Artificial Intelligence 52 (1991) 121–149
Elsevier

On the consistency of defeasible databases*

Moisés Goldszmidt** and Judea Pearl

Cognitive Systems Laboratory, Computer Science Department, 4731 Boelter Hall, University of California, Los Angeles, CA 90024, USA

Revised May 1991

Abstract

Goldszmidt, M. and J. Pearl, On the consistency of defeasible databases, Artificial Intelligence 52 (1991) 121–149.

We propose a norm of consistency for a mixed set of defeasible and strict sentences which, guided by a probabilistic interpretation of these sentences, establishes a clear distinction between exceptions, ambiguities and outright contradictions. A notion of entailment is then defined which represents a minimal core of beliefs that must follow from the database if one is committed to avoid inconsistencies.

The paper establishes necessary and sufficient conditions for consistency, and provides a simple decision procedure for testing the consistency of a database or whether a given sentence is entailed by the database. It is also shown that if all sentences are of *Horn* type, consistency and entailment can be tested in polynomial time. Finally, we discuss procedures for reasoning with inconsistent databases and identifying sentences directly responsible for the inconsistency.

1. Introduction

There is a sharp difference between exceptions and outright contradictions. Two statements like "typically penguins do not fly" and "red penguins fly", can be accepted as a description of a world in which *redness* defines an abnormal or exceptional type of penguins. However, the statements s_1: "typically birds fly" and s_2: "typically birds do not fly" stand in outright contradiction to each other. Whatever interpretation we give to "typically", it is hard to imagine a *world* containing birds in which both s_1 and s_2 would make sense simultaneously. Curiously, such conflicting pairs of sentences can perfectly coexist in

* This work was supported in part by National Science Foundation Grant #IRI-88-21444 and State of California MICRO 90-127. An earlier version of this paper was presented at the Workshop on Uncertainty in AI, August 1989.

** Supported by an IBM graduate fellowship 1990–92.

circumscriptive (McCarthy [19]) or *default logic* (Reiter [25]) theories. Using the *ab* predicate advocated by McCarthy [19], a straightforward way to represent them in the context of circumscription would be:

$$\begin{aligned} s_1' &: \forall x.\mathsf{bird}(x) \wedge \neg ab(x) \supset \mathsf{fly}(x)\,, \\ s_2' &: \forall x.\mathsf{bird}(x) \wedge \neg ab(x) \supset \neg\mathsf{fly}(x)\,, \end{aligned} \tag{1}$$

which is logically equivalent to $\forall x.\mathsf{bird}(x) \supset ab(x)$. Similarly, expressing s_1 and s_2 as *default rules*[1]

$$\begin{aligned} s_1'' &: \frac{\mathsf{bird}(x) : \mathsf{M}\,\mathsf{fly}(x)}{\mathsf{fly}(x)}\,, \\ s_1'' &: \frac{\mathsf{bird}(x) : \mathsf{M}\,\neg\mathsf{fly}(x)}{\neg\mathsf{fly}(x)}\,, \end{aligned} \tag{2}$$

default logic will produce two consistent sets of beliefs: One in which "birds fly" and one in which "birds do not fly".

We contend that a pair such as s_1 and s_2 is not normally used to encode the information that "all birds are *exceptional* (or *abnormal*)" as in the case of circumscription, or to express an *ambiguous* property[2] of birds as in the case of default logic. Rather, this kind of contradictory information is more likely to originate from an *unintentional* mistake. Remarkably, although the distinction between exceptions, ambiguity and contradictions is readily recognized by humans, there is no comprehensive analysis of such utterances in defeasible databases, one that could alert the user to the existence of contradictory, possibly unintended statements. This paper proposes a semantically sound norm for consistency, accompanied by effective procedures for testing inconsistencies and isolating their origins.

It is tempting to assume that pairs such as s_1 and s_2 constitute the only source of inconsistency and that once we eliminate such contradictory pairs, the remaining database will be consistent, i.e., all conflicts could be rationalized as conveying exceptions or ambiguities. Touretzky [27] has shown that this is indeed the case in the domain of acyclic and purely defeasible inheritance networks. However, once the language becomes more expressive, allowing hard rules as well as arbitrary formulae in the antecedents and consequents of the rules, the criterion for consistency becomes more involved. Consider the database

$$\begin{aligned} \Delta = \{&\text{"all birds fly", "typically, penguins are birds",} \\ &\text{"typically, penguins do not fly"}\}\,. \end{aligned}$$

This set of rules, although void of contradictory pairs, also strike us as

[1] The default rule $\mathsf{bird}(x) : \mathsf{M}\,\mathsf{fly}(x)/\mathsf{fly}(x)$ is informally interpreted as "If x is a bird and it is consistent to assume that x can fly, then infer that x can fly" (see [25]).

[2] A property f is ambiguous if neither f nor $\neg f$ can be verified from the database.

inconsistent: If all birds fly, there cannot be a nonempty class of objects (penguins) that are "typically birds" and yet "typically, do not fly". We cannot accept this database as merely depicting exceptions; it appears to be more of a programming "bug" than a genuine description of some state of affairs. If we now change the first sentence to read "typically, birds fly" (instead of "all birds fly"), consistency is restored; we are willing to accept penguins as exceptional birds. This interpretation will remain satisfactory even if we made the second rule strict (to read "all penguins are birds"). Yet, if we further add to Δ the sentence "typically, birds are penguins" we again face intuitive *inconsistency*.

In this paper we propose a probability-based formalism that captures these intuitions. We will interpret a *defeasible* sentence "typically, if ϕ then ψ" (written $\phi \rightarrow \psi$), as the conditional probability statement $P(\psi|\phi) \geq 1 - \varepsilon$, where $\varepsilon > 0$ is an infinitesimal quantity. Intuitively, this amounts to according the consequence ψ a very high likelihood whenever the antecedent ϕ is all that we know. The *strict* sentence "if φ then definitely σ" (written $\varphi \Rightarrow \sigma$), will be interpreted as an extreme conditional probability statement $P(\sigma|\varphi) = 1$. Our criterion for testing consistency translates to that of determining if there exists a probability distribution P that satisfies all these conditional probabilities for every $\varepsilon > 0$. Furthermore, to match our intuition that conditional sentences do not refer to empty classes, nor are they confirmed by merely "falsifying" their antecedents, we also require that P be *proper*, i.e., that it does not render any antecedent as totally impossible. These two requirements constitute the essence of our proposal.

Translated to the language of ranked models (see [16]), our proposal assumes a particularly simple form. A defeasible sentence $\phi \rightarrow \psi$ imposes the constraints that ψ holds in all minimally-ranked models of ϕ and that there will be at least one such model. A strict sentence $\varphi \Rightarrow \sigma$ imposes the constraint that no model satisfies $\varphi \wedge \neg\sigma$ and that at least one satisfies φ. Consistency amounts to requiring the existence of a ranking (mapping of models to integers) that simultaneously satisfies all these constraints (see [13]).

