivism.* Let Σ_3 be the set of first order sentences corresponding to the database shown in Fig. 2.3. Speaking informally, σ_1 suggests that each student is either a graduate or an undergraduate student, but no priority is given; and σ_2 suggests that each graduate student is either a TA or an advanced-graduate student, and unless otherwise specified a graduate student is assumed to be an advanced-graduate.

STUD	GRAD	UGRAD	TA	ADV-GRAD
Babek	Dong	Babek	Mary	Mary
Dong	Mary	Toto		
Mary	Toto			
Toto				
Zanja				

$$\sigma_1 = \mathrm{STUD}(x) \rightarrow \mathrm{UGRAD}(x) \vee \mathrm{GRAD}(x)$$

$$\sigma_2 = \mathrm{GRAD}(x) \wedge \neg\, \mathrm{TA}(x) \rightarrow \mathrm{ADV\text{-}GRAD}(x).$$

FIG. 2.3. Database illustrating positivism.

Speaking loosely, a first-order model $\mathbf{I}$ of Σ_3 is *justified* if each positive atomic fact $P(\mathbf{d})$ which is true in $\mathbf{I}$ is justified by Σ_3 in $\mathbf{I}$, in the sense that $P(\mathbf{d})$ occurs (modulo an assignment of variables to domain elements) in the consequent of a rule of Σ_3, all of whose premises are true in I. (Atomic ground formulas such as STUD (Babek) are viewed as degenerate rules with no premises.) For example, consider the model $\mathbf{I}_1$, where the true positive atomic facts include only the positive atomic ground formulas of Σ_3 and UGRAD(Zanja), ADV-GRAD(Dong). Then $\mathbf{I}_1$ is a justified model of Σ_3: it satisfies Σ_3 in first-order logic, and each positive atomic fact in $\mathbf{I}_1$ is justified by an atomic fact or rule of Σ_3. (In particular, UGRAD(Zanja) is justified by σ_1 and the fact that STUD(Zanja) is true in $\mathbf{I}_1$; and ADV-GRAD (Dong) is justified by σ_2 and the facts that GRAD(Dong) and $\neg$TA(Dong) are true in $\mathbf{I}_1$.)

It is easily seen that none of TA(Babek), TA(Dong), or TA(Zanja) are true in any justified model of Σ_3. This indicates how justification forms a model-theoretic generalization of NAF. On the other hand, the notion of justification alone does not entirely capture some aspects of the CWA captured by minimalism. For example, a second justified model of Σ_3 is $\mathbf{I}_2$, which includes the positive atomic ground formulas of Σ_3 and {UGRAD(Zanja), GRAD(Zanja), ADV-GRAD(Zanja), ADV-GRAD(Dong)}. Here, both UGRAD(Zanja) and GRAD(Zanja) are justified by σ_2 in $\mathbf{I}_2$ because STUD(Zanja) is true in $\mathbf{I}_2$. Thus, while the notion of justification insists that each positive atomic ground formula true in a model is justified by an atomic fact or rule, it does not prevent more than one disjunct of a consequent from being true; that is, it does not minimize the set of true facts. It is easily verified that Σ_3 has a total of 48 justified models (assuming DC and UN as before).

In order to capture more of the intuition of the CWA, the notion of positivism is defined by combining minimalism and justification. Thus, a positivistic model is a justified model which is minimal (among first-order logic models) with respect to $\leqslant$. It is easily verified that there are two positivistic models of Σ_3, one in which UGRAD(Zanja) is true, and one in which GRAD(Zanja) and ADV-GRAD(Zanja) are true. Thus, positivism combines the advantages of minimalism with NAF in a formal, model-theoretic approach.

Before turning to the non-monotonic behavior of minimalism and positivism, we present another example illustrating some of the expressive power of positivism not available in either minimalism or NAF. Consider the rule:

$$\text{PREREQ}(c_1, c_2) \wedge \text{TAKING}(s, c_2) \wedge \neg\text{WAIVED}(s, c_1)$$
$$\rightarrow \text{TOOK}(s, c_1) \vee \text{TESTED-OUT}(s, c_1).$$

This rule states that if a course c_1 is a prerequisite of a course c_2 and student s is taking c_2, then either s took c_1, s tested out of c_1, or s was waived out of c_1. Under positivism, a priority is given to these possibilities: unless otherwise indicated, it is assumed that s either took or tested out of c_1.

A fundamental difference between first order logic and deductive databases incorporating the CWA is that deductive databases are non-monotonic, in the sense that the inclusion of additional information may negate information previously inferred. For example, referring to the database of Example 2.3, suppose that Ninja is a constant in the language. Then under both minimalism and positivism, Σ_3 implies $\neg$STUD(Ninja), $\neg$UGRAD(Ninja), etc. However, if STUD(Ninja) is added to the database, then $\Sigma_3 \cup \{\text{STUD(Ninja)}\}$ implies STUD(Ninja) and no longer implies $\neg$STUD (Ninja) or $\neg$GRAD(Ninja). Thus, the insertion of a new fact causes the deletion of previously inferred facts.

As shown in Section 6, the non-monotonic behavior of minimalism and positivism are fundamentally different (Theorem 6.9). To illustrate this, recall that under positivism, Σ_3 implies ADV-GRAD(Dong). If the atomic fact TA(Dong) is added to the database, then it no longer implies ADV-GRAD(Dong). It can be shown that there is no set Σ' of clauses such that the minimal models of Σ' are the positivistic models of Σ_3 **and** the minimal models of $\Sigma' \cup \{\text{TA(Dong)}\}$ are the positivistic models of $\Sigma_3 \cup \{\text{TA(Dong)}\}$. Thus, in the presence of simple insertions, no minimalist database can simulate the positivist database Σ_3. (In Section 6, we also show that minimalism cannot simulate the "spirit" of NAF.) This is of particular interest because in the absence of updates, both minimalism and positivism can simulate each other under the domain closure assumption (Proposition 6.2 and Theorem 6.3). More generally, this indicates how differences in non-monotonic behavior can sometimes be used to formally distinguish the expressive power of different formalizations of the CWA.

3. Preliminaries

In the following, we assume that the reader is familiar with symbolic logic and more precisely with propositional logic and first-order logic [E]. However, in order to make the discussion clear, we begin by reviewing some well-known concepts of first-order logic and present the main notations used throughout the paper.

A *first-order alphabet* A is a set of five types of symbols: *constant* symbols (finitely many) denoted $a, b, c, \ldots$; *variable* symbols denoted $x, y, z, \ldots$; *predicate* symbols denoted P, Q, R, ..., each of them associated with an integer, called the *arity* of the predicate, and $\mathbf{0}$, $\mathbf{1}$ are among the 0-ary predicates; the *logical* symbols $\forall$ (universal quantifier), $\exists$ (existential quantifier), $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$: and the *punctuation* symbols ( ). (Because our focus is on deductive databases, we do not include function symbols in our alphabets.)

We denote formulas by α, β, γ, ...; sets of formulas by Σ, Γ, Ω, ... The set of all formulas over A is the *first-order language* over A. For this paper, we generally assume that a specific alphabet A is fixed and let $\mathscr{L}$ denote the first-order language over this alphabet.

If α is an atomic formula, then α and $\neg\alpha$ are *literals*, respectively *positive literal* and *negative literal*. A formula is *ground* if it does not contain variable symbols. We assume that the reader is familiar with the notions of *bound* and *free* occurrences of variables. A formula α is *closed* if every occurrence of every variable in α is bound.

An *interpretation* $\mathbf{I}$ of a first-order language $\mathscr{L}$ is a pair $\langle D, I\rangle$, where D is a nonempty *domain* and I is a function defined on the constant and predicate symbols of $\mathscr{L}$ such that:

- If a is a constant, then $I(a) \in D$.
- If P is an n-ary predicate, distinct from $\mathbf{0}$ and $\mathbf{1}$, then $I(P)$ is a function from D^n to $\{0, 1\}$, $I(\mathbf{0}) = 0$ and $I(\mathbf{1}) = 1$.

A *pre-interpretation* $\mathbf{I}$ is a pair $\langle D, I\rangle$, where D is a non-empty domain and I is a function defined on the constant symbols of $\mathscr{L}$ such that for each constant a, $I(a) \in D$.

An *atomic fact* is an expression of the form $P(\mathbf{d})$, where P is a n-ary predicate and $\mathbf{d}$ is a n-vector of elements in D. We often specify the part of the function I concerning the predicates by enumerating all the atomic facts "true" for I.

An assignment μ for an interpretation $\mathbf{I} = \langle D, I\rangle$ is a function mapping terms (variables and constants) of L to elements of D such that if a is constant then $\mu(a) = I(a)$. The *evaluation* to $\{0, 1\}$ of a formula α under an assignement μ according to the interpretation $\mathbf{I} = \langle D, I\rangle$, denoted by $\mathbf{I}(\alpha[\mu])$ or $\mathbf{I}(\alpha)$ if α is closed, is defined in the usual manner.

Let $\alpha \in \mathscr{L}$ be a closed formula, $\Sigma \subseteq \mathscr{L}$ be a set of formulas and $\mathbf{I} = \langle D, I\rangle$ be an interpretation of $\mathscr{L}$, then $\mathbf{I}$ satisfies α, denoted $\mathbf{I} \models^{\text{fol}} \alpha$, iff $I(\alpha) = 1$; $\mathbf{I}$ is a (*first-order*) *model* of Σ iff $\mathbf{I}$ satisfies each formula in Σ. The set of models of Σ is denoted by $\mathscr{M}^{\text{fol}}(\Sigma)$, Σ is *consistent* if $\mathscr{M}^{\text{fol}}(\Sigma) \neq \varnothing$ and *inconsistent* otherwise, Σ *implies* α, denoted $\Sigma \models^{\text{fol}} \alpha$, iff $\forall \mathbf{I} \in \mathscr{M}^{\text{fol}}(\Sigma)$, $\mathbf{I} \models^{\text{fol}} \alpha$.

In what follows we generally include the binary equality predicate, denoted $\simeq$. As a notational convenience, if $\mathbf{I} = \langle D, I\rangle$ is an interpretation of a language having $\simeq$, we write $d \simeq_{\mathbf{I}} e$, or $d \simeq e$ if $\mathbf{I}$ is understood, to mean $I[\simeq](d, e) = 1$, and $d \not\simeq_{\mathbf{I}} e$ to mean $I[\simeq](d, e) = 0$. There are two well-known methods for "forcing" $\simeq$ to behave like equality. The first is fundamentally semantic: an interpretation $\mathbf{I} = \langle D, I\rangle$ is an $\mathscr{E}$-*interpretation* if

- *Reflexivity*: for each $d \in D$, $d \simeq d$;

- *Symmetry*: for each $d, e \in D$, if $d \simeq e$ then $e \simeq d$;
- *Transitivity*: for each $d, e, f \in D$, if $d \simeq e$ and $e \simeq f$, then $d \simeq f$; and
- *Preservation by substitution*: for each $\mathbf{d} = d_1, ..., d_n$ and $\mathbf{e} = e_1, ..., e_n$ from D and n-ary predicate P of $\mathscr{L}$, if $d_i \simeq e_i$ for $i \in [1, ..., n]$, then $I[P](\mathbf{d}) = I[P](\mathbf{e})$.

The second method is fundamentally syntactic: for a language $\mathscr{L}$, the *equality axioms* for $\mathscr{L}$, denoted $EQ_{\mathscr{L}}$ or EQ if $\mathscr{L}$ is understood, are

- *Reflexivity*: $\forall x(x \simeq x)$;
- *Symmetry*: $\forall x \forall y(x \simeq y \rightarrow y \simeq x)$;
- *Transitivity*: $\forall x \forall y \forall z(x \simeq y \wedge y \simeq z \rightarrow x \simeq z)$; and
- *Preservation by substitution*: $\forall x_1 \cdots \forall x_n \ \forall y_1 \cdots \forall y_n(x_1 \simeq y_1 \wedge \cdots \wedge x_n \simeq y_n \wedge P(x_1 \cdots x_n) \rightarrow P(y_1 \cdots y_n))$, for each n-ary predicate symbol P in $\mathscr{L}$.

A well-known result of first-order logic [ChL] states that for a theory Σ, the family of E-interpretations which satisfy Σ is equal to the family of all interpretations satisfying $(\Sigma \cup \mathrm{EQ})$.

We now briefly review two other types of axioms involving the equality predicate and which are classically used to model logic databases:

If a language $\mathscr{L}$ has constant symbols $\mathscr{C} = \{c_1, ..., c_n\}$, then the Domain Closure axiom (DC) for $\mathscr{L}$ is

$$\forall x(x \simeq c_1) \vee \cdots \vee (x \simeq c_n).$$

The Unique Name axiom (UN) for $\mathscr{L}$ is

$$\wedge \{c_i \not\simeq c_j \mid i \neq j\}.$$

Finally, an *Herbrand interpretation* of $\mathscr{L}$ is an interpretation $\mathbf{I} = \langle \mathscr{C}, I \rangle$ whose domain is the set of constant symbols $\mathscr{C}$ of $\mathscr{L}$, such that $I[c] = c$ for each $c \in \mathscr{C}$. If Σ is a set of sentences over a language L then there exists an obvious correspondence between the set of first order models of $\Sigma \cup \mathrm{DC}$ and the set of Herbrand models of Σ.

4. Minimalism

In this section we introduce and compare two related formalizations of the CWA, the model theoretic approach of[2] minimalism [BS] and a proof theoretic formalization based on the default logic of Reiter [R2]. As suggested in Section 2, both find their basis in first-order logic and provide added conditions which in effect enforce the intitive restriction that an atomic fact is false unless there is an

[2] This differs from the usage of McCarthy in [Mc1], where models are used which "minimalize" only certain literals.

explicit reason to make it true. The primary result of this section, Theorem 4.9, shows that in the context of deductive databases these two approaches are equivalent. We also note a significant difference in the approaches when the assumption of domain closure is relaxed.

In our formal discussion, we focus exclusively on clauses which have at least one positive literal. It is thus convenient to define (positive) clauses by:

DEFINITION. For a language $\mathscr{L}$, a *clause* is a closed formula of the form $\alpha = \forall \mathbf{x}(L_1 \wedge \cdots \wedge L_n \rightarrow L_{n+1} \vee \cdots \vee L_{n+m})$, written $L_1 \cdots L_n \rightarrow L_{n+1} \cdots L_{n+m}$ ($n \geqslant 0$ and $m > 0$), where L_j is a positive literal, $j \in [1, ..., n+m]$. If $m = 1$, α is called a *Horn clause*.

In this section, a database is considered to be a (finite) set of clauses.