The idea of attaching probabilistic semantics to conditional sentences goes back to Adams [1, 2] who developed a logic of indicative conditionals based on infinitesimal probabilities. More recently, infinitesimal probabilities were mentioned in [19] as a possible interpretation of circumscription, and were used in [20] to develop a graphical consistency test for inheritance networks, extending that of Touretzky [27]. The proposals in [8, 10, 21] have extended Adams' logic to default schemata, and Lehmann and Magidor [17] have shown the equivalence between Adams' logic and a semantics based on ranked models.[3]

Unfortunately, the notion of consistency treated in [2, 20] was restricted to systems involving purely defeasible sentences. This paper extends Adams'

[3] This equivalence invites another argument in support of our consistency norm: It ensures that the database does not violate, explicitly or implicitly, any of the rules of cumulative (and preferential) reasoning [15]. A formal treatment of infinitesimal probabilities using nonstandard analysis is given in [17], and also mentioned in [26].

consistency results to mixed systems, containing both defeasible and strict information and, as we shall see, the extension is by no means trivial, since a strict sentence $b \Rightarrow f$ must be given a totally different semantics than its material counterpart $b \supset f$. For example, whereas the set of sentences $\{b \supset f, b \supset \neg f\}$ is logically consistent, our semantics must now render the set $\{b \Rightarrow f, b \Rightarrow \neg f\}$ inconsistent.[4]

In addition to extending the consistency criterion to include mixed systems, this paper also presents an effective syntactic procedure for testing this criterion and identifying the set of sentences responsible for the inconsistency. Finally, the paper analyzes a notion of entailment based on consistency considerations. Intuitively, a conclusion is entailed by a database if it is guaranteed an arbitrarily high probability whenever the premises are assigned sufficiently high probabilities. This weak notion of entailment was named p-entailment by Adams [2], ε-entailment by Pearl [21] and preferential entailment by Kraus et al. [15], and it yields (semimonotonically) the most conservative "core" of plausible conclusions that one would wish to draw from a conditional database [22].

The definition for probabilistic entailment can be partially extended to databases containing strict information using a device suggested by Adams [1] where, by definition, conditional sentences whose antecedents have probability zero are assigned probability one. Thus, one could conceivably encode a strict sentence like $\varphi \Rightarrow \sigma$ as the defeasible sentence

$$(\varphi \wedge \neg\sigma) \rightarrow \mathit{False}\ .$$

A more natural proposal was made in the preferential models analysis of Kraus et al. [15]. In their words [15, p. 172]:

> We reserve to ourselves the right to consider universes of reference that are strict subsets of the sets of all models of L. In this way we shall be able to model *strict* constraints such as, *penguins are birds*, in a simple and natural way, by restricting $\mathcal{U}$ to the set of all worlds that satisfy the material implication *penguin* $\supset$ *bird*.

These two proposals suffer from the following weaknesses: First, they do not capture the common understanding that the opposing pair "all birds fly" and "all birds don't fly" is inconsistent, but permit instead the conclusion that birds do not exist, together with other strange consequences such as "typically birds have property P" where P stands for any imaginable property. Our semantics reflects the view, also expressed by Delgrande [5], that one of the previous sentences must be invalid, and that no admissible model should support both

[4] The need to distinguish between $b \Rightarrow f$ and $b \supset f$ is further advocated in [5, 8, 10, 24], where the former is used to express generic knowledge and the latter as an item of evidence. This issue will be further discussed in Section 7.

sentences. Second, these two proposals do not permit us to entail new strict sentences in a more meaningful way than logical deduction. For example, $\neg a$ should not entail $a \Rightarrow b$, in the same way that "I am poor" should not entail "if I were rich, this paper would be accepted". Thus, the special semantics we give to conditional sentences, defeasible as well as strict, avoids such paradoxes of material implication [3] and, hence, it brings mechanical and plausible reasoning closer together.

The paper is organized as follows: Section 2 introduces notation and some preliminary definitions. Consistency and entailment are explored in Section 3. An effective procedure for testing consistency and entailment is presented in Section 4, while Section 5 contains illustrative examples. Section 6 deals with entailment in inconsistent databases and in Section 7 we summarize the main results of the paper. All proofs are given in the appendix.

2. Notation and preliminary definitions

Let $\mathscr{L}$ be a closed set of well-formed propositional formulas, built in the usual way from a *finite* set of propositional variables and the connectives "$\vee$" and "$\neg$" (the other connectives will be used as syntactic abbreviations). Lower case Greek letters ϕ, ψ, φ, σ will stand for formulas of $\mathscr{L}$, and lower case letters from the ordinary alphabet (except d, s and x) will stand for propositional variables.

Let ϕ and ψ be two formulas in $\mathscr{L}$. We will use a new binary connective "$\rightarrow$" to construct a defeasible sentence $\phi \rightarrow \psi$, which may be interpreted as "if ϕ then typically ψ". $\mathscr{D}$ will denote the set of all defeasible sentences, and D will denote a particular set of such sentences. Similarly, given φ and σ in $\mathscr{L}$, the binary connective "$\Rightarrow$" will be used to form a strict sentence $\varphi \Rightarrow \sigma$, which is to be interpreted as "if φ then definitely σ".[5] We will denote the set of all strict sentences by $\mathscr{S}$, and a particular set of such sentences will be denoted by S. Both "$\rightarrow$" and "$\Rightarrow$" can occur only as the main connective. We will use $\mathscr{K}$ to stand for the union of $\mathscr{D}$ and $\mathscr{S}$ (X for the union of some set D and some set S), and x, d, s as variables for sentences in $\mathscr{K}$, $\mathscr{D}$ and $\mathscr{S}$ respectively. We will use the term *conditional* when talking about a sentence that can be either defeasible or strict. If x denotes a conditional sentence with antecedent ϕ and consequent ψ, then the *negation* of x, denoted by $\sim x$, is defined as a conditional with antecedent ϕ and consequent $\neg\psi$. Finally, the *material counterpart* of a conditional sentence with antecedent ϕ and consequent ψ is defined as the formula $\phi \supset \psi$ (where "$\supset$" denotes material implication), and

[5] In the domain of nonmonotonic multiple inheritance networks, the interpretation for the defeasible sentence $\phi \rightarrow \psi$ would be "typically ϕ's are ψ's", while the interpretation for the strict sentence $\varphi \Rightarrow \sigma$ would be "all φ's are σ's".

the material counterpart of a set X of conditional sentences (denoted by $\hat{X}$), is defined as the conjunction of the material counterparts of the sentences in X.

A model M is an assignment of truth values to the propositional variables in $\mathscr{L}$. If there are n propositional variables in $\mathscr{L}$, there will be 2^n different models (or truth assignments) for $\mathscr{L}$. Let $\mathscr{M}$ denote the set of all possible models for $\mathscr{L}$. The satisfaction of a formula ϕ by a model M is defined as usual, and will be written as $M \models \phi$. We say that a sentence $x \in \mathscr{X}$ with antecedent ϕ and consequent ψ is *verified* by M, if $M \models \phi \wedge \psi$. x is *falsified* by M, if $M \models \phi \wedge \neg\psi$. Finally, x is considered as *satisfied* by M, if $M \models \phi \supset \psi$ (M satisfies the material counterpart of x).

Definition 2.1 (*Probability assignment*). Let P be a probability function on models, such that $P(M) \geq 0$ and $\sum_{M \in \mathscr{M}} P(M) = 1$. We define a probability assignment P on a formula $\phi \in \mathscr{L}$ as:

$$P(\phi) = \sum_{M \models \phi} P(M) . \tag{3}$$

A probability assignment on a defeasible sentence $\phi \rightarrow \psi \in \mathscr{D}$ is defined as:

$$P(\phi \rightarrow \psi) = \begin{cases} \dfrac{P(\phi \wedge \psi)}{P(\phi)} = P(\psi|\phi) , & \text{if } P(\phi) > 0 , \quad (4a) \\ 1 , & \text{if } P(\phi) = 0 . \quad (4b) \end{cases}$$

We assign probabilities to the sentences in $\mathscr{S}$ in exactly the same fashion. P will be considered *proper* for a conditional x, if $P(\phi) \neq 0$, and it will be proper for a set $X = D \cup S$ if it is proper for every conditional in X.

The probability assignment above attaches a conditional probability interpretation to the sentences in $\mathscr{X}$. Equation (4a) states that the probability of a conditional sentence x with antecedent ϕ and consequent ψ is equal to the probability of x being verified (i.e. $M \models \phi \wedge \psi$), divided by the probability of its being either verified or falsified (i.e. $M \models \phi$).

Up to this point the only difference between defeasible sentences and strict sentences was syntactic. They were assigned probabilities in the same fashion and were verified and falsified under the same truth assignments. Their differences will become clear in the next section, where we formally introduce the notion of *consistency*.