Throughout the paper, we do not restrict our attention to Herbrand interpretation as it is usually done in the context of databases. Our aim is to keep the discussion as general as possible. However, whenever necessary, we assume that the domain closure axioms hold.

As suggested in Section 2, in the minialist approach of [BS] the focus is on models of a set Σ of clauses which satisfy as few atomic facts as possible. To accomplish this, a partial ordering on interpretations is established:

DEFINITION. Let $\mathbf{I} = \langle D, I \rangle$ and $\mathbf{I}' = \langle D, I' \rangle$ be two interpretations of $\mathscr{L}$ with the same preinterpretation. Then $\mathbf{I} \leqslant \mathbf{I}'$ if for each n-ary predicate P of $\mathscr{L}$ and each n-vector $\mathbf{d}$ from D, $\mathbf{I}[P](\mathbf{d}) \leqslant \mathbf{I}'[P](\mathbf{d})$.

Speaking roughly, we focus on the models of a set of clauses which are minimal with respect to $\leqslant$, and say that Σ "minimally implies" σ if all minimal models of Σ satisfy σ. As explained in [BS] and illustrated in Example 4.3 below, however, anomalies can arise if all minimal models of Σ are considered. We thus restrict our attention to

DEFINITION. An interpretation $\mathbf{I} = \langle D, I \rangle$ is *discriminant* if for each pair a, b of distinct constants in $\mathscr{L}$, $I[a] \neq I[b]$.

In general, discriminant is not the semantic analog of the unique name assumption, because it may occur in a discriminant interpretation $\mathbf{I} = \langle D, I \rangle$ that for two distinct constants a and b we have $I[a] \neq I[b]$ and $I[a] \simeq I[b]$.

We are now able to define the following key notion.

DEFINITION. Let Σ be a set of clauses. An interpretation $\mathbf{I}$ is a *minimal* model for Σ if it is a discriminant E-interpretation satisfying Σ with the property that: if $\mathbf{I}'$ is a discriminant E-interpretation which satisfies Σ such that $\mathbf{I}' \leqslant \mathbf{I}$, then $\mathbf{I}' = \mathbf{I}$. The family of minimal models of Σ is denoted $\mathscr{M}^{\min}(\Sigma)$.

EXAMPLE 4.1. Let $\mathscr{L}$ have constants a, b; unary predicates P, Q, R; and binary predicate $\simeq$; and let $\Sigma = \{P(a), P(x) \to Q(x) \vee R(x)\}$. Let $\mathbf{I} = (\{d_1, d_2, d_3\}, I)$, where $I[a] = d_1$ and $I[b] = d_2$, and where the true atomic facts are $\{P(d_1), Q(d_1)\}$. Then $\mathbf{I}$ is a minimal model of Σ. Let $\mathbf{I}'$ have the same pre-interpretation as $\mathbf{I}$ and have true atomic facts $\{P(d_1), R(d_1)\}$. Then $\mathbf{I}'$ is another minimal model of Σ. In fact, $\mathbf{I}$ and $\mathbf{I}'$ are the only minimal models of Σ with the same pre-interpretation as $\mathbf{I}$.

If Σ is an arbitrary set of closed formulas, then there may be a model of Σ which cannot be minimized. However, if Σ is a set of clauses then this cannot occur [BS], and speaking intuitively, this implies that for each first-order (discriminant E-interpretation) I, which is a model of Σ, there is a minimal model I' of Σ such that $I' \leqslant I$. In this sense, each model of Σ is "represented" by some minimal model of Σ. This motivates

Definition. Let Σ be a set of clauses and σ a closed formula. Then Σ *minimally entails* σ, denoted $\Sigma \models^{\min} \sigma$, if $\mathbf{I} \models^{\text{fol}} \sigma$ for each $\mathbf{I} \in \mathscr{M}^{\min}(\Sigma)$.

It is straightforward to show that if Σ is a set of clauses, then the notion of minimal implication of a closed formula σ (i.e., $\Sigma \models^{\min} \sigma$) is identical to the more general notion of "subimplication" [BS] of σ by Σ.

Example 4.2. Let Σ be as in Example 4.1. Then it is relatively straightforward to show that $\Sigma \models^{\min} \neg P(b)$; $\Sigma \models^{\min} \forall x(x \simeq a \vee \neg P(x))$; $\Sigma \not\models^{\min} \neg Q(a)$; $\Sigma \not\models^{\min} \neg R(a)$; and $\Sigma \models^{\min} \forall x(\neg Q(x) \vee \neg R(x))$.

The following example illustrates why the definition of minialism uses only discriminant models.

Example 4.3. Again let Σ be as in Example 4.1, and consider the model $\mathbf{I}_1 = \langle D_1, I_1 \rangle$, where $D_1 = \{d_1, d_2\}$, $I_1[a] = d_1$, $I_1[b] = d_1$, and where the true atomic facts are $\{P(d_1), Q(d_1)\}$. Then $\mathbf{I}_1$ is not discriminant, and $\mathbf{I}_1 \models^{\text{fol}} P(b)$ even though $\Sigma \models^{\min} \neg P(b)$. Intuitively, there is no reason to believe that $a \simeq b$ in Σ, but there is no model less than $\mathbf{I}_1$ for which $a \not\simeq b$ holds.

The above example shows that nondiscriminant models of a database Σ can force equality between two constants, even though Σ does not imply it. It is easily verified that if the equality relation $\simeq$ does not occur in Σ, then each minimal model of Σ will necessarily satisfy the unique name assumption. More generally, in a database Σ possibly involving $\simeq$, minimalism can be used to enforce a relaxed version of the unique name assumption, in which constants are assumed to refer to distinct objects in the world, unless there is an explicit reason to identify them.

One motivation for using minialism in defining database semantics comes from the field of logic programming. Specifically, if Σ is a set of Horn clauses, then the semantics associated to Σ in logic programming (see [Ll], [AvE]) is precisely that of minimalism. The definition of GCWA [Mi] also makes use of minimal models.

We now turn to the syntactic formalization of the CWA based on the default logic of Reiter [R2]. As will be seen (Proposition 4.6), in the context of deductive databases and the CWA, this default logic has a relatively simple form. In the development here, this simple form is first defined and illustrated, and then shown to be equivalent to Reiter's definition.

Suppose now that Σ is a set of clauses. Speaking intuitively, formalizing the CWA using default logic involves defining a set of first-order theories Ω which extend Σ.

DEFINITION. A *CWA-default extension* of a theory Σ over $\mathscr{L}$ is a first-order logic theory Ω such that[3]:

$$\Omega = \mathrm{Cl}(\Sigma \cup \{\neg P(\mathbf{a}) \mid P \text{ is a predicate of } \mathscr{L}, \mathbf{a} \text{ is a vector of constants of } \mathscr{L}, \text{ and } P(\mathbf{a}) \notin \Omega\}).$$

The family of *CWA-default models* of Σ, denoted $\mathscr{M}^{\mathrm{def}}(\Sigma)$, is $\cup\{\mathscr{M}^{\mathrm{fol}}(\Omega) \mid \Omega$ is a CWA-default extension of $\Sigma\}$. For a closed formula σ over $\mathscr{L}$, Σ implies σ *under CWA-default*, denoted $\Sigma \models^{\mathrm{def}} \sigma$, if $\mathbf{I} \models^{\mathrm{fol}} \sigma$ for each $\mathbf{I} \in \mathscr{M}^{\mathrm{def}}(\Sigma)$.

Note that the condition on a CWA-default extension Ω involves Ω on both sides of the equation. Intuitively, if Ω is an extension, then Σ and the negative ground literals in Ω are sufficient to imply all of the positive ground literals of Ω (and in fact, all other closed formulas in Ω). Also, it is easily verified that if Ω is a CWA-default extension of Σ, then Ω is inconsistent iff Σ is inconsistent.

Because CWA-default extensions are defined in a purely proof-theoretic manner, another natural (equivalent) way to define CWA-default implication is: Σ implies σ under CWA-default if $\Omega \vdash \sigma$ for each CWA-default extension of Σ. The definitions for CWA-default extensions are illustrated by

EXAMPLE 4.4. Recall the language and set Σ of clauses from Example 4.1. The CWA-default extensions for Σ are the logical closures of

$$\Omega_1 = \{P(a), P(x) \rightarrow Q(x) \vee R(x), Q(a), \neg R(a), \neg P(b), \neg Q(b), \neg R(b)\}$$

$$\Omega_2 = \{P(a), P(x) \rightarrow Q(x) \vee R(x), \neg Q(a), R(a), \neg P(b), \neg Q(b), \neg R(b)\}.$$

On the other hand, the closure of

$$\Sigma_3 = \{P(a), P(x) \rightarrow Q(x) \vee R(x), Q(a), R(a), \neg P(b), \neg Q(b), \neg R(b)\}$$

is not an extension of Σ. This is because neither $Q(a)$ nor $R(a)$ are in $\mathrm{Cl}(\Sigma \cup \{\neg T(\mathbf{c}) \mid T$ is a predicate of $\mathscr{L}$, $\mathbf{c}$ a vector of constants of $\mathscr{L}$ and $T(\mathbf{c}) \notin \Omega_3\})$.

Finally, it is clear that $\Sigma \models^{\mathrm{def}} Q(a) \vee R(a)$, $\Sigma \models^{\mathrm{def}} \neg(Q(a) \wedge R(a))$, etc.

[3] For a first-order theory Ψ, $\mathrm{Cl}(\Psi)$ denotes the logical closure of Ψ, i.e., $\{\psi$ over $\mathscr{L} \mid \Psi \vdash \psi\}$.

We now show the equivalence of CWA-default extensions of CWA-default theories as defined in [R2]. To do this, we first review the relevant definitions from [R2], and state a result from there.

In [R2], a *default* is an expression of the shape $\alpha(\mathbf{x})$: $M\beta_1(\mathbf{x}), \ldots, M\beta_m(\mathbf{x})/\gamma(\mathbf{x})$, where $\alpha(\mathbf{x})$, $\beta_1(\mathbf{x}), \ldots, \beta_m(\mathbf{x})$, and $\gamma(\mathbf{x})$ are first-order formulas, and M is a new symbol. Intuitively, this expression says that if α holds for some constants $\mathbf{a}$, and it is also consistent to assume $\beta_1(\mathbf{a})$ through $\beta_m(\mathbf{a})$, then $\gamma(\mathbf{a})$ should hold. A *default theory* is a pair $\Delta = (\mathrm{DEF}, \Sigma)$, where Σ is a conventional first-order theory, and DEF is a set of defaults. Following [R2], the defaults which express the CWA have a particularly simple form:

DEFINITION. Let $\mathscr{L}$ be a first-order language, and P be a predicate of $\mathscr{L}$. The *CWA default* for P is an expression of the form $M\neg P(\mathbf{x})/\neg P(\mathbf{x})$. The *CWA default theory* for a database Σ over language $\mathscr{L}$ is the default theory $\Delta = (\mathrm{DEF}, \Sigma)$ such that DEF is the set of CWA defaults for all predicates in $\mathscr{L}$.

Given a default theory $\Delta = (\mathrm{DEF}, \Sigma)$, Reiter defines the family of *extensions* of Δ to be a set of first-order theories, each of which includes Σ and incorporates the defaults in a particular way. The family of extensions is defined using a least-fixed point operation, which is applied to a *closed* default theory Δ' associated with Δ. A default theory is closed if all of the formulas occurring in all defaults are closed. In the context of a database Σ, the closed default theory corresponding to the CWA-default theory $\Delta = (\mathrm{DEF}, \Sigma)$ is $\Delta' = (\mathrm{DEF}', \Sigma)$, where

$$\mathrm{DEF}' = \{M\neg P(\mathbf{a})/\neg P(\mathbf{a}) \mid P(\mathbf{a}) \text{ a ground positive atom of } \mathscr{L}\}.$$

The following characterization of extensions due to Reiter will be used.

RESULT 4.5 [R2]. Let Ω be a set of formulas over a language $\mathscr{L}$ and $\Delta = (\mathrm{DEF}, \Sigma)$ be a closed default theory. Let $\Omega_0 = \Sigma$, and for each $i \geqslant 0$, let $\Omega_{i+1} = \mathrm{Cl}(\Omega_i) \cup \Gamma_i$, where $\Gamma_i = \{\gamma \mid \alpha : M\beta_1, \ldots, M\beta_n/\gamma \in \mathrm{DEF},\ \alpha \in \Omega_i, \text{ and } \neg\beta_j \notin \Omega$ for $j \in [1 \cdots n]\}$.

Then Ω is an extension (in the sense of [R2]) of Δ iff $\Omega = \bigcup_{i=0}^{\infty} \Omega_i$.

We now have

PROPOSITION 4.6. *Let Σ be a database over language $\mathscr{L}$, and $\Delta = (\mathrm{DEF}, \Sigma)$ be the associated CWA-default theory. Then Ω is a CWA-default extension for Σ (as defined here) iff Ω is an extension for Δ (as defined in* [R2]).

Proof. Let $\Delta' = (\mathrm{DEF}', \Sigma)$ be the closed default theory corresponding to Δ. Suppose first that Ω is a CWA-default extension of Σ. Let $\Omega_0, \Omega_1, \ldots$ be defined for Δ' as in Result 4.5. Thus, $\Omega_0 = \Sigma$, and

$$\Omega_1 = \mathrm{Cl}(\Sigma) \cup \Gamma_0 = \mathrm{Cl}(\Sigma) \cup \{\neg P(\mathbf{a}) \mid P(\mathbf{a}) \notin \Omega\}.$$

By the definition of CWA-default extension, we know $\Omega = \mathrm{Cl}(\Omega_1)$. On the other hand, note that $\Gamma_i = \Gamma_0$ for each $i \geqslant 0$ because in each default of DEF′ there is no α term. It follows that $\Omega_i = \mathrm{Cl}(\Omega_1)$ for $i > 1$, and so $\mathrm{Cl}(\Omega_1) = \bigcup_{i=0}^{\infty} \Omega_i$. This implies that $\Omega = \bigcup_{i=1}^{\infty} \Omega_i$; and by Result 4.5, Ω is an extension of Δ'.

For the converse, suppose that Ω is an extension of Δ'. Then $\Omega = \bigcup_{i=0}^{\infty} \Omega_i$, where the Ω_i are defined for Δ' according to Result 4.5. As above $\Gamma_i = \Gamma_0$ for each $i \geqslant 0$. It follows that $\Omega_i = \Omega_2 = \mathrm{Cl}(\Omega_1)$ for each $i \geqslant 2$. Since $\Omega_0 = \Sigma \subseteq \Omega_1$, this yields

$$\Omega = \bigcup_{i=0}^{\infty} \Omega_i = \mathrm{Cl}(\Omega_1) = \mathrm{Cl}(\Sigma \cup \Gamma_0).$$

Thus, Ω satisfies the definition of CWA-default extension. ∎

We turn now to the comparison of minimalism and CWA-defaults. Example 4.7 shows that these two formalisms have different impact on domain elements not identified by constants. And Theorem 4.9 shows that under the assumption of domain closure the two formalisms are equivalent.