3. Probabilistic consistency and entailment

In all theorems and definitions below, we will consider the language $\mathscr{L}$ as fixed, and d', s', x' will stand for new defeasible, strict and conditional sentences respectively, with antecedents and consequents in $\mathscr{L}$.

Definition 3.1 (*Probabilistic consistency*). Let D and S be sets of defeasible and strict sentences respectively. We say that $X = D \cup S$ is *probabilistically consistent* (p-consistent) if, for every $\varepsilon > 0$, there is a probability assignment P, which is proper for X, such that $P(\psi|\phi) \geq 1 - \varepsilon$ for all defeasible sentences $\phi \rightarrow \psi$ in D, and $P(\sigma|\varphi) = 1$ for all strict sentences $\varphi \Rightarrow \sigma$ in S.

Intuitively, consistency means that it is possible for all defeasible sentences to become as close to certainty as desired, while all strict sentences hold with absolute certainty. Another way of formulating consistency is as follows: consider a constant $\varepsilon > 0$ and let $\mathscr{P}_{X,\varepsilon}$ stand for the set of proper probability assignments for X such that if $P \in \mathscr{P}_{X,\varepsilon}$ then $P(\psi|\phi) \geq 1 - \varepsilon$ for every $\phi \rightarrow \psi \in D$, and $P(\sigma|\varphi) = 1$ for every $\varphi \Rightarrow \sigma \in S$. Consistency insists on $\mathscr{P}_{X,\varepsilon}$ being nonempty for every $\varepsilon > 0$.

Before developing a syntactical test for consistency (Theorem 3.3), we need to define the concept of *toleration*:

Definition 3.2 (*Toleration*). Let x be a sentence with antecedent ϕ and consequent ψ. We say that x is *tolerated* by a set X, if there exists a model M such that M satisfies the formula $\phi \wedge \psi \wedge \hat{X}$.[6]

Thus, x is tolerated by a set of conditional sentences X, if there is a model M which verifies x and satisfies every sentence in X (i.e., no sentence in X is falsified by M).

Theorem 3.3. *Let $X = D \cup S$ be a nonempty set of defeasible and strict sentences. X is p-consistent iff every nonempty subset $X' = D' \cup S'$ of X complies with one of the following*:

(1) *If D' is not empty, then there must be at least one defeasible sentence in D' tolerated by X'.*

(2) *If D' is empty (i.e., $X' = S'$), each strict sentence in S' must be tolerated by S'.*

The following corollary ensures that, in order to determine p-consistency, it is not necessary to literally check every nonempty subset of X.

Corollary 3.4. *$X = D \cup S$ is p-consistent iff we can build an ordered partition of $D = [D_1, D_2, \ldots, D_n]$ where*:

(1) *for all $1 \leq i \leq n$, each sentence in D_i is tolerated by $S \cup \bigcup_{j=i+1}^{j=n} D_j$,*

(2) *every sentence in S is tolerated by S.*

Corollary 3.4 reflects the following considerations (see proof in Appendix

[6] Recall that $\hat{X}$ denotes the conjunction of the material counterparts of the sentences in X.

A). If X is p-consistent, Theorem 3.3 ensures the construction of the ordered partition. Conversely, if this partition can be built, the proof of Theorem 3.3 shows that a probability assignment can be constructed to comply with the requirements of Definition 3.1. Corollary 3.4 yields a simple and effective decision procedure for determining p-consistency and identifying the inconsistent subset in X (see Section 4).

Before turning our attention to the task of entailing new sentences, we need to make explicit a particular form of inconsistency:

Definition 3.5 (*Substantive inconsistency*). Let X be a p-consistent set of conditional sentences, and let x' be a conditional sentence with antecedent ϕ. We will say that x' is *substantively inconsistent* with respect to X, if $X \cup \{\phi \rightarrow True\}$ is p-consistent but $X \cup \{x'\}$ is p-inconsistent.

Nonsubstantive inconsistency occurs whenever the antecedent of a conditional sentence is logically incompatible with the strict sentences of a consistent set X. It will become apparent from the theorems to follow, that a sentence x is nonsubstantively inconsistent with respect to a consistent X, iff both $X \cup \{x\}$ and $X \cup \{\sim x\}$ are inconsistent.

The concept of *entailment* introduced below is based on the same probabilistic interpretation as the one used in the definition of p-consistency. Intuitively, we want p-entailed conclusions to receive arbitrarily high probability in every proper probability distribution in which the defeasible premises have sufficiently high probability, and in which the strict premises have probability equal to one.

Definition 3.6 (*p-entailment*). Given a p-consistent set X of conditional sentences, X p-entails $\phi' \rightarrow \psi'$ (written $X \models_{\mathrm{p}} \phi' \rightarrow \psi'$) if for all $\varepsilon > 0$ there exists $\delta > 0$ such that:

(1) there exists at least one $P \in \mathscr{P}_{X,\delta}$[7] such that P is proper for $\phi' \rightarrow \psi'$;

(2) every $P' \in \mathscr{P}_{X,\delta}$ satisfies $P'(\psi' | \phi') \geq 1 - \varepsilon$.

Theorem 3.7 relates the notions of entailment and consistency:

Theorem 3.7. *If X is p-consistent, X p-entails $\phi' \rightarrow \psi'$ iff $\phi' \rightarrow \neg\psi'$ is substantively inconsistent with respect to X.*

Definition 3.8 and Theorem 3.9 below characterize the conditions under which conditional conclusions are guaranteed not only very high likelihood but also absolute certainty. We call this form of entailment *strict p-entailment*:

[7] Recall that given a consistent $X = D \cup S$, $\mathscr{P}_{X,\delta}$ stands for the set of probability assignments proper for X, such that if $P \in \mathscr{P}_{X,\varepsilon}$ then $P(\psi|\phi) \geq 1 - \delta$ for every $\phi \rightarrow \psi \in D$, and $P(\sigma|\varphi) = 1$ for every $\varphi \Rightarrow \sigma \in S$ (see Definition 3.1).

Definition 3.8 (*strict p-entailment*). Given a p-consistent set X of conditional sentences, X strictly p-entails $\varphi' \Rightarrow \sigma'$ (written $X \models_s \varphi' \Rightarrow \sigma'$) if for all $\varepsilon > 0$:

(1) there exists at least one $P \in \mathscr{P}_{X,\varepsilon}$ such that P is proper for $\varphi' \Rightarrow \sigma'$;

(2) every $P' \in \mathscr{P}_{X,\varepsilon}$ satisfies $P'(\sigma' | \varphi') = 1$.

Theorem 3.9. *If $X = D \cup S$ is p-consistent, X strictly p-entails $\varphi' \Rightarrow \sigma'$ iff $S \cup \{\varphi' \rightarrow True\}$ is p-consistent and there exists a subset S' of S such that $\varphi' \Rightarrow \neg\sigma'$ is not tolerated by S'.*

Examples of strict p-entailment are contraposition,

$$\{\phi \Rightarrow \psi\} \models_s \neg\psi \Rightarrow \neg\phi,^{8}$$

and chaining,

$$\{\phi \Rightarrow \sigma, \sigma \Rightarrow \psi\} \models_s \phi \Rightarrow \psi .$$

Note that strict p-entailment subsumes p-entailment, i.e., if a conditional sentence is strictly p-entailed then it is also p-entailed. Also, to test whether a conditional sentence is strictly p-entailed we need to check its status only with respect to the strict set in X. This confirms the intuition that we cannot deduce "hard" rules from "soft" ones. However, strict p-entailment is different from logical entailment because the requirements of substantive consistency and properness for the probability distributions distinguishes strict sentences from their material counterpart. For example, consider the database $X = S = \{c \Rightarrow \neg a\}$ which is clearly p-consistent. While $\hat{X}$ logically entails $c \wedge a \supset b$, X does not strictly p-entail $c \wedge a \Rightarrow b$, since the antecedent $c \wedge a$ is always falsified.