EXAMPLE 4.7. Recall the theory Σ from Example 4.1 and its two CWA-default extensions Ω_1 and Ω_2 from Example 4.4. It is clear that the models $\mathbf{I}$ and $\mathbf{I}'$ of Example 4.1 are CWA-default models of Σ. (In particular, $\mathbf{I} \models^{\mathrm{fol}} \Omega_1$ and $\mathbf{I}' \models^{\mathrm{fol}} \Omega_2$.)

Note that except for the clause from Σ, neither of these extensions places any constraints on the behavior of domain elements not corresponding to constants. Indeed, consider the model $\mathbf{I}''$ of $\mathcal{M}^{\mathrm{def}}(\Sigma)$ which has the same pre-interpretation as $\mathbf{I}$ and $\mathbf{I}'$, and which has the true facts $\{P(d_1), Q(d_1), Q(d_3), R(d_3)\}$. It now follows that $\Sigma \not\models^{\mathrm{def}} \forall x(\neg Q(x) \vee \neg R(x))$ but $\Sigma \models^{\mathrm{min}} \forall x(\neg Q(x) \vee \neg R(x))$.

As illustrated by this example, CWA-default extensions do not enforce a behavior analogous to minimalism on domain elements not identified by constants. (This is also not surprising as the CWA-default approach is proof theoretic.) It is therefore natural to compare minimalism and CWA-defaults for databases Σ in conjunction with the domain closure axiom. (Note that DC is a clause in our current formalism).

Suppose that Σ is a set of clauses and Ω is a CWA-default extension of $\Sigma \cup \mathrm{DC}$. It is easily verified for each atom $P(\mathbf{a})$ of the underlying language that Ω contains $P(\mathbf{a})$ or $\neg P(\mathbf{a})$. Since Ω also includes DC, Ω is a complete theory.

The inclusion of DC in the databases under consideration obviously means that the predicate $\simeq$ is present. This is not a problem in connection with minimalism, because all minimal models are E-interpretations by definition. In the largely proof-theoretic realm of CWA-defaults, we augment databases by the equality axioms EQ.

To prove the equivalence of minimalism and CWA-defaults under DC we need some notation and a technical lemma.

Notation. For an interpretation $\mathbf{I} = \langle D, I \rangle$ over a language $\mathscr{L}$, $\bar{I}$ denotes the set $\{\neg P(\mathbf{a}) \mid P(\mathbf{a})$ is an atom of $\mathscr{L}$ and $\mathbf{I} \models^{\text{fol}} \neg P(\mathbf{a})\}$.

LEMMA 4.8. *Let $\mathbf{I} = \langle D, I \rangle$ and $\mathbf{J} = \langle D', J \rangle$ be two discriminant E-interpretations such that $\mathbf{I} \models^{\text{fol}} \text{DC}$ and $\mathbf{J} \models^{\text{fol}} \text{DC} \cup \bar{I}$. Then there exists a discriminant E-interpretation $\mathbf{K} = \langle D, K \rangle$ with the same pre-interpretation as $\mathbf{I}$ such that $\mathbf{K} \leqslant \mathbf{I}$ and for each first-order formula σ, $\mathbf{K} \models^{\text{fol}} \sigma$ iff $\mathbf{J} \models^{\text{fol}} \sigma$.*

Proof. Let $c_1, ..., c_n$ be an enumeration of the constant symbols of $\mathscr{L}$, and let $\{D_i \mid i = 1 \cdots n\}$ be a partition of D such that:

- for $i \in [1 \cdots n]$, $I(c_i) \in D_i$, and
- for $i \in [1 \cdots n]$, $d, d' \in D_i \Rightarrow d \simeq_{\mathbf{I}} d'$.

(Note here that this partition is not necessarily unique.) Define the interpretation $\mathbf{K} = \langle D, K \rangle$ such that:

- for each i in $[1 \cdots n]$, $K[c_i] = I[c_i]$, and
- for each l-predicate P and l-vector $d_1 \cdots d_l$ of D, $K[P](d_1 \cdots d_l) = 1$ iff $\mathbf{J} \models^{\text{fol}} P(c_{i_1} \cdots c_{i_l})$, where $d_j \in D_{i_j}$ for $j \in [1 \cdots l]$.

Speaking intuitively, the basic technique to prove that $\mathbf{K}$ has the desired properties involves switching back and forth between the world of constant symbols as interpreted by $\mathbf{J}$, and the world of domain elements in $\mathbf{K}$.

By construction, $\mathbf{K}$ is discriminant. To see that $\mathbf{K}$ is an E-interpretation, we must verify the four properties specified in Section 3, namely: reflexivity, symmetry, transitivity, and preservation by substitution. We consider symmetry here; demonstration of the other properties is analogous. To see that $\simeq_{\mathbf{K}}$ satisfies symmetry, suppose $d, d' \in D$ satisfy $d \simeq_{\mathbf{K}} d'$. By construction of $\mathbf{K}$, this implies $d \in D_i$, $d' \in D_j$ for some i, j in $[1 \cdots n]$ and $\mathbf{J} \models^{\text{fol}} c_i \simeq c_j$. Because $\mathbf{J}$ is an E-interpretation, $\mathbf{J} \models^{\text{fol}} c_j \simeq c_i$ and so $d' \simeq_{\mathbf{K}} d$.

It is clear also that $\mathbf{K} \models^{\text{fol}} \text{DC}$. Because $\mathbf{K} \models^{\text{fol}} \text{DC}$, in order to show that $\mathbf{K} \models^{\text{fol}} \sigma$ iff $\mathbf{J} \models^{\text{fol}} \sigma$ for each first-order formula σ it suffices to show that $\mathbf{K} \models^{\text{fol}} P(\mathbf{a})$ iff $\mathbf{J} \models^{\text{fol}} P(\mathbf{a})$ for each atom $P(\mathbf{a})$. Suppose $\mathbf{K} \models^{\text{fol}} P(c_{i_1}, ..., c_{i_l})$. Then $K[P](K(c_{i_1}), ..., K(c_{i_l})) = 1$. Since $K(c_{i_j}) = I(c_{i_j}) \in D_{i_j}$ for each $j \in [1 \cdots l]$, we have $\mathbf{J} \models^{\text{fol}} P(c_{i_1} \cdots c_{i_l})$. The converse follows from a similar argument.

Finally, we show that $\mathbf{K} \leqslant \mathbf{I}$. By construction of $\mathbf{K}$, it has same pre-interpretation as $\mathbf{I}$. By the previous paragraph and the assumption that $\mathbf{J} \models^{\text{fol}} \bar{I}$, we have $\mathbf{K} \models^{\text{fol}} \bar{I}$. From this it is straightforward to show that $\mathbf{K} \leqslant \mathbf{I}$. ∎

We now have

THEOREM 4.9. *Let Σ be a set of clauses over a language $\mathscr{L}$. Then for each closed formula σ over $\mathscr{L}$, $(\Sigma \cup \text{DC}) \models^{\min} \sigma$ iff $(\Sigma \cup \text{DC} \cup \text{EQ}) \models^{\text{def}} \sigma$.*

Note here that this result has been generalized, for example, in [BF], where a class of default theories larger than the class of CWA-default theories is studied.

Proof. To prove the result, it suffices to show for each discriminant interpretation $\mathbf{I}$ that $\mathbf{I} \in \mathscr{M}^{\min}(\Sigma \cup \mathrm{DC})$ iff $\mathbf{I} \in \mathscr{M}^{\mathrm{def}}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ})$.

To begin, let $\mathbf{I}$ be a minimal model of $\Sigma \cup \mathrm{DC}$ and let $E_{\mathbf{I}} = \{\sigma \mid \sigma \in \mathscr{L}$ and $\mathbf{I} \models^{\mathrm{fol}} \sigma\}$. It is clear that $\forall \sigma \in \mathscr{L}$, $\mathbf{I} \models^{\mathrm{fol}} \sigma$ iff $E_{\mathbf{I}} \vdash \sigma$. We now show that $E_{\mathbf{I}}$ is a CWA-default extension of $\Sigma \cup \mathrm{DC} \cup \mathrm{EQ}$, i.e., that

$$E_{\mathbf{I}} = \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \{\neg P(\mathbf{a}) \mid P(\mathbf{a}) \notin E_{\mathbf{I}}\}).$$

It is easily verified that the right-hand side of this equality is

$$\mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \{\neg P(\mathbf{a}) \mid \mathbf{I} \models^{\mathrm{fol}} \neg P(\mathbf{a})\}) = \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I}).$$

By construction, we have: $\mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I}) \subseteq E_{\mathbf{I}}$. It thus remains to show that: $E_{\mathbf{I}} \subseteq \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I})$.

To obtain this final inclusion, it suffices to show both of

$$\{P(\mathbf{a}) \mid P(\mathbf{a}) \in E_{\mathbf{I}}\} \subseteq \{P(\mathbf{a}) \mid P(\mathbf{a}) \in \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I})\} \tag{1}$$

and

$$\{\neg P(\mathbf{a}) \mid \neg P(\mathbf{a}) \in E_{\mathbf{I}}\} \subseteq \{\neg P(\mathbf{a}) \mid \neg P(\mathbf{a}) \in \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I})\}. \tag{2}$$

Now, (2) follows from the definition of $E_{\mathbf{I}}$. To see (1), suppose that $P(\mathbf{a}) \in E_{\mathbf{I}}$ and $P(\mathbf{a}) \notin \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I})$. Then $\mathbf{I} \models^{\mathrm{fol}} P(\mathbf{a})$, and there exists a discriminant E-interpretation $\mathbf{J}$ such that $\mathbf{J} \models^{\mathrm{fol}} \Sigma \cup \mathrm{DC} \cup \bar{I} \cup \{\neg P(\mathbf{a})\}$. By Lemma 4.8, there exist a discriminant E-interpretation $\mathbf{K}$ such that $\mathbf{K} \leqslant \mathbf{I}$ and $\mathbf{K} \models^{\mathrm{fol}} \Sigma \cup \mathrm{DC} \cup \bar{I} \cup \{\neg P(\mathbf{a})\}$. Because $\mathbf{K} \models^{\mathrm{fol}} \neg P(\mathbf{a})$ and $\mathbf{I} \models^{\mathrm{fol}} P(\mathbf{a})$, we see that $\mathbf{K} \neq \mathbf{I}$, which contradicts the minimality of $\mathbf{I}$. Thus (1) holds, and more generally, $E_{\mathbf{I}}$ is a CWA-default extension of $\Sigma \cup \mathrm{DC} \cup \mathrm{EQ}$. It follows that $\mathbf{I} \in \mathscr{M}^{\mathrm{def}}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ})$.

For the converse, suppose now that Ω is a CWA-default extension of $\Sigma \cup \mathrm{DC} \cup \mathrm{EQ}$, and that $\mathbf{I} = \langle D, I \rangle$ is a discriminant model in $\mathscr{M}^{\mathrm{fol}}(\Omega)$. Clearly, $\mathbf{I} \models^{\mathrm{fol}} \Sigma \cup \mathrm{DC}$. To show that $\mathbf{I} \in \mathscr{M}^{\min}(\Sigma \cup \mathrm{DC})$, we suppose that there exists $\mathbf{J}$ in $\mathscr{M}^{\min}(\Sigma \cup \mathrm{DC})$ such that $\mathbf{J} \leqslant \mathbf{I}$, and prove that $\mathbf{J} = \mathbf{I}$.

Because Ω is a CWA-default extension of $\Sigma \cup \mathrm{DC} \cup \mathrm{EQ}$, Ω is complete. Thus, $\Omega = E_{\mathbf{I}}$, where $E_{\mathbf{I}} = \{\sigma \mid \sigma \in L$, and $\mathbf{I} \models^{\mathrm{fol}} \sigma\}$. It follows that

$$\begin{aligned} E_{\mathbf{I}} &= \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \{\neg P(\mathbf{a}) \mid P(\mathbf{a}) \notin E_{\mathbf{I}}\}) \\ &= \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \{\neg P(\mathbf{a}) \mid \mathbf{I} \models^{\mathrm{fol}} \neg P(\mathbf{a})\}) \\ &= \mathrm{Cl}(\Sigma \cup \mathrm{DC} \cup \mathrm{EQ} \cup \bar{I}). \end{aligned}$$

Because $\mathbf{J} \leqslant \mathbf{I}$, $\mathbf{J} \models^{\mathrm{fol}} \bar{I}$. Also, $\mathbf{J} \models^{\mathrm{fol}} \Sigma \cup \mathrm{DC} \cup \mathrm{EQ}$. Thus, $\mathbf{J} \models^{\mathrm{fol}} E_{\mathbf{I}}$. Since $\mathbf{J} \leqslant \mathbf{I}$, $\mathbf{J}$ has the same preinterpretation as $\mathbf{I}$. This implies that $\mathbf{J} = \mathbf{I}$. Thus, $\mathbf{I} \in \mathscr{M}^{\min}(\Sigma \cup \mathrm{DC})$ as desired. ∎

5. POSITIVISM

In this section, we introduce the notion of positivism, a new semantic formalization of the CWA. Positivism combines the semantics of minimalism and *justification*, a model-theoretic analog, and generalization of the negation as failure (NAF) principle of Clark [Cl]. As noted in the Introduction the notion of justified model is equivalent to the notion of supported model, independently and simultaneously introduced in [ABW]. The first part of the section introduces justification, and studies it in isolation. A main result here (Theorem 5.4) states that justification accurately expresses the semantics of NAF as characterized by Clark's "completion." After that, positivism is introduced and studied using several examples and results.

Following the spirit of Clark [Cl] and logic programming [CIT] in general, we begin by slightly modifying the type of formulas that are used to describe a database.

DEFINITION. A *positivistic clause* (P-clause) of a language $\mathscr{L}$ is a closed formula of $\mathscr{L}$ of the form $\alpha = \forall \mathbf{x}(L_1 \wedge \cdots \wedge L_n \rightarrow L_{n+1} \vee \cdots \vee L_{n+m})$, written $L_1 \cdots L_n \rightarrow L_{n+1} \cdots L_{n+m}$, where $n \geqslant 0$; $m > 0$; for $i = 1 \cdots n$, L_i is a literal, called *premise* of α; and for $1 = 1 \cdots m$, L_{n+i} is a positive literal, called *potential consequence* of α. If $m = 1$, then α is called a *positivistic Horn clause* (PH-clause).

Intuitively, P-clauses are introduced to enable us to distinguish rules with negative premises, e.g., $\neg Q \rightarrow P$, from rules with disjunctive consequents, e.g., $Q \vee P$. The first formula is interpreted as "if nothing can make you believe Q is true, then P is true," whereas the second one asserts "either believe Q or P to be true." This intuition is captured in a largely model-theoretic manner by:

DEFINITION. Let Σ be a set of P-clauses and $\mathbf{I} = \langle D, I \rangle$ be an interpretation:

- A positive atomic fact $P(\mathbf{d})$ such that $I[P](\mathbf{d}) = 1$ is *justified* by Σ in $\mathbf{I}$ if there exists a P-clause $\alpha = (L_1 \cdots L_n \rightarrow L_{n+1} \cdots L_{n+m})$ in Σ such that for some j in $[1 \cdots m]$, $L_{n+j} = P(\mathbf{t})$, and there exists an assignment μ with $\mu(\mathbf{t}) = \mathbf{d}$ and $\mathbf{I}(L_i[\mu]) = 1$ for $i \in [1 \cdots n]$.
- The interpretation $\mathbf{I}$ is a *justified* model of Σ if:
  - $\mathbf{I}$ is a discriminant first order model of $\Sigma \cup \mathrm{EQ}$, and
  - each atomic fact true in $\mathbf{I}$ is justified by $\Sigma \cup \mathrm{EQ}$ in $\mathbf{I}$.

The set of justified models of Σ is denoted by $\mathscr{M}^{\text{jus}}(\Sigma)$. Finally, a closed formula σ is *implied under justification* by Σ, denoted $\Sigma \models^{\text{jus}} \sigma$, if $\mathbf{I} \models^{\text{fol}} \sigma$ for each $\mathbf{I} \in \mathscr{M}^{\text{jus}}(\Sigma)$.

The following simple example shows that the semantics of justification is different than that of minimalism. (See Example 2.3 for a more involved example of justification.)

EXAMPLE 5.1. Let P and Q be two predicates of rank 0, and $\Sigma = \{\neg P \rightarrow Q\}$. Then $\mathbf{I}_1 = \{Q\}$ is a justified model of Σ. $\mathbf{I}_1$ is also a minimal model of Σ. On the other hand, $\mathbf{I}_2 = \{P\}$ is a minimal model of Σ which is not a justified model of Σ. Thus, $\Sigma \models^{\text{jus}} Q$ and $\Sigma \models^{\text{jus}} \neg P$, but $\Sigma \not\models^{\min} Q$ and $\Sigma \not\models^{\min} \neg P$.

We now present results which show that in the case where Σ is a set of PH-clauses, the semantics of justification as defined here correspond closely to the semantics of NAF. To present these results, we include a brief review of the NAF algorithm, the notion of global answer, and the Clark "completion" of a set of PH-clauses.

We begin by recalling some results about the NAF inference rule. In [Cl], Clark considers databases as sets of PH-clauses and proposes a query evaluation algorithm which is essentially a linear resolution procedure except for the use of the NAF inference rule. Intuitively, Clark's inference rule states that any ground negative literal is evaluated by failure of all possible proofs of the corresponding positive ground literal. As indicated in Remark 5.5 below there are some subtleties concerning the use of equality in connection with NAF; and so for the current discussion we assume that the equality predicate $\simeq$ does not occur in the sets Σ of PH-clauses under consideration.

In the following, we assume the reader is familiar with the terminology of resolution [ChL]. A *valid query* for Σ is a conjunction $L_1 \wedge \cdots \wedge L_n$, where each L_i is a literal from $\mathscr{L}$ not involving $\simeq$. Given a query Q, the evaluation procedure of Clark constructs an evaluation tree for Q, using a non-deterministic sub-procedure whose purpose is to constructively explore one path of the evaluation tree. To make the discussion clear, this non-deterministic procedure is outlined here by the Resolution by NAF algorithm [CL]:

ALGORITHM 5.2 (Resolution by Negation as Failure).

- **input**: a set of PH-clauses Σ, and a valid query $Q = Q_1 \wedge \cdots \wedge Q_n$, Q_i is a literal, for $i \in [1 \cdots n]$.
- **output**: FAILS, SUCCEEDS.

If $n = 0$ **then** return SUCCEEDS **else**
  Select[4] $i \in [1 \cdots n]$ such that Q_i is a positive literal or Q_i is ground negative literal.
  If Q_i is a positive literal **then**
    Non-deterministically select a PH-clause α in Σ. Let $\alpha = L_1 \cdots L_m \rightarrow L_{m+1}$
    If there exists a most general unifier θ for Q_i and L_{m+1} **then**
      enter the Algorithm with Σ, $(Q_1 \wedge \cdots \wedge Q_{i-1} \wedge L_1 \wedge \cdots \wedge L_m \wedge Q_{i+1} \wedge \cdots \wedge Q_n)\theta$.
    else return FAILS.

[4] A rule for selecting literals in PH-clauses is assumed to be fixed.

If $Q_i = \neg R(\mathbf{a})$ is a negative ground literal **then**
  If every execution of Algorithm FAILS with the input Σ, $R(\mathbf{a})$ **then**
    enter Algorithm with Σ, $(Q_1 \wedge \cdots \wedge Q_{i-1} \wedge Q_{i+1} \wedge \cdots \wedge Q_n)$.
  else return FAILS.

The resolution by NAF algorithm is initially called with the identity substitution θ_0 for $\mathbf{x}$, and refines this substitution during its (recursive) execution. If the resolution by NAF algorithm succeeds, it provides a substitution θ for the free variables $\mathbf{x}$ in Q. The notion of a *global answer* combines all the individual answers obtained by different executions of the algorithm and is defined as follows. (This definition focusses on queries having at least one free variable, but is easily adjusted when Q is a ground query.)

DEFINITION. Let Σ be a set of PH-clauses and Q a valid query, where $\mathbf{x}$ is the sequence of free variables in Q and $|\mathbf{x}| > 0$. Let $\theta_1 \cdots \theta_k$ be all the substitutions obtained by the complete evaluation of Q.

- If $k = 0$, then the *global answer* to Q is the formula $\forall \mathbf{x}(Q \leftrightarrow 0)$, and
- If $k \neq 0$, then the *global answer* to Q is the formula $\forall \mathbf{x}(Q \leftrightarrow \alpha_1 \vee \cdots \vee \alpha_k)$, where for $i \in [1 \cdots k]$, $\alpha_i = ((x_1 \simeq a_i^1) \wedge \cdots \wedge (x_n \simeq a_i^n))$ with $\theta_i = \langle x_1 \cdots x_n / a_i^1 \cdots a_i^n \rangle$.

Note that in the general case, it is not guaranteed that the resolution by NAF algorithm terminates. (In particular, the algorithm may go into an infinite loop on some branch, or flounder, i.e., stop before completion because it is faced with a conjunction of non-ground literals.) In [Cl], a sufficient condition on the set of PH-clauses specifying the database is given that ensures termination of the NAF algorithm for any valid query. In order to validate the NAF algorithm, Clark uses the notion of completed database associated with a set Σ of PH-clauses. We next give the definition of this completion.

Let $\alpha = L_1 \cdots L_n \rightarrow P(t_1 \cdots t_m)$ be a PH-clause in Σ. Let $x_1, ..., x_n$ be variables not occurring in Σ, and let rew(α) be the formula: $[\exists y_1 \cdots \exists y_l (x_1 \simeq t_1) \wedge \cdots \wedge (x_m \simeq t_m) \wedge L_1 \wedge \cdots \wedge L_n] \rightarrow P(x_1 \cdots x_m)$, where $y_1 \cdots y_l$ are the variables occurring in α.

Given a set Σ of PH-clauses over $\mathscr{L}$, and a predicate P in $\mathscr{L}$ other than $\simeq$, let $\{\gamma_1 \rightarrow P(\mathbf{x}), ..., \gamma_k \rightarrow P(\mathbf{x})\} = \{\text{rew}(\alpha) \mid \alpha \in \Sigma$ and the consequent predicate of α is $P\}$. Then the *completed definition* of P according to Σ, denoted cp(Σ, P), is:

- if $k = 0$, $\forall \mathbf{x}(P(\mathbf{x}) \leftrightarrow \mathbf{0})$, and
- if $k > 0$, $\forall \mathbf{x}(P(\mathbf{x}) \leftrightarrow \gamma_1 \vee \cdots \vee \gamma_k)$.

We now are able to define the notion of a completed database. This definition uses the preceding completion definition of the predicate and also the equality and unique name axioms.

DEFINITION. Let Σ be a set of PH-clauses not involving the equality $\simeq$, over the first-order language $\mathscr{L}$. The completion of Σ is the first-order theory $\mathrm{CP}(\Sigma) = \{\mathrm{cp}(\Sigma, P) \mid P \text{ is a predicate in } \mathscr{L} \text{ other than } \simeq\} \cup \mathrm{EQ} \cup \mathrm{UN}_{\mathscr{L}}$.

The validation of the resolution by NAF algorithm is now given by

RESULT 5.3 [Cl]. Let Σ be a set of PH-clauses such that $\mathrm{CP}(\Sigma)$ is consistent, and Q be a valid query. If each path of the evaluation of Q on Σ using Algorithm 5.2 terminates then $\mathrm{CP}(\Sigma) \vdash \alpha$, where α is the global answer to Q.

This concludes our review of NAF and Clark's completion. The following result shows the close relationship between the semantics of Clark's completion and justification. (See also Theorem 5.10, which shows that positivism generalizes NAF.)

THEOREM 5.4. *Let Σ be a set of PH-clauses not involving the equality predicate $\simeq$, over the language $\mathscr{L}$. Then, for each interpretation* $\mathbf{I}$, $\mathbf{I} \in \mathscr{M}^{\mathrm{fol}}(\mathrm{CP}(\Sigma))$ *iff* $\mathbf{I} \in \mathscr{M}^{\mathrm{jus}}(\Sigma)$ *and* $\mathbf{I} \models^{\mathrm{fol}} \mathrm{UN}$.

Proof. ■ Suppose that $\mathbf{I} \in \mathscr{M}^{\mathrm{jus}}(\Sigma)$ and $\mathbf{I} \models^{\mathrm{fol}} \mathrm{UN}$. We show that $\mathbf{I} \models^{\mathrm{fol}} \mathrm{CP}(\Sigma)$. Because $\mathbf{I}$ is a justified model it satisfies EQ, and by assumption it satisfies UN.

Now consider a formula α in $\mathrm{CP}(\Sigma)$ which is a completed definition of some predicate P (other than $\simeq$). Suppose that $\alpha = \forall \mathbf{x}(P(\mathbf{x}) \leftrightarrow \gamma_1 \wedge \cdots \vee \gamma_k)$. In order to show that $I \models^{\mathrm{fol}} \alpha$, it suffices to show that for each assignment μ for $\mathbf{x}$, $\mathbf{I}((P(\mathbf{x}) \leftrightarrow \gamma_1 \vee \cdots \vee \gamma_k)[\mu]) = 1$. As $\mathbf{I} \models^{\mathrm{fol}} \Sigma$, it is clear that $\mathbf{I}((\gamma_1 \vee \cdots \vee \gamma_k \rightarrow P(\mathbf{x}))[\mu]) = 1$. It remains to show that $\mathbf{I}((P(\mathbf{x}) \rightarrow \gamma_1 \vee \cdots \vee \gamma_k)[\mu]) = 1$.

Suppose that $\mathbf{I}(P(\mathbf{x})[\mu]) = 1$. As $\mathbf{I} \models^{\mathrm{jus}} \Sigma$, the atomic fact $P(\mu(\mathbf{x}))$ is justified by Σ in $\mathbf{I}$. So there exists a clause $L_1 \cdots L_m \rightarrow P(\mathbf{t})$ in Σ and an assignment ν for $\mathbf{x}$ such that $\nu(\mathbf{t}) = \mu(\mathbf{x})$ and $\mathbf{I}((L_1 \wedge \cdots \wedge L_m)[\nu]) = 1$. Note that for some j in $[1 \cdots k]$, $\gamma_j = \exists \mathbf{y}(\mathbf{x} \simeq \mathbf{t} \wedge L_1 \wedge \cdots \wedge L_m)$. It follows that $\mathbf{I}((\gamma_j)[\mu]) = 1$, and so $\mathbf{I}((P(\mathbf{x}) \rightarrow \gamma_1 \vee \cdots \vee \gamma_k)[\mu]) = 1$ and $\mathbf{I} \models^{\mathrm{fol}} \alpha$. A similar argument can be used to show that $\mathbf{I} \models^{\mathrm{fol}} \alpha$, where $\alpha = \forall \mathbf{x} P(\mathbf{x}) \leftrightarrow 0$.

■ For the converse, suppose that $\mathbf{I} \models^{\mathrm{fol}} \mathrm{CP}(\Sigma)$. Clearly, $\mathbf{I} \models^{\mathrm{fol}} \Sigma \cup \mathrm{EQ}$. Also, because $\mathbf{I} \models^{\mathrm{fol}} \mathrm{UN}$ it easily follows that $\mathbf{I}$ is discriminant. It now remains to show that each positive fact true in $\mathbf{I}$ is justified by $\Sigma \cup \mathrm{EQ}$ in $\mathbf{I}$. Suppose $P(\mathbf{d})$ is a positive fact true in $\mathbf{I}$. If P is $\simeq$ then it is easily verified that $P(\mathbf{d})$ is justified by some member of EQ because $I[\simeq]$ is an equivalence relation. Now suppose that P is not $\simeq$. Let $\forall \mathbf{x}(P(\mathbf{x}) \leftrightarrow \gamma_1 \vee \cdots \vee \gamma_k)$ be the associated completed definition of P (clearly $k > 0$) and let μ be an assignment such that $\mu(\mathbf{x}) = \mathbf{d}$. Because $\mathbf{I} \models^{\mathrm{fol}} \mathrm{CP}(\Sigma)$, $\mathbf{I}((P(\mathbf{x}) \rightarrow \gamma_1 \vee \cdots \vee \gamma_k)[\mu]) = 1$. Thus for some j in $[1 \cdots k]$, $\mathbf{I}((\gamma_j)[\mu]) = 1$. Let $\gamma_j = \exists \mathbf{y}(\mathbf{x} \simeq \mathbf{t} \wedge L_1 \wedge \cdots \wedge L_m)$. There exists an extension ν of μ such that $\mathbf{I}((\mathbf{x} \simeq \mathbf{t} \wedge L_1 \wedge \cdots \wedge L_m)[\nu]) = 1$. In particular then, $\mathbf{I}((L_i)[\nu]) = 1$ for each i in $[1 \cdots m]$; this justifies $P(\mathbf{d})$. ▌

REMARK 5.5. In this remark we explore reasons why $\simeq$ is not permitted in databases interpreted using NAF, and how $\simeq$ can be included in the context of Clark's completion. First, we note that the NAF algorithm as presented above does not give the appropriate semantics in cases where the query involves $\simeq$.