For completeness, we now present two additional theorems relating consistency and entailment. Similar versions of these theorems, for the case of purely defeasible sentences, first appeared in [2]. They follow from previous theorems and definitions.

Theorem 3.10. *If X does not p-entail $\phi' \rightarrow \psi'$, and $\phi' \rightarrow \psi'$ is substantively inconsistent with respect to X, then for all $\varepsilon > 0$ there exists a probability assignment $P' \in \mathscr{P}_{X,\varepsilon}$ which is proper for X and $\phi' \rightarrow \psi'$ such that $P'(\psi' | \phi') \leqslant \varepsilon$.*

Theorem 3.11. *If $X = D \cup S$ is p-consistent, then*

- *it cannot be the case that both $\phi \rightarrow \psi$ and $\phi \rightarrow \neg\psi$ are substantively inconsistent with respect to X;*
- *it cannot be the case that both $\varphi \Rightarrow \sigma$ and $\varphi \Rightarrow \neg\sigma$ are substantively inconsistent with respect to S.*

[8] Whenever $\neg\psi$ is satisfiable.

4. An effective procedure for testing consistency

In accordance with Theorem 3.3 and following Corollary 3.4, the consistency of a database $X = D \cup S$ can be tested in two phases: In the first phase, until D is empty, we repeatedly remove a sentence from D that is tolerated by the rest of the sentences in $D \cup S$. In the second phase we must test whether every sentence in S is tolerated by the rest of S (without removing any sentence). If both phases can be successfully completed X is consistent, else X is inconsistent.

PROCEDURE TEST_CONSISTENCY
INPUT: a set $X = D \cup S$ of
defeasible and strict sentences
1. LET $D' := D$
2. WHILE D' is not empty DO
3. Find a sentence $d : \phi \rightarrow \psi \in D'$ such that d is tolerated by $S \cup D'$
4. IF d is found then
LET $D' := D' - \{d\}$
ELSE HALT: X is INCONSISTENT
ENDWHILE
5. LET $S' := S$
6. WHILE S' is not empty DO
7. Pick any sentence $s : \varphi \Rightarrow \sigma \in S'$ and test if s is tolerated by S
8. IF s is tolerated then
LET $S' := S' - \{s\}$
9. ELSE HALT: X is INCONSISTENT
ENDWHILE
10. X is CONSISTENT
END PROCEDURE

The same procedure can be used for entailment, since to determine whether a defeasible sentence d' is entailed by X we need only test the consistency of $X \cup \{\sim d'\}$ and $X \cup \{d'\}$ (to make sure that the former is substantively inconsistent). Given that the above procedure is correct, the next theorem establishes an upper bound for the complexity of deciding p-consistency (and p-entailment). Theorem 4.1 and the correctness of the procedure TEST_CONSISTENCY are proven in the appendix.

Theorem 4.1. *The worst-case complexity of testing consistency (or entailment) is bounded by* $[\mathcal{PS} \times (\frac{1}{2}|D|^2 + |S|)]$ *where* $|D|$ *and* $|S|$ *are the number of defeasible and strict sentences respectively, and* $\mathcal{PS}$ *is the complexity of propositional satisfiability for the material counterpart of the sentences in the database.*

Thus, the complexity of deciding p-consistency and p-entailment is no worse than that of propositional satisfiability. Although the general satisfiability problem is NP-complete, useful sublanguages (e.g. Horn clauses) are known to admit polynomial algorithms [6].

The order in which sentences are removed in procedure TEST_CONSISTENCY induces natural priorities among defaults that were used to great advantage in several proposals for default reasoning [9, 11, 13, 23]. These priorities have an alternative epistemic interpretation in the theory of belief revision described by Gärdenfors [7]. The fact that a conditional $\varphi \rightarrow \psi$ is tolerated by all those sentences that were not previously removed from X means that if φ holds, then ψ can be asserted without violating any sentence in X that is more deeply entrenched than this conditional. In other words, adding the assertion $\phi \wedge \psi$ would require a minimal revision of the set of beliefs supported by X. The formal relation between the default priorities used in system Z [23] and the postulates for epistemic entrenchment in believe revision [7] is studied in [4]. The origin of this priority ordering can be traced back to Adams [2], where it is used to build "nested sequences" of confirmable subsets of X yielding consistent high probability models. Such "nested sequences" are used in the proof of Theorem 3.3 (see Appendix A). A similar construction was also used in [16, Theorem 5] to prove the co-NP-completeness of p-entailment in purely defeasible databases.[9]

Once a set of sentences is found to be p-inconsistent, it would be useful to identify the sentences that are *directly responsible* for the contradiction. Unfortunately, the toleration relation is not strong enough to accomplish this task since it is incapable of distinguishing a sentence "causing" the inconsistency, from one that is a "victim" of the inconsistency. For example, consider the inconsistent set

$$D_i = \{\phi \rightarrow \psi, \phi \rightarrow \neg\psi, \phi \rightarrow \sigma\} .$$

Since no sentence in D_i is tolerated, the consistency test will immediately halt and declare D_i inconsistent. Yet $\phi \rightarrow \sigma$ can hardly be held responsible for the inconsistency; the reason $\phi \rightarrow \sigma$ is not tolerated is due to the pair $\{\phi \rightarrow \psi, \phi \rightarrow \neg\psi\}$, whose material counterpart renders ϕ impossible.[10] It would be inappropriate to treat a sentence as the source of inconsistency merely because it is not tolerated in the context of an inconsistent subset. Rather, we would like to proclaim a sentence *inconsistent* if its removal would improve the consistency of the database. In other words, a conditional sentence x is inconsistent with respect to a set X if and only if there is an inconsistent subset of X that becomes consistent after x is removed. Formally:

[9] This was pointed out to us by an anonymous reviewer.

[10] Note that $\{\phi \supset \psi, \phi \supset \neg\psi\} \models \neg\phi$.

Definition 4.2 (*inconsistent sentence*). A sentence x is *inconsistent with respect to a set* X iff there exists a subset X' of X such that $X' \cup \{x\}$ is *p-inconsistent*, but X' in itself is *p-consistent*.

The problem of deciding whether a given sentence is inconsistent is a tough one because, unlike the test for set inconsistency, the search for the indicative subset X' cannot be systematized as in procedure TEST_CONSISTENCY. All indications are that the search for such a subset will require exponential time. Simple minded procedures based on removing one sentence at a time and testing for consistency in the remaining set do not yield the desired results. In

$$X' = \{a \rightarrow b,\ a \rightarrow \neg b,\ a \rightarrow c,\ a \Rightarrow \neg c\}$$

every sentence is inconsistent, however it is necessary to remove at least two sentences at a time in order to render the remaining set consistent. Likewise, in

$$X'' = \{a \rightarrow b,\ a \rightarrow \neg b,\ a \rightarrow c,\ c \Rightarrow \neg b\}$$

every sentence is inconsistent, yet only the removal of $a \rightarrow b$ renders the remaining set consistent. Approximate methods for identifying inconsistent sentences are discussed in Section 6 and in the proof of Theorem 6.10 (see the appendix).

5. Examples

Example 5.1. On *birds* and *penguins*. We begin by testing the consistency[11] of the database presented in the introduction:

(1) $b \Rightarrow f$ ("all birds fly"),

(2) $p \rightarrow b$ ("typically, penguins are birds"),

(3) $p \rightarrow \neg f$ ("typically, penguins don't fly").