For example, consider the empty database Σ and the query $Q = x \simeq y$. Processing Q on Σ using Algorithm 5.2, we obtain the global answer $Q \leftrightarrow \mathbf{0}$. Obviously, the expected answer is given by the substitution $\langle x, y/x, x \rangle$.

A simple attempt to remedy this problem would be to include EQ with a database Σ when using the NAF algorithm. However, the structure of the NAF algorithm never halts when using $\Sigma \cup \mathrm{EQ}$.

The above observations provide motivation for prohibiting $\simeq$ both in Σ and queries when using NAF. Note that under these restrictions the intuition behind NAF "implies" UN, because there is no reason to identify any distinct pair of constants.

In order to generalize Theorem 5.4 to databases involving $\simeq$, the definition of the completed database is modified as

$$\mathrm{CP}_{\simeq}(\Sigma) = \{\mathrm{cp}(\Sigma \cup \mathrm{EQ}, P) \mid P \text{ is a predicate in } \mathscr{L}\} \cup \mathrm{EQ}.$$

Note that $\mathrm{cp}(\Sigma \cup \mathrm{EQ}, \simeq)$, the completed definition of $\simeq$ is included here, and that the UN is omitted. Using an argument similar to that of Theorem 5.4 it is easily seen that: Let Σ be a set of PH-clauses. Then $\mathbf{I} \in \mathscr{M}^{\mathrm{jus}}(\Sigma)$ iff $\mathbf{I}$ is discriminant and $\mathbf{I} \in \mathscr{M}^{\mathrm{fol}}(\mathrm{CP}_{\simeq}(\Sigma))$.

As indicated in Section 2, the notion of justification alone is not sufficient to capture the CWA when disjunction is allowed in the consequents of the clauses. Moreover, even when restricting databases to be Horn databases (no negative premises or disjunctive consequents), justified models do not accurately capture the CWA. We briefly illustrate this.

EXAMPLE 5.6. (a) Let $\Sigma_1 = \{P \to Q,\ Q \to P\}$. Then $\mathscr{M}^{\mathrm{jus}}(\Sigma_1) = \mathscr{M}^{\mathrm{fol}}(\mathrm{CP}(\Sigma_1))$ has two models, namely $\varnothing$ and $\{P, Q\}$. Although the NAF algorithm does not halt on Σ_1, only the first model satisfies the least-fixed point semantics usually associated with logic programs [L1].

(b) Let $\Sigma_2 = \{P, P \to Q \vee R\}$. There are three justified models of Σ_2, namely $\{P, Q\}$, $\{P, R\}$, and $\{P, Q, R\}$. However, only the first two of these are minimal models.

(c) The preceding example suggests that the semantics of $\vee$ under the CWA should be that of "exclusive-or" as found in first-order logic. The following example indicates why this is not always correct. Let $\Sigma_3 = \{P, P \to Q \vee R, P \to R \vee S, P \to Q \vee S\}$. There are three minimal models of Σ_3: $\{P, Q, R\}$, $\{P, Q, S\}$, $\{P, R, S\}$. None of these satisfies the sentences obtained by replacing "or" by "exclusive-or" in Σ_3. Note also that there are four justified models of Σ_3: the three minimal models, and also $\{P, Q, R, S\}$.

Motivated by the above examples, the notion of a positivstic model combines the notion of a minimal model and a justified model. Other, recently introduced approaches which combine minimalism and justification are briefly discussed in Section 7.

DEFINITION. The set of *positivistic models* of Σ is the set $\mathcal{M}^{\text{pos}}(\Sigma) = \mathcal{M}^{\text{jus}}(\Sigma) \cap \mathcal{M}^{\text{min}}(\Sigma)$. Also, Σ *positivistically implies* α, denoted $\Sigma \models^{\text{pos}} \alpha$, iff $\forall \mathbf{I} \in \mathcal{M}^{\text{pos}}(\Sigma)$, $\mathbf{I} \models^{\text{fol}} \alpha$.

This definition is illustrated by Example 2.3. In order to highlight some subtleties in the way that positivism combines minimalism and justification, we present an example and a provocative result. The example shows that the family of positivistic models of a set of P-clauses, i.e., the set of models which are both minimal and justified, is not necessarily the set of minimal models among $\mathcal{M}^{\text{jus}}(\Sigma)$.

EXAMPLE 5.7. (a) Let $\Sigma_1 = \{P \vee Q,\ P \to R,\ Q \to S,\ \neg R \to T,\ \neg S \to T\}$. The positivistic models of Σ_1 are $\{P, R, T\}$ and $\{Q, S, T\}$. Note that $\{P, Q, R, S\}$ is minimal among justified models of Σ_1; it is not positivistic because it is not minimal among the first-order models of Σ_1 ($\{P, R, S\}$ also satisfies Σ_1).

(b) Consider the set $\Sigma_2 = \{\neg P \to Q,\ \neg Q \to P,\ P \to P,\ Q \to Q,\ P \to R,\ Q \to S,\ \neg R \to T,\ \neg S \to T\}$ of PH-clauses. As in part (a), there are models which are minimal among $\mathcal{M}^{\text{jus}}(\Sigma_2)$ which are not positivistic models of Σ_2.

As illustrated by the above examples, in the general case minimal among justified is not the same as positivistic, i.e., minimal and justified. A natural question is to find classes of databases in which these two notions coincide. One possible class is the family of databases Σ for which the completion $\text{CP}(\Sigma)$ is a complete theory in first-order logic. Speaking intuitively, this class seems like a natural candidate, because for each Σ in this class there is exactly one justified model of Σ which satisfies UN (by Theorem 5.4). The following result shows that even in this class, the two notions are not equivalent.

PROPOSITION 5.8. *There exists a set Σ of PH-clauses such that* $\text{CP}(\Sigma)$ *is complete, where the justified model of Σ satisfying UN is not a minimal model of Σ.*

Proof. Let $\Sigma = \{P \to P,\ Q \to P,\ \neg R \to P,\ P \wedge \neg Q \to R\}$. We first show that the completion $\text{CP}(\Sigma)$ is complete in first-order logic. From the definition,

$$\text{CP}(\Sigma) = \{P \leftrightarrow P \vee Q \vee \neg R,\ Q \leftrightarrow 0,\ R \leftrightarrow P \wedge \neg Q\}.$$

It is easily verified that $\text{CP}(\Sigma) \models^{\text{fol}} \neg Q$. Since $P \wedge \neg Q \leftrightarrow R \in \text{CP}(\Sigma)$ this implies $\text{CP}(\Sigma) \models^{\text{fol}} P \leftrightarrow R$. Also, $P \leftrightarrow P \vee Q \vee \neg R$ yields $\text{CP}(\Sigma) \models^{\text{fol}} \neg R \to P$. It now follows that $\text{CP}(\Sigma) \models^{\text{fol}} P$, and $\text{CP}(\Sigma) \models^{\text{fol}} R$. $\text{CP}(\Sigma)$ is therefore complete, and by Theorem 5.4, here is exactly one justified model of Σ.

From the preceding, it is clear that the only justified model of Σ is $\{P, R\}$, and so this model is minimal among justified models. On the other hand, $\{R\}$ is a model of Σ which is strictly less than $\{P, R\}$ under $\leqslant$; thus $\{P, R\}$ is not a positivistic model of Σ. ∎

We conclude this section with two results indicating that positivism generalizes previously proposed formalizations of the CWA. The first result focusses on sets Σ in which there are no negative premises.

PROPOSITION 5.9. *Let Σ be a set of clauses (we insist here on the fact that no negative literal occurs in the premises of the clauses of Σ). Then $\mathcal{M}^{\min}(\Sigma) = \mathcal{M}^{\text{pos}}(\Sigma)$.*

Proof. Since $\mathcal{M}^{\text{pos}}(\Sigma) = \mathcal{M}^{\min}(\Sigma) \cap \mathcal{M}^{\text{jus}}(\Sigma)$, in order to show that $\mathcal{M}^{\min}(\Sigma) = \mathcal{M}^{\text{pos}}(\Sigma)$, it suffices to show that $\mathcal{M}^{\min}(\Sigma) \subseteq \mathcal{M}^{\text{jus}}(\Sigma)$. Let $\mathbf{I} = \langle D, I \rangle \in \mathcal{M}^{\min}(\Sigma)$ and let $P(\mathbf{d})$ be a fact true in $\mathbf{I}$. Suppose that $P(\mathbf{d})$ is not justified by Σ in $\mathbf{I}$. We will obtain a contradiction by exhibiting a model $\mathbf{I}' \leqslant \mathbf{I}$ such that $\mathbf{I}' \neq \mathbf{I}$ and $\mathbf{I}' \models^{\text{fol}} \Sigma$. Specifically, let the interpretation $\mathbf{I}' = \langle D, I' \rangle$ be the same as $\mathbf{I}$, except that $I'[P](\mathbf{d}) = 0$. To see that $\mathbf{I}' \models^{\text{fol}} \Sigma$, let $\alpha = L_1 \cdots L_n \rightarrow L_{n+1} \cdots L_{n+m}$ be a clause in Σ, and μ an assignment for the variables occurring in α.

- If $L_{n+i}[\mu] \neq P(\mathbf{d})$ for each i, $1 \leqslant i \leqslant m$, then obviously $\mathbf{I}'(\alpha[\mu]) = 1$.
- Otherwise, since $P(\mathbf{d})$ is not justified by Σ in $\mathbf{I}$, $\mathbf{I}(L_1 \wedge \cdots \wedge L_n[\mu]) = 0$ and so $\mathbf{I}'(L_1 \wedge \cdots \wedge L_n[\mu]) = 0$. Thus, $\mathbf{I}'(\alpha[\mu]) = 1$.

This demonstrates that $\mathbf{I}' \models^{\text{fol}} \Sigma$, and the contradiction is established. ∎

We conclude this section by showing that positivism generalizes the NAF principle.

THEOREM 5.10. *Let Σ be a set of PH-clauses in which $\simeq$ does not occur, such that $\text{CP}(\Sigma)$ is consistent; let Q be a valid query; and suppose that each path of the evaluation of Q on Σ using Algorithm 5.2 terminates. Then $\Sigma \models^{\text{pos}} \alpha$, where α is the global answer of the query Q.*

Proof. Let Σ, Q, and α be as in the statement of the theorem. Suppose that $\mathbf{I} \in \mathcal{M}^{\text{pos}}(\Sigma)$. Then $\mathbf{I} \models^{\text{fol}}$ UN because $\mathbf{I} \in \mathcal{M}^{\min}(\Sigma)$ and Σ does not involve $\simeq$. Also, $\mathbf{I} \in \mathcal{M}^{\text{jus}}(\Sigma)$ and thus by Theorem 5.4, $\mathbf{I} \in \mathcal{M}^{\text{fol}}(\text{CP}(\Sigma))$. From this, Result 5.3 implies that $\mathbf{I} \models^{\text{fol}} \alpha$. This holds for each positivistic model $\mathbf{I}$ of Σ and so $\Sigma \models^{\text{pos}} \alpha$ as desired. ∎

6. EXPRESSIVE POWER AND NON-MONOTONICITY

It is clear that the semantics of both minimalism and positivism are non-monotonic. We begin the section by showing that when insertions are ignored, positivism can simulate minimalism. Also, still ignoring insertions, minimalism can simulate positivism if domain closure is assumed, but not in the general case. We then proceed to a comparative study of the non-monotonicity of these two alternative formalizations of the CWA. In particular, we show that in the presence of simple insertions, minimalism cannot simulate positivism, even if domain closure is assumed.

For the sake of our further discussion, we introduce:

DEFINITION. A *minimalist database* is a pair $(\mathscr{L}, \Sigma)$, where Σ is set of clauses over $\mathscr{L}$; and a *positivist database* is a pair $(\mathscr{L}, \Sigma)$, where Σ is a set of P-clauses over $\mathscr{L}$.

When the first-order language used to specify a database is understood, the database is specified using only a set of clauses or respectively P-clauses.

We start our comparison of the minimalism and positivism approaches of the CWA by showing that two databases equivalent under first-order logic are still equivalent under minimal interpretation, but not necessarily under positivistic interpretation.

REMARK 6.1. Let $\mathscr{L}$ be a first-order language. Then:

■ For each pair Σ, Σ' of minimalist databases, $\mathscr{M}^{\mathrm{fol}}(\Sigma) = \mathscr{M}^{\mathrm{fol}}(\Sigma')$ implies $\mathscr{M}^{\min}(\Sigma) = \mathscr{M}^{\min}(\Sigma')$. (By definition of minimalism.)

■ There are two positivist databases Σ, Σ' such that $\mathscr{M}^{\mathrm{fol}}(\Sigma) = \mathscr{M}^{\mathrm{fol}}(\Sigma')$ and $\mathscr{M}^{\mathrm{pos}}(\Sigma) \neq \mathscr{M}^{\mathrm{pos}}(\Sigma')$. (Consider the databases $\Sigma_1 = \{\neg P \rightarrow Q\}$ and $\Sigma_2 = \{P \vee Q\}$.)

Note that each minimalist database can be viewed as a positivist database while there are positivist databases which are not minimalist databases. Furthermore, the next result applies Proposition 5.9 to show that each minimalist database can be simulated by a positivist database.

PROPOSITION 6.2. *Let $\mathscr{L}$ be a first-order language. For each minimalist database Σ, $\mathscr{M}^{\mathrm{pos}}(\Sigma) = \mathscr{M}^{\min}(\Sigma)$ and so for each formula σ in $\mathscr{L}$, $\Sigma \models^{\mathrm{pos}} \sigma$ iff $\Sigma \models^{\min} \sigma$.*

We now present three results which explore simulation of positivism by minimalism. The first of these shows that if domain closure is assumed, then minimalism can simulate positivism.