Clearly none of the defeasible sentences in the example can be tolerated by the rest. Consider a model M, such that $M \models p \wedge b$ (testing whether sentence (2) is tolerated); if $M \models f$, sentence (3) will be falsified, while if $M \models \neg f$, sentence (1) will be falsified. Thus, we conclude that there is no model such that sentence (2) is tolerated. A similar situation arises when we check if sentence (3) can be tolerated. Changing sentence (1) to be defeasible yields the familiar "penguin triangle"

$$D_p = \{b \rightarrow f,\ p \rightarrow b,\ p \rightarrow \neg f\}$$

[11] The terms *consistency* and *p-consistency* will be used interchangeably.

which is consistent: i.e., $b \rightarrow f$ is tolerated by sentences (2) and (3) through the model M', where $M' \models b \wedge f$ and $M' \models \neg p$, and once sentence (1) is removed, the remaining sentences *tolerate* each other. D_p becomes inconsistent by adding the sentence $b \rightarrow p$ ("typically, birds are penguins"), in conformity to the graphical criterion of Pearl [20, 21]. Note that by Theorem 3.7, the sentence $b \rightarrow \neg p$ ("typically, birds are not penguins") is then p-entailed by D_p. To demonstrate an inconsistency that cannot be detected by such graphical criteria, consider adding to D_p the sentence $p \wedge b \rightarrow f$. Again no sentence will be tolerated and the set will be proclaimed inconsistent, thus showing (by Theorem 3.7) that $p \wedge b \rightarrow \neg f$ is p-entailed by D_p as expected ("typically, penguin-birds don't fly"). Interestingly, all these conclusions remain valid upon changing sentence (2) into a strict conditional $p \Rightarrow b$ (which is the usual way of representing the penguin triangle), showing that strict class subsumption is not really necessary for facilitating specificity-based preferences in this example.

Example 5.2. On *quakers* and *republicans*. Consider a database containing the following set of sentences:

(1) $n \rightarrow r$ ("typically, Nixonites[12] are republicans"),
(2) $n \rightarrow q$ ("typically, Nixonites are quakers"),
(3) $q \Rightarrow p$ ("all quakers are pacifists"),
(4) $r \Rightarrow \neg p$ ("all republicans are non-pacifists"),
(5) $p \rightarrow c$ ("typically, pacifists are persecuted").

Sentence (5) is tolerated by all others, but (1) is not tolerated by (2)–(4), nor is (2) tolerated by $\{(1), (3), (4)\}$. Hence, the database is inconsistent. The following modification renders the database consistent:

(1) $n \Rightarrow r$ ("all Nixonites are republicans"),
(2) $n \Rightarrow q$ ("all Nixonites are quakers"),
(3) $q \rightarrow p$ ("typically, quakers are pacifists"),
(4) $r \rightarrow \neg p$ ("typically, republicans are non-pacifists"),
(5) $p \rightarrow c$ ("typically, pacifists are persecuted").

Indeed, there is a basic conceptual difference between the former case and this one. If all quakers are pacifists and all republicans are non-pacifists, our intuition immediately reacts against the idea of finding an individual that is both a quaker and a republican. The modified database, on the other hand, allows a "Nixonite" that is both a quaker and a republican to be either pacifist or non-pacifist. Note that both $n \rightarrow p$ and $n \rightarrow \neg p$ are consistent when added to the database so neither one is p-entailed, and we can assert that the conclusion

[12] "Nixonites" is a fictitious name for people that share Richard Nixon's cultural background.

is *ambiguous* (i.e., we cannot decide whether a "Nixonite" is typically a "pacifist" or not).

Finally, if we make sentences (2) and (4) be the only strict rules, we get a database similar in *structure* to the example depicted by network Γ_6 in [14]:

(1) $n \rightarrow r$ ("typically, Nixonites are republicans"),
(2) $n \Rightarrow q$ ("all Nixonites are quakers"),
(3) $q \rightarrow p$ ("typically, quakers are pacifists"),
(4) $r \Rightarrow \neg p$ ("all republicans are non-pacifists"),
(5) $p \rightarrow c$ ("typically, pacifists are persecuted").

Not surprisingly, the criterion of Theorem 3.3 renders this database consistent and $n \rightarrow \neg p$ is p-entailed in conformity with the intuition expressed in [14].

6. Reasoning with inconsistent databases

The theory developed in previous sections presents desirable features both from the semantics and computational standpoints. However, the entailment procedure insists on starting with a consistent set of conditional sentences. In this section we plan to relax this requirement and explore two proposals for making entailment insensitive to contradictory statements in unrelated portions of the database, so that mistakes in the encoding of properties about penguins and birds would not tamper with our ability to reason about politics (e.g. quakers and republicans). The first proposal amounts to accepting local inconsistencies as deliberate albeit strange expressions, while the second treats them as programming "bugs".

In Definition 2.1 a conditional sentence $\phi \rightarrow \psi$ was assigned the conditional probability $P(\psi|\phi)$ if P was *proper* for $\phi \rightarrow \psi$ (i.e., if $P(\phi) > 0$). In our first proposal for reasoning with inconsistent databases, we will regard improper probability assignments as admissible, and define $P(\psi|\phi) = 1$ whenever $P(\phi) = 0$.[13] With this approach any set X of conditional sentences[14] can be represented by the trivial high probability distribution in which some antecedents receive zero probability. Also, strict sentences like $\varphi \Rightarrow \sigma$ can be represented as $\varphi \wedge \neg\sigma \rightarrow \mathit{False}$, since we can now use $P(\varphi \wedge \neg\sigma) = 0$ to get $P(\sigma|\varphi) = 1$. As before, we say that a sentence $\phi \rightarrow \psi$ is *implied*[15] by a (possibly inconsistent) set X if $\phi \rightarrow \psi$ receives arbitrarily high probability in all probability assignments in which sentences in X receive arbitrarily high probability.

[13] Even though $P(\phi \rightarrow \psi) = 1$ if $P(\phi) = 0$ in Definition 2.1, $P(\phi \rightarrow \psi)$ was not related to a conditional probability in those cases.

[14] As long as $\hat{X}$ is satisfiable. If $\hat{X}$ is not satisfiable this proposal cannot do better than propositional logic: any conditional sentence will be trivially entailed.

[15] We will use the term "implication" instead of "entailment" to stress the fact that the set of premises may constitute an inconsistent set. We will however keep the "$\models$" symbol for simplicity.

Definition 6.1 (*p_1-implication*). Given a set X of conditional sentences and a conditional sentence $\phi' \rightarrow \psi'$, X p_1-implies $\phi' \rightarrow \psi'$, written $X \models_{p_1} \phi' \rightarrow \psi'$, if for all $\varepsilon > 0$ there exists a $\delta > 0$ such that for all probability assignments P, if $P(\psi|\phi) \geq 1 - \delta$ for all $\phi \rightarrow \psi \in X$, and $P(\sigma|\varphi) = 1$ for all $\varphi \Rightarrow \sigma \in X$ then $P(\psi'|\phi') \geq 1 - \varepsilon$.

The only difference between Definition 6.1 and that of p-entailment (Definition 3.6) is that none of the probability assignments in the definition above are constrained to be proper.

Any inconsistent set X will have a nonempty subset violating one of the conditions of Theorem 3.3. Given that almost all properties stated in this section will refer to such sets, we find it convenient to introduce the following definition:

Definition 6.2 (*Unconfirmable sets*). A set $X = D \cup S$ is said to be *unconfirmable* if one of the following conditions is true:

(1) If D is nonempty, then there cannot be a defeasible sentence in D that is tolerated by X.

(2) If D is empty (i.e., $X = S$) then there must be a strict sentence in S which is not tolerated by X.

Note that a set X_u can be unconfirmable, while both a superset of X_u or one of its subsets can be confirmable. The problem of deciding whether a sentence is p_1-implied is no worse than that of deciding p-entailment as shown by the next theorem proven in [12]:

Theorem 6.3. *A set of conditional sentences X p_1-implies $\phi \rightarrow \psi$ iff $\phi \rightarrow \neg\psi$ belongs to an unconfirmable subset of $X \cup \{\phi \rightarrow \neg\psi\}$.*

This unconfirmable subset can be identified using the consistency test discussed in Section 4, and it follows that p_1-implication also requires a polynomial number of satisfiability tests. Moreover, p_1-entailment is equivalent to p_1-implication if the set X is consistent (see Theorem 6.9 below). As an example, consider the union of

$$D_p = \{b \rightarrow f,\ p \rightarrow b,\ p \rightarrow \neg f\}$$

(encoding the so-called penguin triangle), and the inconsistent set

$$D_i = \{\phi \rightarrow \psi,\ \phi \rightarrow \neg\psi,\ \phi \rightarrow \sigma\}\ .$$

Some of the sentences p_1-implied by $X_i = D_p \cup D_i$ are: $p \wedge b \rightarrow \neg f$ (typically, penguin-birds don't fly), $b \rightarrow \neg p$ (typically, birds are not penguins), and $\phi \rightarrow \sigma$. Some of the sentences *not* p_1-implied by X_i are $p \wedge b \rightarrow f$ and $p \rightarrow \psi$. Thus, despite its inconsistency, not all sentences are p_1-implied by X_i. How-

ever, this example also demonstrates a disturbing feature of p_1-implication; not only are $\phi \rightarrow \psi$ and $\phi \rightarrow \neg\psi$ p_1-implied, but also $\phi \rightarrow \neg\sigma$ and $\phi \rightarrow p$. Thus, although the natural properties of penguins remain unperturbed by the inconsistency of D_i, strange sentences like $\phi \rightarrow p$ are deduced even though there is no argument to support them (see [14] for similar considerations on inconsistent sentences in the context of inheritance networks).

To reveal the source of this phenomenon, it is useful to declare a *formula* to be *inconsistent*, if the formula is *False* by default:

Definition 6.4 (*Inconsistent formulas*). Given a set X and a formula ϕ, we say that ϕ is an *inconsistent formula* with respect to X, iff X p_1-implies $\phi \rightarrow$ False.

The following theorem is an easy consequence of Theorem 6.3. It relates p_1-implication to Definition 6.4 above, and provides an alternative definition of inconsistent formulas in terms of propositional entailment:

Theorem 6.5. *Consider a set X of conditional sentences and the formulas σ and ψ:*

(1) $X \models_{p_1} \sigma \rightarrow \psi$ *iff* σ *is an* inconsistent formula *with respect to* $X \cup \{\sigma \rightarrow \neg\psi\}$.

(2) *If σ is an* inconsistent formula *with respect to X, any conditional sentence with σ as antecedent will be p_1-implied by X.*

(3) *A formula σ is inconsistent with respect to a set X iff there exists an* unconfirmable *subset X' of X such that $\hat{X}' \models \neg\phi$*[16] *where σ is the antecedent of a sentence in X'.*

Theorem 6.5(2) explains why a sentence like $\phi \rightarrow p$ is p_1-implied by X_i: ϕ is an inconsistent formula with respect to X_i, hence *any* sentence with ϕ as antecedent will be trivially p_1-implied by X_i.

This deficiency of p_1-implication is removed in our second proposal for reasoning with inconsistent databases which we call *p_2-implication*. The intuition behind p_2-implication is to consider a sentence as "implied" only if its negation would introduce a *new* inconsistency into the database. Previous inconsistencies are thus considered as "bugs" and are simply ignored:

Definition 6.6 (p_2-*implication*). Given a set X, we say that $\phi \rightarrow \psi$ is p_2-implied by X, written $X \models_{p_2} \phi \rightarrow \psi$, iff $\phi \rightarrow \psi$ is *not an inconsistent sentence* with respect to X (see Definition 4.2) but its negation $\phi \rightarrow \neg\psi$ is.

The requirement that not both $\phi \rightarrow \psi$ and $\phi \rightarrow \neg\psi$ be inconsistent serves two

[16] Recall that $\hat{X}$ denotes the conjunction of the material counterparts of the conditional sentences in X.

purposes: first, as with p-entailment, it constitutes a *safeguard* against sentences being trivially implied by virtue of their antecedent being false. Second, if both sentences were inconsistent, the contradiction that originates when either of them is added to X must have been previously embedded in X, and therefore cannot be *new*. In our previous example, the sentences

$$p \wedge b \rightarrow \neg f\,, \qquad b \rightarrow \neg p\,, \qquad \phi \rightarrow \sigma$$

are p_2-implied by X_i, however, contrary to p_1-implication the sentences

$$\phi \rightarrow \psi\,, \qquad \phi \rightarrow \neg\psi\,, \qquad \phi \rightarrow \neg\sigma\,, \qquad \phi \rightarrow p$$

are not. As stated in Theorem 6.9 below, p_2-implication is strictly stronger than p_1-implication and is equivalent to p-entailment if the set X is p-consistent.

Since the notion of p_2-implication is based on the concept of an inconsistent sentence (Definition 4.2) there is strong evidence that any procedure for deciding p_2-implication will be exponential (see Section 4). To obtain a more efficient decision procedure, we propose to weaken the definition of an inconsistent sentence. Instead of testing whether a given sentence is responsible for an inconsistency, we will test whether the sentence is responsible for creating an inconsistent *formula* (see Theorem 6.5(3) above).

Definition 6.7 (*Weakly inconsistent sentence*). A sentence x is *weakly inconsistent* with respect to a set X, iff there exists an unconfirmable subset X_u of $X \cup \{x\}$, such that $\hat{X}_u \models \neg\phi$ but $\hat{X}'_u \not\models \neg\phi$ where $X'_u = X_u - \{x\}$, and ϕ is the antecedent of some sentence in X_u.

This leads naturally to the notion of weak p_2-implication:

Definition 6.8 (*wp_2-implication*). Given a set X, a sentence $\phi \rightarrow \psi$ is wp_2-implied by X, written $X \models_{wp_2} \phi \rightarrow \psi$, iff $\phi \rightarrow \neg\psi$ is weakly inconsistent with respect to X.

Similar to both p_1- and p_2-implications, the set $X_i = D_p \cup D_i$ wp_2-implies the sentences $p \wedge b \rightarrow \neg f$, $b \rightarrow \neg p$, and $\phi \rightarrow \sigma$. More importantly, contrary to p_1-implication (but similar to p_2-implication) the undesirable sentences $\phi \rightarrow \neg\sigma$ and $\phi \rightarrow p$ are not wp_2-implied by X_i, and, in general, wp_2-implication will not sanction a sentence merely because its antecedent is inconsistent. However, unlike p_2-implication, wp_2-implication will sanction any sentence whose consequence is the negation of an inconsistent formula (for example $p \rightarrow \neg\phi$).

The notion of wp_2-implication is situated somewhere between p_1-implication and p_2-implication as the next couple of theorems indicate. It rests semantically on both, since it requires the concepts of inconsistent formulas and inconsistent

sentences, and preserves some of the computational advantages of p_1-implication.

Theorem 6.9.

(1) *Given a* p-consistent *set* X, *the notions of p-entailment,* p_1*-implication,* wp_2*-implication and* p_2*-implication are equivalent.*

(2) *Given a* p-inconsistent *set* X, p_2*-implication is strictly stronger than* wp_2*-implication, and* wp_2*-implication is strictly stronger than* p_1*-implication.*

Theorem 6.10.[17] *If the set* X *is acyclic and of* Horn *form,* wp_2*-implication can be decided in polynomial time.*

The reason wp_2-implication is *harder* than p_1-implication is the need to search for a suitable unconfirmable subset X_u (see Definition 6.8).