THEOREM 6.3. *Let $\mathscr{L}$ be a first-order language. For each positivist database Σ there is a minimalist database Σ' such that $\mathscr{M}^{\min}(\Sigma' \cup \mathrm{DC}) = \mathscr{M}^{\mathrm{pos}}(\Sigma \cup \mathrm{DC})$, and so for each sentence σ in $\mathscr{L}$, $\Sigma' \cup \mathrm{DC} \models^{\mathrm{pos}} \sigma$ iff $\Sigma \cup \mathrm{DC} \models^{\min} \sigma$.*

Proof. Let $\mathscr{C}$ be the set of constant symbols of the first order language $\mathscr{L}$. Let Σ be a set of P-clauses over $\mathscr{L}$. For each $\mathbf{I} \in \mathscr{M}^{\mathrm{pos}}(\Sigma \cup \mathrm{DC})$ let $\phi_{\mathbf{I}} = \bigwedge \{P(\mathbf{c}) \mid P$ is a predicate in $\mathscr{L}$, $\mathbf{c}$ is a vector of constants, and $\mathbf{I} \models^{\mathrm{fol}} P(\mathbf{c})\}$. Now, let $\Phi = \bigvee \{\phi_{\mathbf{I}} \mid \mathbf{I} \in$

$\mathcal{M}^{pos}(\Sigma \cup DC)\}$. (Note that Φ is finite because $\mathcal{L}$ is finite.) And finally, let Σ' be the set of clauses corresponding to the conjunctive normal form of Φ.

- We now argue that $\mathcal{M}^{min}(\Sigma' \cup DC) \subseteq \mathcal{M}^{pos}(\Sigma \cup DC)$. So let $\mathbf{I} = \langle D, I \rangle \in \mathcal{M}^{min}(\Sigma' \cup DC)$. Then, $\mathbf{I} \models^{fol} \Phi \cup DC$ and so, there exists $\mathbf{J} \in \mathcal{M}^{pos}(\Sigma \cup DC)$ such that $\mathbf{I} \models^{fol} \phi_{\mathbf{J}}$.

It is straightforward to verify that $\mathbf{J} \models^{fol} \bar{I}$, where $\bar{I}$ is as defined in Section 3. Hence using Lemma 4.8, we deduce that there exists $\mathbf{K}$ having the same pre-interpretation as $\mathbf{I}$ such that $\mathbf{K} \leqslant \mathbf{I}$, $\mathbf{K} \models^{fol} \Sigma \cup DC$, and $\mathbf{K} \models^{fol} \Phi \cup DC$ whence $\mathbf{K} \models^{fol} \Sigma' \cup DC$. Because $\mathbf{I}$ is in $\mathcal{M}^{min}(\Sigma' \cup DC)$, it follows that $\mathbf{K} = \mathbf{I}$. On the other hand, using the fact that $\mathbf{J} \in \mathcal{M}^{pos}(\Sigma \cup DC)$ it is straightforward to show that $\mathbf{K} \in \mathcal{M}^{pos}(\Sigma \cup DC)$. In conclusion, we have $\mathbf{I} \in \mathcal{M}^{pos}(\Sigma \cup DC)$.

- To see that $\mathcal{M}^{pos}(\Sigma \cup DC) \subseteq \mathcal{M}^{min}(\Sigma' \cup DC)$, let $\mathbf{I} \in \mathcal{M}^{pos}(\Sigma \cup DC)$. Clearly $\mathbf{I} \models^{fol} \Sigma' \cup DC$. Assume there exists $\mathbf{J} \in \mathcal{M}^{min}(\Sigma' \cup DC)$ such that $\mathbf{J} \leqslant \mathbf{I}$. By the previous part of the proof, we have that $\mathbf{J} \in \mathcal{M}^{pos}(\Sigma \cup DC)$. From this it follows that $\mathbf{J} \in \mathcal{M}^{min}(\Sigma \cup DC)$. Since $\mathbf{I} \in \mathcal{M}^{min}(\Sigma \cup DC)$, we conclude that $\mathbf{I} = \mathbf{J}$. In conclusion, $\mathbf{I} \in \mathcal{M}^{min}(\Sigma' \cup DC)$. ▌

Positivism provides the means to express a rich semantics in an elegant and terse manner. Speaking intuitively, in the above proof the minimalist database Σ' constructed to simulate the positivist database Σ expresses this semantics in a way that could be more difficult to understand and hides the intended meaning of the clauses in Σ.

The following result shows that minimalism cannot simulate positivism in the general case, that is, if domain closure is not assumed.

THEOREM 6.4. *There exists a positivist database* $(\mathcal{L}, \Sigma)$ *such that there is no minimalist database* $(\mathcal{L}, \Sigma')$ *with the property that* $\forall \sigma \in \mathcal{L}$, $\Sigma \models^{pos} \sigma$ *iff* $\Sigma' \models^{min} \sigma$.

In order to proceed to the proof, we need

DEFINITION. Let $\mathbf{I} = \langle D, I \rangle$ be an interpretation over the language $\mathcal{L}$ having $\{c_1, ..., c_n\}$ as constant symbols. A *sub-interpretation* of $\mathbf{I}$, is an interpretation $\hat{\mathbf{I}} = (\hat{D}, \hat{I})$, where

- $\{I(c_i) \mid 1 \leqslant i \leqslant n\} \subseteq \hat{D} \subseteq D$, and
- $\hat{I}$ is the restriction of I on $\hat{D}$.

It is well-known that

LEMMA 6.5. *Let* Σ *be a set of universally quantified sentences over* $\mathcal{L}$, $\mathbf{I}$ *be an interpretation of* $\mathcal{L}$, *and* $\hat{\mathbf{I}}$ *a sub-interpretation of* $\mathbf{I}$. *Then* $\mathbf{I} \models^{fol} \Sigma$ *implies* $\hat{\mathbf{I}} \models^{fol} \Sigma$.

Proof of Theorem 6.4. Let $\mathcal{L}$ be the first-order language having the constant symbol a, and the unary predicate symbols P, Q, and R. Let Σ be $\{P(a), \neg P(x) \rightarrow Q(a), \neg Q(a) \rightarrow R(a)\}$. Now suppose that Σ' is a minimalist database such that

$\forall \sigma \in \mathscr{L}$, $\Sigma \models^{\text{pos}} \sigma$ iff $\Sigma' \models^{\text{min}} \sigma$. Let $\tau = R(a) \vee \exists x(\neg P(x))$. We claim that $\Sigma \models^{\text{pos}} \tau$ but $\Sigma' \not\models^{\text{min}} \tau$.

■ To see that $\Sigma \models^{\text{pos}} \tau$, let $\mathbf{I} = \langle D, I \rangle \in \mathscr{M}^{\text{pos}}(\Sigma)$. Assume that $\mathbf{I} \models^{\text{fol}} \forall x P(x)$. Since $\mathbf{I}$ is a justified model of Σ, $\mathbf{I} \models^{\text{fol}} Q(a)$ iff $Q(\bar{a})$ is justified in $\mathbf{I}$ by some clause in $\Sigma \cup \text{DC}$, where $\bar{a} = I(a)$. Now, the only clause in Σ with the predicate Q occurring in its consequence is $\neg P(x) \to Q(a)$. By assumption, $\mathbf{I} \models^{\text{fol}} \forall x P(x)$, and so Σ does not justify $Q(\bar{a})$ in $\mathbf{I}$. Also, because $\mathbf{I}$ is minimal, for each pair e, d in D we have $d \simeq_{\mathbf{I}} e$ iff $d = e$. It follows that no clause of EQ justifies $Q(\bar{a})$ in $\mathbf{I}$. Thus $\mathbf{I} \models^{\text{fol}} \neg Q(a)$. Finally, since $\neg Q(a) \to R(a) \in \Sigma$ and $\mathbf{I} \models^{\text{fol}} \Sigma$, $\mathbf{I} \models^{\text{fol}} R(a)$. Thus $\mathbf{I} \models^{\text{fol}} R(a) \vee \exists x(\neg P(x))$, i.e., $\mathbf{I} \models^{\text{fol}} \tau$.

Assume now that $\mathbf{I} \not\models^{\text{fol}} \forall x P(x)$. Then $\mathbf{I} \models^{\text{fol}} \neg \exists x P(x)$ and $\mathbf{I} \models^{\text{fol}} \tau$.

■ We now show that $\Sigma \not\models^{\text{pos}} \neg Q(a) \vee R(a)$. Let $\mathbf{I} = (\{\bar{a}, d\}, I)$, where $I(a) = \bar{a}$, and the true facts for I are: $P(\bar{a})$, and $Q(\bar{a})$. It is easily verified that $\mathbf{I} \in \mathscr{M}^{\text{pos}}(\Sigma)$ and $\mathbf{I} \not\models^{\text{fol}} \neg Q(a) \vee R(a)$.

■ Turning to Σ', we first show that $\Sigma' \models^{\text{min}} P(a)$. Since $\Sigma \models^{\text{fol}} P(a)$, $\Sigma \models^{\text{pos}} P(a)$. By choice of Σ', $\Sigma' \models^{\text{min}} P(a)$.

■ We are now prepared to show that $\Sigma' \not\models^{\text{min}} \tau$. By assumption, $\forall \sigma \in \mathscr{L}$, $\Sigma \models^{\text{pos}} \sigma$ iff $\Sigma' \models^{\text{min}} \sigma$.

The preceding step of the proof implies that $\Sigma' \not\models^{\text{min}} \neg Q(a) \vee R(a)$. Therefore, there exists $\mathbf{J} = \langle D, J \rangle \in \mathscr{M}^{\text{min}}(\Sigma')$ such that $\mathbf{J} \not\models^{\text{fol}} \neg Q(a) \vee R(a)$, i.e., $\mathbf{J} \models^{\text{fol}} Q(a) \wedge \neg R(a)$. Also, $\Sigma' \models^{\text{min}} P(a)$ as noted above, so $\mathbf{J} \models^{\text{fol}} P(a)$. Let $\bar{a} = I(a)$ and consider the sub-interpretation $\hat{\mathbf{J}} = (\{\bar{a}\}, \hat{J})$ of $\mathbf{J}$, where $\hat{J}$ is the restriction of J on $\{\bar{a}\}$. Thus, the true facts for $\hat{J}$ are: $P(\bar{a})$, and $Q(\bar{a})$.

By Lemma 6.5, we have that $\hat{\mathbf{J}} \models^{\text{fol}} \Sigma'$. Thus there exists $\mathbf{K} = (\{\bar{a}\}, K) \in \mathscr{M}^{\text{min}}(\Sigma')$ such that $\mathbf{K} \leqslant \hat{\mathbf{J}}$. By the third step of the proof, we have that $\mathbf{K} \models^{\text{fol}} P(a)$ and so either $\mathbf{K} = \hat{\mathbf{J}}$ or $\mathbf{K}$ is defined by the following true fact: $P(\bar{a})$. In either case, $\mathbf{K} \not\models^{\text{fol}} \tau$ and so $\Sigma' \not\models^{\text{min}} \tau$. ∎

To understand the intuition behind the above proof more completely, we note that (up to isomorphism), the only positivistic model $\mathbf{I}_1$ of Σ with a 1-element domain $\{\bar{a}\}$ is characterized by the true facts $P(\bar{a})$, and $R(\bar{a})$; and (up to isomorphism) the only positivistic model of Σ with 2-element domain $\{\bar{a}, d\}$ is characterized by the true facts $P(\bar{a})$, and $Q(\bar{a})$. Thus the presence of a non-constant domain element has two fundamentally distinct implications:

- The "insertion" of the positive ground literal $Q(a)$.
- The "deletion" of the positive ground literal $R(a)$.

The above proof relies on "deletion," or more generally on the fact that the presence of a non-constant domain element can cause such a "deletion." As indicated by the following, for each $n > 0$, examples can be constructed such that the presence of at least n non-constant domain elements are needed to cause a "deletion.")

EXAMPLE 6.6. Let $\mathscr{L}$ be the first-order language having the constant symbols $c_1 \cdots c_k$, $k \geqslant 2$, the n-ary predicate P, and the unary predicate Q. Consider the positivist database $\Sigma = \{P(\mathbf{t}) \mid \mathbf{t}$ is a sequence of terms in $c_1 \cdots c_k, x_1 \cdots x_{n-1}\} \cup \{\neg P(x_1 \cdots x_n) \to Q(c_1), \neg Q(c_1) \to Q(c_2)\}$. Each positivistic model of Σ with less than n non-constant domain elements asserts $Q(c_1)$ and $\neg Q(c_2)$, but each positivistic model of Σ with at least n non-constant domain elements will assert $\neg Q(c_1)$ and thus $Q(c_2)$.

Intuitively, its seems that simulation of positivism by minimalism in the static case cannot be established in general because of this "deletion" effect. On the other hand, the following example exhibits a positivist database such that the addition of non-constant domain elements results exclusively in "insertions" and no "deletion." We then show how this positivist database can be simulated by a minimalist database.

EXAMPLE 6.7. Let $\Sigma = \{P(a), \neg P(x) \to Q(a)\}$ be a positivist database. Then consider the minimalist database $\Sigma' = \{P(a), x \simeq a \vee Q(a)\}$. It is clear that Σ' satisfies

$$\forall \sigma \in \mathscr{L}, \qquad \Sigma \models^{\text{pos}} \sigma \qquad \text{iff} \quad \Sigma' \models^{\text{min}} \sigma.$$

Although we do not pursue formally the question here, it would be interesting to determine whether the following is true: If a positivist database Σ has the (intuitive) property that "the addition of non-constant domain elements does not lead to deletions," then Σ can be simulated by a minimalist database. (It can be shown in these cases that the explicit use of the equality predicate $\simeq$ is needed in the construction of the minimalist database.)

We conclude our discussion of static simulation of positivism by minimalism by showing that "local" databases can be simulated by minimalist databases.

DEFINITION. A P-clause σ is *local* if each variable occuring in σ occurs in a positive literal in the premise of σ. A database Σ is *local* if each P-clause of Σ is local.

Note that the above definition of a local database is slightly stronger than the covering axiom of [S1]. Intuitively, localness is sufficient to guarantee that non-constant domain elements do not cause "deletions." More specifically, the following result states that if a database Σ is local then all the facts of a positivistic model of Σ involving non-constant elements are false.

LEMMA 6.8. *Let Σ be a local positivist database over $\mathscr{L}$ with constant symbols $\mathscr{C}$, and let $\mathbf{I} \in \mathscr{M}^{\text{pos}}(\Sigma)$. Let P be a k-ary predicate and $\mathbf{d} = d_1 \cdots d_k$ be a vector of domain elements such that there exists i in $[1 \cdots k]$, $\forall c \in \mathscr{C}$, $d_i \not\simeq_{\mathbf{I}} I(c)$. Then $I[P](\mathbf{d}) = 0$.*

Proof. Let $\mathbf{I} \in \mathcal{M}^{\text{pos}}(\Sigma)$. Suppose there is a k-ary predicate P and a vector of domain elements $\mathbf{d} = d_1 \cdots d_k$ such that there exists i in $[1 \cdots k]$, $\forall c \in \mathscr{C}$, $d_i \not\simeq_{\mathbf{I}} I(c)$, and $I[P](\mathbf{d}) = 1$. Define $\mathbf{J} = \langle D, J \rangle$ by: $J[Q](\mathbf{e}) = 1$ iff $I[Q](\mathbf{e}) = 1$, $\mathbf{e} = e_1 \cdots e_k$, and for each i in $[1 \cdots k]$ there exists a constant c such that $e_i \simeq_{\mathbf{I}} I(c)$. Clearly, $\mathbf{J}$ is an E-interpretation, $\mathbf{J} \leqslant \mathbf{I}$ and $\mathbf{J} \neq \mathbf{I}$.

We now show that $\mathbf{J} \models^{\text{fol}} \Sigma$. In order to do so, consider a P-clause $\sigma = L_1 \cdots L_n \to M_1 \cdots M_m$ in Σ with variables $x_1 \cdots x_1$. Let μ be an assignment for $x_1 \cdots x_1$ and suppose that for j in $[1 \cdots n]$, $J(L_j[\mu]) = 1$. Because σ is local, each variable x_i, i in $[1 \cdots l]$, occurs in some positive literal L_i and by construction of $\mathbf{J}$, $\exists c \in \mathscr{C}$, $\mu[x_i] \simeq_{\mathbf{I}} I(c)$. Thus $I(L_j[\mu]) = 1$ for j in $[1 \cdots n]$. Since $\mathbf{I} \models^{\text{fol}} \Sigma$, this implies that $I(M_{j_0}[\mu]) = 1$ for some j_0 in $[1 \cdots m]$, so $J(M_{j_0}[\mu]) = 1$ for some j_0 in $[1 \cdots m]$, and $J(\sigma[\mu]) = 1$.

In conclusion, $\mathbf{J} \models^{\text{fol}} \Sigma$ and so $\mathbf{I}$ is not a minimal model of Σ, a contradiction. ∎

We now have

PROPOSITION 6.9. *Let Σ be a local positivist database over $\mathscr{L}$. Then there exists a minimalist database Σ' such that $\forall \sigma \in \mathscr{L}$, $\Sigma \models^{\text{pos}} \sigma$ iff $\Sigma' \models^{\text{min}} \sigma$.*

Sketch of Proof. The proof of this result is very similar to the proof of Theorem 6.3. In particular, let Σ' be constructed as in that proof. Using Lemma 6.8 and the fact that in each positivistic model of Σ', each predicate is false on any vector $\mathbf{d}$ involving at least one non-constant domain element, it can be shown as in the proof of Theorem 6.3 that $\mathcal{M}^{\text{min}}(\Sigma') = \mathcal{M}^{\text{pos}}(\Sigma)$. ∎

As stated above, minimalism and positivism are non-monotonic. That is, given a minimalist database Σ, there may be formulas α, β such that $\Sigma \models^{\text{min}} \alpha$ but $\Sigma \cup \{\beta\} \not\models^{\text{min}} \alpha$ (and analogously for the positivistic case). This raises the fundamental question of whether the non-monotonicity of the two approaches is essentially equivalent. More specifically, if a positivist database Σ and a minimalist database Σ' are equivalent do they remain equivalent when updated? While defining the meaning of arbitrary updates to (sets of closed formulas which represent) databases is an essentially unresolved problem (e.g., see [KUV]), consideration of simple insertions alone is sufficient to obtain some interesting results.

DEFINITION. Let $(\mathscr{L}, \Sigma)$ be a (positivist or minimalist) database. A (*simple*) *insertion* to $(\mathscr{L}, \Sigma)$ is an expression of either the form:

- $P(\mathbf{c})$, where P is a predicate in $\mathscr{L}$ and $\mathbf{c}$ is a sequence of constants in $\mathscr{L}$; or
- new(c), where c is a constant symbol not in $\mathscr{L}$.

The *result* of an insertion $\imath$ to $\mathscr{D} = (\mathscr{L}, \Sigma)$, denoted $\mathscr{D}\imath$, is the database defined by:

- If $\imath = P(\mathbf{c})$, then $\mathscr{D}\imath = (\mathscr{L}, \Sigma \cup \{P(\mathbf{c})\})$
- If $\imath = \text{new}(c)$, then $\mathscr{D}\imath = (\mathscr{L} \cup \{c\}, \Sigma)$.

A *valid sequence* of insertions to $(\mathscr{L}_0, \Sigma_0)$ is a (finite) sequence $\boldsymbol{\imath} = \imath_1, ..., \imath_n$ of insertions such that for each i in $[1 \cdots n]$, $\imath_1, ..., \imath_{i-1}$, is a valid sequence of insertions to $(\mathscr{L}_0, \Sigma_0)$ and $\imath_i$ is an insertion to $(\mathscr{L}_{i-1}, \Sigma_{i-1})$, where $(\mathscr{L}_k, \Sigma_k) = (\mathscr{L}_{k-1}, \Sigma_{k-1})\imath_k$ for k in $[1 \cdots i-1]$. The *result* of a valid sequence of insertions $\boldsymbol{\imath}$ to $(\mathscr{L}_0, \Sigma_0)$, denoted $(\mathscr{L}_0, \Sigma_0)\boldsymbol{\imath}$, is the database $(\mathscr{L}_n, \Sigma_n)$.

Note here that we do not consider insertions of new predicate symbols in the database nor insertions of arbitrary clauses.

The following interesting result gives some indication as to the expressive power of minimalism in the presence of insertions. Intuitively, it says that two minimalist databases are equivalent *in the context of first-order logic* iff they are equivalent under insertions in the context of minimalism.

PROPOSITION 6.10. *Let $\mathscr{L}$ be a first-order language, and Σ, Σ' be two minimalist databases over $\mathscr{L}$. Then $\mathscr{M}^{\text{fol}}(\Sigma) = \mathscr{M}^{\text{fol}}(\Sigma')$ iff $\mathscr{M}^{\text{min}}(\Sigma\boldsymbol{\imath}) = \mathscr{M}^{\text{min}}(\Sigma'\boldsymbol{\imath})$ for each valid sequence of insertions $\boldsymbol{\imath}$ to Σ.*

Proof. It is clear that if $\Sigma \equiv^{\text{fol}} \Sigma'$ then for each $\boldsymbol{\imath}$, $\Sigma\boldsymbol{\imath} \equiv^{\text{fol}} \Sigma'\boldsymbol{\imath}$, and so $\mathscr{M}^{\text{min}}(\Sigma\boldsymbol{\imath}) = \mathscr{M}^{\text{min}}(\Sigma'\boldsymbol{\imath})$. For the converse, suppose that $\Sigma \not\equiv^{\text{fol}} \Sigma'$. Without loss of generality, we can assume that there is a discriminant interpretation $\mathbf{I} = \langle D, I \rangle$ which satisfies Σ and violates some clause α in Σ'.

We now argue that there is a finite sub-interpretation $\mathbf{J} = \langle D', J \rangle$ of $\mathbf{I}$ which satisfies Σ and violates Σ'. Because α is a P-clause, α is universally quantified. Let $\mathbf{x}$ the vector of constants occuring in α. Because $\mathbf{I} \not\models^{\text{fol}} \alpha$, there is some assignment μ for $\mathbf{x}$ such that $\mathbf{I}(\alpha[\mu]) = 0$. Let $D' = \{I(c) \mid c \text{ is a constant symbol in } \mathscr{L}\} \cup \{\mu(x) \mid x \text{ is a variable in } \alpha\}$, and let $\mathbf{J} = \langle D', J \rangle$, where J is the restriction of I on D'. It is clear that $\mathbf{J} \not\models^{\text{fol}} \alpha$ and hence violates Σ'. Also Lemma 6.5 implies that $\mathbf{J} \models^{\text{fol}} \Sigma$.

Let $c_1, ..., c_n$ be an enumeration of the constant symbols in $\mathscr{L}$, and let $d_1, ..., d_n, d_{n+1}, ..., d_{n+m}$ be an enumeration of D' such that $d_i = J(c_i)$ for $i \in [1 \cdots n]$. Let $c_{n+1}, ..., c_{n+m}$ be new constant symbols. Finally, let

- $\imath_1 = \text{new}(c_{n+1}) \cdots \text{new}(c_{n+m})$, and
- $\imath_2$ be an enumeration of $\Omega = \{P(c_{i_1}, ..., c_{i_l}) \mid P \text{ is a predicate symbol in } \mathscr{L}, i_j \in [1 \cdots n+m] \text{ for } j \text{ in } [1 \cdots l], \text{ and } J[P](d_{i_1}, ..., d_{i_l}) = 1\}$.

We now define an interpretation $\mathbf{K} = \langle D', K \rangle$ over language $\mathscr{L}$ augmented with $\{c_{n+1}, ..., c_{n+m}\}$. Specifically, define $\mathbf{K}$ so that $K(c_i) = d_i$ for i in $[1 \cdots n+m]$, and for each P and $\mathbf{d}$ from D', $K[P](\mathbf{d}) = J[P](\mathbf{d})$. (Intuitively, $\mathbf{K}$ is the same as $\mathbf{J}$, with the new constants interpreted in the natural manner.)

It is clear that $\mathbf{K} \models^{\text{fol}} \Sigma$ and $\mathbf{K} \models^{\text{fol}} \Omega$ by construction. In fact, $\mathbf{K}$ is a minimal model of Ω, and so $\mathbf{K} \in \mathscr{M}^{\text{min}}(\Sigma\boldsymbol{\imath}_1\boldsymbol{\imath}_2)$. On the other hand, $\mathbf{K} \not\models^{\text{fol}} \Sigma'$ because $\mathbf{J} \not\models^{\text{fol}} \Sigma'$. Thus, $\mathbf{K} \notin \mathscr{M}^{\text{fol}}(\Sigma' \cup \Omega)$, and hence $\mathbf{K} \notin \mathscr{M}^{\text{min}}(\Sigma'\boldsymbol{\imath}_1\boldsymbol{\imath}_2)$. ∎

From Proposition 6.2 and considering the types of insertions allowed, it is easily verified that

THEOREM 6.11. *Let $\mathscr{L}$ be a first-order language. For each minimalist database Σ there is a positivist database Σ' such that for each valid sequence of insertions $\imath$ to Σ and each closed formula σ, $\Sigma'\imath \models^{\text{pos}} \sigma$ iff $\Sigma\imath \models^{\text{min}} \sigma$.*

Intuitively speaking, the preceding result states that it is possible to simulate the non-monotonic behavior of minimalism by positivist non-monotonicity. On the other hand, the reverse is not possible.

THEOREM 6.12. *Let $\mathscr{L}$ be a first-order language. There exists a positivist database Σ such that no minimalist database Σ' has the following property: For each valid sequence $\imath$ of insertions to $\Sigma \cup \mathrm{DC}$ and each closed formula σ, $(\Sigma' \cup \mathrm{DC})\imath \models^{\text{min}} \sigma$ iff $(\Sigma \cup \mathrm{DC})\imath \models^{\text{pos}} \sigma$.*

Proof. It suffices to examine the propositional case. Consider the propositional language $\mathscr{L} = \{P, Q\}$, and let $\Sigma = \{\neg P \rightarrow Q\}$. Suppose that Σ' is a minimalist database such that for all valid sequences $\imath$ of insertions to Σ and all σ in $\mathscr{L}$, $\Sigma'\imath \models^{\text{min}} \sigma$ iff $\Sigma\imath \models^{\text{pos}} \sigma$.

Let $\imath = P$ (that is, $\imath$ inserts P). Then $\Sigma\imath \not\models^{\text{pos}} Q$. We will show that $\Sigma'\imath \models^{\text{min}} Q$.

Note that $\Sigma \models^{\text{pos}} Q \wedge \neg P$, and so $\Sigma' \models^{\text{min}} Q \wedge \neg P$. Thus the only minimal model of Σ' is $\{Q\}$. This implies that $\mathscr{M}^{\text{fol}}(\Sigma') = \{\{P, Q\}, \{Q\}\}$ or $\mathscr{M}^{\text{fol}}(\Sigma') = \{\{Q\}\}$. We consider these two cases separately:

Case 1. $\mathscr{M}^{\text{fol}}(\Sigma') = \{\{P, Q\}, \{Q\}\}$. In this case, $\Sigma' \equiv^{\text{fol}} \{Q\}$. Then $\Sigma'\imath = \Sigma \cup \{P\} \equiv^{\text{fol}} \{P, Q\}$. Thus, $\mathscr{M}^{\text{min}}(\Sigma'\imath) = \{\{P, Q\}\}$, and so $\Sigma'\imath \models^{\text{min}} Q$.

Case 2. $\mathscr{M}^{\text{fol}}(\Sigma') = \{\{Q\}\}$. This implies that $\Sigma' \equiv^{\text{fol}} \{Q \wedge \neg P\}$. It follows that $\Sigma'\imath \equiv^{\text{fol}} \{Q \wedge \neg P, P\}$. Thus $\Sigma'\imath$ has no first-order models, and hence no minimal models. $\Sigma'\imath$ therefore implies all formulas under minimalism, and in particular, $\Sigma'\imath \models^{\text{min}} Q$. ∎

A non-propositional local positivist database Σ which cannot be simulated by any minimalist database under insertions (even when domain closure is assumed) is:

$$\Sigma = \{P(a), P(x) \wedge \neg Q(x) \rightarrow R(x)\}.$$

Note that the NAF algorithm will terminate on any query over this database Σ. In view of Theorem 5.10, this indicates that in the context of simple insertions, the (intuitive) NAF semantics of Σ cannot be simulated by minimalism.

The preceding theorems show that in the context of simple insertions, the expressive power of the positivism approach is strictly greater than that of the minimalism approach.

7. STRATIFICATION

In this section we briefly discuss "stratification," an approach which is related to positivism, for defining the semantics of databases involving negation in the premises. This notion was introduced in recent papers. We focus here on two of these. The first of these [ABW] focusses on logic programming, and does not permit disjunction in the consequents. The second [P1] builds on [ABW], and also on the notion of positivism as introduced in [BH] and the current paper. As we shall see, in some cases the semantics of stratification is intuitively more appealing than that of positivism. It is important to note that the results of Section 6 concerning the expressive power of minimalism and positivism have natural extension to stratification.