7. Discussion

We have formalized a norm of consistency for mixed sets of conditionals, ensuring that every group of sentences be satisfiable in a nontrivial way, one in which the antecedent and consequent of at least one sentence are both true. We showed that any group of sentences that is not satisfiable this way must contain conflicts that cannot be reconciled by appealing to exceptions or ambiguities, and are normally considered contradictory, i.e., unfit to represent world knowledge. Using this norm, an effective procedure was devised to test for inconsistencies, and a tight relation between entailment and consistency was established, permitting entailment to be decided using consistency tests. These tests were shown to require polynomial complexity relative to propositional satisfiability. We also discussed ways of drawing conclusions from inconsistent databases as well as uncovering sets of sentences directly responsible for such inconsistencies.

One of the key requirements in our definition of consistency is that no conditional sentence in X should have an impossible antecedent and, moreover, that no antecedent should become absolutely impossible as exceptions (to default sentences) become less likely (i.e., as ε becomes small). This requirement mirrors our understanding that it is fruitless to build databases for nonexisting classes, and counterintuitive to deduce (even defeasibly) conditional sentences having impossible antecedents. Consequently, pairs such as $\{\phi \rightarrow \psi, \phi \rightarrow \neg\psi\}$ or $\{\phi \Rightarrow \psi, \phi \Rightarrow \neg\psi\}$ are labeled inconsistent and treated as unintentional mistakes. The main application of the procedures proposed in

[17] The proof of this theorem can be found in the appendix.

this paper is to warn users and knowledge providers of such "bugs", lest they yield undesirable inferences.

This paper also presents a new formalization of strict conditional sentences within the analysis of probabilistic consistency, totally distinct from their material counterpart. The importance of this distinction has been recognized by several researchers (e.g., Poole [24], Delgrande [5], Geffner [8], Geffner and Pearl [10] and others) and has both theoretical and practical implications.

In ordinary discourse conditionals are recognized by universally quantified subsumptions such as "all penguins are birds" or, in case of ground sentences, by the use of the English word "*If*" (e.g., "If Tweety is a penguin then she is a bird"). The function of these *indicators* is to alert the listener that the assertion made is not based on evidence pertaining to the specific individual, but rather on generic background knowledge pertaining to the individual's class (e.g., being a penguin). It is this pointer to the background information that is lost if one encodes a conditional sentence as a Boolean expression, and it is this information that is crucial for adequately processing specificity preferences.

Intuitively, background knowledge encodes the general tendency of things to happen, i.e., relations that hold true in all worlds, while evidential knowledge describes that which actually happened, i.e., relations in our particular world. Thus, conditional sentences, both defeasible and indefeasible, play a role similar to that of meta-inference rules: they tell us how to draw conclusions from specific observations about a particular situation or a particular individual, but do not themselves convey such observations. It is for this reason that we chose to use a separate connective "$\Rightarrow$" to denote strict conditionals, as is done by Horty and Thomason [14] in the context of inheritance networks. Strict conditionals, by virtue of pointing to generic background knowledge, are treated as part of the database, while propositional formulas, including material implications, are used only to formulate queries, but are excluded from the database itself. By so doing, the sentence $p \Rightarrow b$ is treated as a constraint over the set of admissible probability assignments, while the propositional formulae $p \supset b$ is treated as a specific evidence or observation, on which these probability assignments are to be conditioned.

It does indeed make a profound difference whether our knowledge of *Tweety*'s birdness comes from generic background knowledge about penguins, or from specific observations conducted on *Tweety*. In natural language, the latter case would normally be phrased by nonconditional sentences such as "it is not true that *Tweety* is both a penguin and a nonbird", which is equivalent to the material implication

$$penguin(Tweety) \supset bird(Tweety)\,.$$

The practical aspects of this distinction can best be demonstrated using the penguin example.[18] Assume we know that "typically, birds fly" and "typically,

[18] Taken from [10].

penguins do not fly". If we are told that Tweety is a penguin, and that "all penguins are birds", we would like to conclude that Tweety does not fly. By the same token if we are told that Tweety is a bird and that "all birds are penguins" we would have to conclude that Tweety does fly. However, note that both $\{p, p \supset b\}$ and $\{b, b \supset p\}$ are logically equivalent to $\{p, b\}$, which totally ignores the relation between penguins and birds, and should yield identical conclusions regardless of whether penguins are a subclass of birds or the other way around. Thus, when treated as material implications, information about class subsumption is permitted to combine with properties attributed to individuals and this crucial information gets lost.

This distinction was encoded in [10] by placing strict conditionals together with defaults in a "background context", separate from the "evidential set" which was reserved for observations made on a particular state of affairs. In [16] it is stated that "dealing with hard constraints, in addition to soft ones, involves relativizing to some given set of tautologies". Here, again, strict conditionals would receive different treatment than ground formulas; only the former are permitted to influence rankings among worlds. We believe that the separate connective "$\Rightarrow$" used in our treatment makes this distinction clear and natural, and the uniform probabilistic semantics given to both strict and defeasible sentences adequately captures the notion of consistency in systems containing such mixtures.

The notion of p-entailment is known to yield a rather conservative set of conclusions (e.g., one that does not permit chaining or contraposition), we therefore do not propose p-entailment as a complete characterization of defeasible reasoning. It nevertheless yields a core of plausible consequences that should be maintained in every system that reasons defeasibly. Extensions of p-entailment can be found in [9, 11, 16]. All these formalisms, as well as circumscription (McCarthy [19]), default logic (Reiter [25]) and argument-based systems (Loui [18], Horty and Thomason [14]) could benefit from a preliminary test of consistency such as the one proposed in this paper.

Appendix A. Theorems and proofs

Since some of the proofs below refer to *unconfirmable* sets, we recall their definition:

Definition 6.2 (*Unconfirmable sets*). A set $X = D \cup S$ is said to be *unconfirmable* if one of the following conditions is true:

(1) If D is nonempty, then there cannot be a defeasible sentence in D that is tolerated by X.

(2) If D is empty (i.e., $X = S$) then there must be a strict sentence in S which is not tolerated by X.

Essentially, unconfirmable sets are those that violate the conditions of Theorem 3.3.

Theorem 3.3. *Let $X = D \cup S$ be a nonempty set of defeasible and strict sentences. X is p-consistent iff every nonempty subset $X' = D' \cup S'$ of X complies with one of the following:*

(1) *If D' is not empty, then there must be at least one defeasible sentence in D' tolerated by X'.*

(2) *If D' is empty (i.e., $X' = S'$), each strict sentence in S' must be tolerated by S'.*

Proof. We first prove the only-if part. We want to show that if there exists a nonempty subset of X which is unconfirmable, then X is not p-consistent. The proof is facilitated by introducing the notion of *quasi-conjunction* (see [2]): Given a set of defeasible sentences

$$D = \{\phi_1 \rightarrow \psi_1, \ldots, \phi_n \rightarrow \psi_n\}$$

the *quasi-conjunction* of D is the defeasible sentence,

$$C(D) = [\phi_1 \vee \cdots \vee \phi_n] \rightarrow [(\phi_1 \supset \psi_1) \wedge \cdots \wedge (\phi_n \supset \psi_n)] . \qquad \text{(A.1)}$$

The quasi-conjunction $C(D)$ bears interesting relations to the set D. In particular, if there is a defeasible sentence in D which is tolerated (by D) by some model M, $C(D)$ will be verified by M. This is so because the verification of at least one sentence of D by M guarantees that the antecedent of $C(D)$ (i.e., the formula $[\phi_1 \vee \cdots \vee \phi_n]$ in equation (A.1) is satisfied by M, and the fact that no sentence in D is falsified guarantees that the consequent of $C(D)$ (i.e., the formula $[(\phi_1 \supset \psi_1) \wedge \cdots \wedge (\phi_n \supset \psi_n)]$ in equation (A.1) is also satisfied by M. Similarly, if at least one sentence of D is falsified by a model M', its quasi-conjunction is also falsified by M' since in this case, the consequent of $C(D)$ is not satisfied by M' (at least one of the material implications in the conjunction is falsified by M'). Additionally, let

$$U_p(C(D)) = 1 - P(C(D))$$

(the *uncertainty* of $C(D)$) where $P(C(D))$ is the probability assigned to the quasi-conjunction of D according to equation (4a), then, it is shown in [1] that the uncertainty of the quasi-conjunction of D is less or equal to the sum of the uncertainties of each of the sentences in D, i.e.,

$$U_p(C(D)) \leqslant \sum_i (1 - P(\psi_i|\phi_i)) ,$$

where the sum is taken over all $\phi_i \rightarrow \psi_i$ in D.