As with positivism, the starting point when defining database semantics using stratification is a set Σ of P-clauses. However, this set must satisfy a certain hierarchical condition, which is reminiscent of, but much more general than, the hierarchy condition of [Cl]. Intuitively, the condition permits recursion among predicates which appear positively in premises, but does not permit it when they appear negatively in the premises:

DEFINITION. Let $\mathscr{A}$ be an alphabet with predicates $\mathscr{P}$ and constants $\mathscr{C}$. Let Σ be a positivist database over $\mathscr{A}$. A *stratification* for Σ is a sequence $\mathscr{S} = \mathscr{P}_1, ..., \mathscr{P}_n$, where $\mathscr{P}_1, ..., \mathscr{P}_n$ is a partition of $\mathscr{P}$ such that for each clause $L_1 \wedge \cdots \wedge L_n \to L_{n+1} \vee \cdots \vee L_{n+m}$ there is some j such that:

- if $i \in [1 \cdots m]$, then the predicate of L_{n+i} is in $\mathscr{P}_j$;
- if $i \in [1 \cdots n]$, and L_i is the positive literal $P(\mathbf{t})$, then the predicate P is in $\mathscr{P}_k$ for some $k \leqslant j$; and
- if $i \in [1 \cdots n]$ and L_i is the negative literal $\neg P(\mathbf{t})$, then the predicate P is in $\mathscr{P}_k$ for some $k < j$.

Σ is *stratifiable* if there exists a stratification for Σ.

Speaking intuitively, stratified models are obtained by constructing positivistic models at each level of a stratification. Following much of the literature (an exception is [P2]), we assume DC and focus exclusively on Herbrand interpretations in this discussion.

DEFINITION. Let Σ be a positivist database over alphabet $\mathscr{A}$ with stratification $\mathscr{S} = \mathscr{P}_1, ..., \mathscr{P}_n$. For each i, let Σ_i be the set of clauses of Σ whose consequent predicates are in $\mathscr{P}_i$. A Herbrand interpretation $\mathbf{I}$ is a *stratified model* of Σ according to $\mathscr{S}$ if $\mathbf{I} = \mathbf{I}_n$ for some sequence $\mathbf{I}_1, ..., \mathbf{I}_n$ of interpretations over $\mathscr{A}$ such that

$$\forall j \in [1 \cdots n], \qquad \mathbf{I}_j \in \mathscr{M}^{\text{pos}}(\text{DC} \cup \Sigma_j \cup \Omega_{\mathbf{I}_{j-1}}),$$

where for an interpretation $\mathbf{J}$, $\Omega_{\mathbf{J}}$ denotes the set of (positive) atoms which are satisfied by $\mathbf{J}$.

Note here that it is not sufficient in the above definition to take minimal models of $(\text{DC} \cup \Sigma_j \cup \Omega_{\mathbf{I}_{j-1}})$. Let us consider the database $\Sigma = \{P \to R, \neg R \to S\}$. Then Σ

is obviously a stratifiable database and $\{\neg P, \neg R, S\}$ is the stratified model of Σ. Assuming that minimal models are taken instead of positivist models in the above definition of stratified models, it turns out then that $\{\neg P, R, \neg S\}$ is also a stratified model, although intuitively, it does not correspond to the intended meaning of Σ.

It can be shown [ABW, P1] that the semantics of stratified models is independent of the chosen stratifications; in other words, if $\mathbf{I}$ is a stratified model of Σ according to a stratification $\mathscr{S}$ for Σ and if $\mathscr{S}'$ is another stratification for Σ, then $\mathbf{I}$ is a stratified model of Σ according to $\mathscr{S}'$. Thus, we define $\mathscr{M}^{\text{strat}}(\Sigma)$ to be the set of all stratified models of Σ according to some (any) stratification for Σ.

The following proposition provides an easily verified, non-constructive characterization of stratified models. This characterization is related to the characterization of stratified models using prioritorized circumscription [Li2, Mc2] presented in [P1].

PROPOSITION 7.1. *Let Σ be a positivist database over $\mathscr{A}$ with stratification $\mathscr{S} = \mathscr{P}_1, ..., \mathscr{P}_n$, and let $\mathbf{I}$ be an Herbrand interpretation over $\mathscr{A}$. Let $\mathbf{I}_j$ be the interpretation obtained from $\mathbf{I}$ be deleting all true facts involving predicates not in $\bigcup_{i=1}^{j} \mathscr{P}_i$; and let Ω_j be the set of clauses of Σ whose consequents are contained in $\bigcup_{i=1}^{j} \mathscr{P}_i$. Then $\mathbf{I} \in \mathscr{M}^{\text{strat}}(\Sigma)$ iff for each j in $[1 \cdots n]$, $\mathbf{I}_j \in \mathscr{M}^{\min}(\Omega_j)$.*

The following example illustrates the above definitions and result and also shows that in some cases the semantics of stratification and positivism coincide.

EXAMPLE 7.2. Let Σ be the positivist database from Example 2.3. One stratification for Σ is $\mathscr{S} = \{\text{STUD, GRAD, UGRAD, TA}\}, \{\text{ADV-GRAD}\}$; and another one is $\mathscr{S}' = \{\text{STUD}\}, \{\text{GRAD, UGRAD}\}, \{\text{TA}\}, \{\text{ADV-GRAD}\}$. Using $\mathscr{S}$, Σ_1 will contain σ_1 and all atoms of Σ except ADV-GRAD(Mary); and $\Sigma_2 = \{\sigma_2, \text{ADV-GRAD(Mary)}\}$.

Consider the (Herbrand) interpretation $\mathbf{I}$ containing all of the atoms of Σ and GRAD(Zanja), ADV-GRAD(Zanja); let $\mathbf{I}_2 = \mathbf{I}$; and let $\mathbf{I}_1$ be the interpretation containing GRAD(Zanja) and all of the atoms of Σ except ADV-GRAD(Mary). Then the sequence $\mathbf{I}_1, \mathbf{I}_2$ satisfies the definition of stratified (and also the conditions of Proposition 7.1); therefore $\mathbf{I}$ is a stratified model of Σ.

It is easily verified that Σ has exactly two stratified models and that these are the only positivistic Herbrand models of Σ. More generally, suppose that Σ' contains $\{\sigma_1, \sigma_2\}$ and an arbitrary set of positive atoms over the language of Σ. It can be shown that the set of stratified models of Σ' is equal to the set of positivistic Herbrand models of Σ'.

Assuming domain closure, it can be shown for any stratifiable positivist database Σ that the set of stratified models of Σ is contained in the set of positivistic Herbrand models of Σ. In particular, then, the set of stratified models of Σ is a subset of the minimal models of Σ.

The following example illustrates that the semantics of stratification may be different than the semantics of positivism.

EXAMPLE 7.3. Consider the propositional database $\Sigma = \{P \vee Q, \ \neg P \rightarrow R, \neg Q \rightarrow R\}$. The only stratification is $\{P, Q\}$, $\{R\}$, and the stratifiable models are $\mathbf{I}_1 = \{P, R\}$ and $\mathbf{I}_2 = \{Q, R\}$. However, there is a third positivistic model of Σ, namely $\mathbf{I}_3 = \{P, Q\}$.

As argued in [P1], it appears to be intuitively more natural to associate only the models $\mathbf{I}_1$ and $\mathbf{I}_2$ with Σ. This is based on the intuition that the P-clauses $\neg P \rightarrow R$ and $\neg Q \rightarrow R$ should not "affect" the truth of the propositions P and Q, because P and Q occur in these clauses only as (negative) premises. Thus, while the semantics of stratification is not defined for all positivist databases, it eliminates counter-intuitive models in some cases.

Our final example illustrates that in positivism the inclusion of a clause $P(x) \rightarrow P(x)$ has the effect of "neutralizing" the impact of justification on the predicate P. This is easily illustrated in the propositional case:

EXAMPLE 7.4. (a) Recall that the propositional positivist database $\Sigma_1 = \{\neg P \rightarrow Q\}$ has one positivistic model, namely $\{Q\}$. This is also the only stratified model of Σ_1. Consider $\Sigma_2 = \{P \rightarrow P, \ \neg P \rightarrow Q\}$. This has only $\{Q\}$ as a stratified model, but both $\{P\}$ and $\{Q\}$ as positivistic models. In particular, the presence of $P \rightarrow P$ in Σ_2 essentially provides a mechanism whereby P is justified in any model in which P is true. (On the other hand, the only positivistic model of $\{P \rightarrow P\}$ is the empty model; in this case the requirement of minimalism "overrides" the effect of $P \rightarrow P$.)

(b) We now present a more provocative illustration of clauses $P \rightarrow P$ neutralizing justification. Let $n \geqslant 2$ and

$$\Sigma_3 = \{P_1 \vee P_2\} \cup \{\neg P_i \rightarrow P_{i+1} \mid 2 \leqslant i \leqslant n\}.$$

There are two positivistic models of Σ_3, namely

$$\mathbf{I}_1 = \{P_{2i+1} \mid 0 \leqslant i < n/2\},$$

$$\mathbf{I}_2 = \{P_{2i} \mid 1 \leqslant i \leqslant n/2\};$$

these are also the only stratified models of Σ_3.

Now let k be fixed, $2 \leqslant k \leqslant n/2$. Then the stratified models of $\Sigma_3 \cup \{P_{2k} \rightarrow P_{2k}\}$ are again $\mathbf{I}_1$ and $\mathbf{I}_2$. On the other hand, there are three positivistic models of this database, namely $\mathbf{I}_1, \mathbf{I}_2$, and $\mathbf{I}_3 = \{P_{2i+1} \mid 0 \leqslant i < k\} \cup \{P_{2j} \mid k \leqslant j \leqslant n/2\}$. Thus, the inclusion of $P_{2k} \rightarrow P_{2k}$ permits the propagation of true odd propositional variables in $\mathbf{I}_1$ to be "broken" at P_{2k}. Similar remarks apply for odd k.

As suggested in the Introduction, the results of Section 6 can be naturally extended to compare the expressive power of minimalism, positivism, and stratification. In fact, the proof technique of Theorem 6.3 can be applied to show that in the static case, each stratified database can be simulated by a minimalist database (and hence by a positivist database). In the static case both stratified and positivist databases can be simulated by minimalist databases and hence by each other. Also, because the counterexample database of Theorem 6.12 is stratifiable, we see that there is a stratifiable database which cannot be simulated by any minimalist database in the context of simple insertions. However, it remains open whether positivist databases can simulate stratifiable databases in the context of simple insertions, or vise versa.

Acknowledgments

The authors thank the members of the 1986 PODS Conference Program Committee for suggestions which led to a clearer exposition of the underlying concepts of this paper.

References

[ABW] K. Apt, H. Blair, and A. Walker, Towards a theory of declarative knowledge, *in* "Workshop on Foundations of Deductive Databases and Logic Programming, 1986," pp. 546–628.

[AvE] K. Apt and M. H. van Emden, Contributions to the theory of logic programming, *J. Assoc. Comput. Mach.* **29**, No. 3 (1982), 841–862.

[BF] N. Bidoit and C. Froidevaux, Minimalism subsumes default logic and circumscription in stratified logic programming, *in* "Proceedings, IEEE Symp. on Logic in Computer Science, 1987," pp. 89–97.

[BH] N. Bidoit and R. Hull, Positivism vs minimalism in deductive databases, *in* "Proceedings, ACM SIGACT-SIGMOD Symp. on Principles of Database Systems, 1986," pp. 123–132.

[BS] G. Bossu and P. Siegel, Saturation, nonmonotonic reasoning and the closed world assumption, *Artif. Intell.* **25** (1985), 13–63.

[ChL] C. L. Chang and R. C. T. Lee, "Symbolic Logic and Mechanical Theorem Proving," Academic Press, New York, 1973.

[Cl] K. L. Clark, Negation as failure, *in* "Logic and Databases" (H. Gallaire and J. Minker, Eds.), pp. 293–322, New York, 1978.

[ClT] K. L. Clark and S.-A. Tamlund (Eds.), "Logic Programming," Academic Press, New York, 1982.

[Co] A relational model for large shared data banks, *Comm. ACM* **13**, No. 6 (1970), 377–387.

[E] H. B. Enderton, "A Mahematical Introduction to Logic," Academic Press, New York, 1972.

[GMN] H. Gallaire, J. Minker, and J.-M. Nicolas, Logic and databases: A deductive approach, *Comput. Surveys* **16**, No. 2 (1984), 153–185.

[GPP] M. Gelfond, H. Przymusinska, and T. Przymusinski, The extended closed world assumption and its relationship with parallel circumscrition, *in* "Proceedings, ACM SIGACT-SIGMOD Symp. on Principles of Database Systems, 1986," pp. 133–139.

[KUV] G. M. Kuper, J. D. Ullman, and M. Vardi, On the equivalence of logical databases, *in* "Proceedings, ACM SIGACT-SIGMOD Symp. on Principles of Database Systems, 1984," pp. 221–228.

[Li1] V. Lifschitz, Closed-world databases and circumscrition, *Artif. Intell.* **27** (1985), 229–235.

[Li2] V. LIFSCHITZ, Computing circumscrition, *in* "Proceedings, Internat. Joint Conf. Artif. Intell., 1985," pp. 121–127.

[Ll] J. W. LLOYD, "Foundations of Logic Programing," Springer-Verlag, Berlin, 1984.

[Mc1] J. MCCARTHY, Circumscrition "A Form of Non-Monotonic Reasoning," *Artif. Intell.* **13** (1980), 27–39.

[Mc2] J. MCCARTHY, Applications of circumscrition to formalizing common sense knowledge, *in* "AAAI Workshop on Non-Monotonic Reasoning, 1984," pp. 295–323.

[Mi] J. MINKER, On indefinite databases and the closed world assumption, *in* "Proceedings, 6th Conf. on Automated Deduction, Lecture Notes in Computer Science Vol. 138," pp. 292–308, Springer-Verlag, New York/Berlin, 1982.

[P1] T. C. PRZYMUSINSKI, On the semantics of stratified deductive databases, *in* "Workshop on Foundations of Deductive Databases and Logic Programming, 1986," pp. 433–443.

[P2] T. C. PRZYMUSINSKI, On the declarative semantics of logic programs, invited paper, J. Logic Programming, in press.

[R1] R. REITER, On closed world data bases, *in* "Logic and Databases" (H. Gallaire and J. Minker, Eds.), pp. 55–76, Plenum, New York, 1978.

[R2] R. REITER, A logic for default reasoning, *Artif. Intell.* **13** (1980), 81–132.

[R3] R. REITER, Equality and domain closure in first-order databases, *J. Assoc. Comput. Mach.* **27**, No. 2 (1980), 235–249.

[R4] R. REITER, Towards a reconstruction of relational database theory, *in* "On Conceptual Modelling" (M. L. Brodie, J. Mylopoulos, and J. W. Schmidt, Eds.), pp. 191–238, Springer-Verlag, New York/Berlin, 1984.

[S1] J. C. SHEPHERDSON, Negation as failure: A comparison of Clark's completed data base and Reiter's closed world assumption, *J. Logic Programming* **1** (1984), 51–79.

[S2] J. C. SHEPHERDSON, Negation as failure, II, *J. Logic Programming* **1** (1985), 185–202.