We are now ready to proceed with the proof. Let $X' = D' \cup S'$ be a nonempty subset of X where D' is a subset of D and S' is a subset of S. If X' is unconfirmable, then one of the following cases must occur:

Case 1: *S' is empty and D' is unconfirmable.*[19] In this case, the quasi-conjunction for D' is not verifiable; from equation (4a), we have that for any P which is proper for $C(D')$, $P(C(D'))=0$ and $U_p(C(D'))=1$. It follows, by the properties of the quasi-conjunction outlined above that $\sum_i (1-P(\psi_1'|\phi_i'))$ over all $\phi_i' \rightarrow \psi_i'$ in D' is at least 1. If the number of sentences in D' is $n \geqslant 1$, then,

$$n - \sum_{i=1}^{n} P(\psi_i'|\phi_i') \geqslant 1 , \tag{A.2}$$

$$\sum_{i=1}^{n} P(\psi_i'|\phi_i') \leqslant n-1 , \tag{A.3}$$

which implies that at least one sentence in D' has probability smaller than $1-1/n$. Hence, it is impossible to have $P(\psi_i'|\phi_i') \geqslant 1-\varepsilon$, for every $\varepsilon > 0$, for every defeasible sentence $\phi_i' \rightarrow \psi_i' \in D'$. Thus, X is p-inconsistent.

Case 2: *D' is empty.* If S' is unconfirmable, then there must be at least one sentence $\varphi' \Rightarrow \sigma' \in S'$ such that no model M' verifies $\varphi' \Rightarrow \sigma'$ without falsifying another sentence in S'. We show by contradiction that there is no probability assignment P to the sentences in S' such that $P(\sigma|\varphi)=1$ for all $\varphi \Rightarrow \sigma \in S'$ and P is proper for every sentence in S'. Assume there exists such a P. From equation (4a)

$$P(\sigma|\varphi) = \frac{\sum_{M \models \varphi \wedge \sigma} P(M)}{\sum_{M \models \varphi \wedge \sigma} P(M) + \sum_{M \models \varphi \wedge \neg\sigma} P(M)} = 1 , \tag{A.4}$$

which immediately implies that if a model M'' falsifies any sentence $\varphi'' \Rightarrow \sigma'' \in S'$ (including $\varphi' \Rightarrow \sigma'$), then $P(M'')$ must be zero, else $P(\sigma''|\varphi'')$ will not equal 1. Thus, $P(M')=0$ for every M' verifying $\varphi' \Rightarrow \sigma'$ since M' must falsify another sentence in S'. But then either $P(\sigma'|\varphi')=0$, or P is not proper for $\varphi' \Rightarrow \sigma'$: A contradiction. We conclude that if S' is unconfirmable then X is p-inconsistent.

Case 3: *Neither D' nor S' are empty and X' is unconfirmable.* That is, either the quasi-conjunction $C(D')$ is not verifiable or every M' that verifies a defeasible sentence in D' falsifies at least one sentence in S'. The first situation will lead us back to Case 1 while the second leads to a contradiction similar to Case 2 above. In either case, X is not p-consistent.

We now prove the if part. Assume that every nonempty subset of $X = D \cup S$ complies with the conditions of Theorem 3.3. Then the following two constructions are feasible:

- We can construct a finite "nested decreasing sequence" of nonempty subsets of X, namely $X_1, \ldots, X_m$ ($X = X_1$), and an associated sequence of truth assignments $M_1, \ldots, M_m$ such that M_i satisfies all the sentences in X_i

[19] This case is covered by [2, Theorem 1.1].

and verifies at least one defeasible sentence in X_i, and the sets in the sequence present the following characteristics:

(1) X_{i+1} is the proper subset of X_i consisting of all the sentences of D_i not verified by M_i, for $i = 1, \ldots, m-1$, plus the sentences in S;

(2) all sentences in D_m are verified by M_m.

- We can construct a sequence $M_{m+1}, \ldots, M_n$ that will *confirm* $X_{m+1} = S$. That is, the sequence $M_{m+1}, \ldots, M_n$ will verify every sentence in S without falsifying any. We will associate with $M_{m+1}, \ldots, M_n$ the "nested decreasing sequence" $X_{m+1}, \ldots, X_n$ where X_{i+1} is the proper subset of X_i consisting of all the sentences of S_i not verified by M_i for $i = m+1, \ldots, n$.

We can now assign probabilities to the truth assignments $M_1, \ldots, M_n$ in the following way: For $i = 1, \ldots, n-1$,

$$P(M_i) = \varepsilon^{i-1}(1-\varepsilon) \tag{A.5}$$

and

$$P(M_n) = \varepsilon^{n-1}. \tag{A.6}$$

We must show that, in fact, every $\phi \rightarrow \psi$ in D obtains $P(\psi|\phi) \geqslant 1-\varepsilon$ and that every $\varphi \Rightarrow \sigma$ in S obtains $P(s) = 1$. Since every $\phi \rightarrow \psi$ is verified in at least one of the members of the sequence $X_1, \ldots, X_n$, using equations (4a), (A.5) and (A.6) we have that for $i < n$:

$$P(\psi_i|\phi_i) \geqslant \frac{\varepsilon^{i-1}(1-\varepsilon)}{\varepsilon^{i-1}(1-\varepsilon) + \varepsilon^{i-2}(1-\varepsilon) + \cdots + \varepsilon^{n-1}} = 1-\varepsilon \tag{A.7}$$

and $P(\psi_n|\phi_n) = 1$ if it is only verified by the last model when S is originally empty. Finally, since no $\varphi \Rightarrow \sigma$ in S is ever falsified by the sequence of truth assignments $M_1, \ldots, M_n$ and each and every $\varphi \Rightarrow \sigma$ is verified at least once, it follows from equation (4a) and the process by which we assigned probabilities to $M_1, \ldots, M_n$ that indeed $P(\sigma|\varphi) = 1$ for every $\varphi \Rightarrow \sigma \in S$. □

Corollary 3.4. *$X = D \cup S$ is p-consistent iff we can build an ordered partition of $D = [D_1, D_2, \ldots, D_n]$ where:*

(1) *for all $1 \leqslant i \leqslant n$, each sentence in D_i is tolerated by $S \cup \bigcup_{j=i+1}^{j=n} D_j$,*

(2) *every sentence in S is tolerated by S.*

Proof. If X is p-consistent, by Theorem 3.3 we must be able to find a tolerated defeasible sentence in every subset $X' = D' \cup S'$ (of X) where D' is nonempty, and it follows that the construction of the ordered partition $D = [D_1, D_2, \ldots, D_n]$ is possible. Similarly, by Theorem 3.3, if X is p-consistent every strict sentence in S must be tolerated by S. On the other hand, if both conditions in the corollary hold, we use the set of models (M_i) that renders the sentences in each D_i tolerated by the set $S \cup \bigcup_{j=i+1}^{j=n} D_j$ to construct a high

probability model for X, following the probability assignments of equations (A.5) and (A.6). □

Theorem 3.7. *If X is p-consistent, X p-entails $\phi' \rightarrow \psi'$ iff $\phi' \rightarrow \neg\psi'$ is substantively inconsistent with respect to X.*

Proof. We first prove the only-if part. (If X p-entails $\phi' \rightarrow \psi'$, then $\phi' \rightarrow \neg\psi'$ is substantively inconsistent with respect to X.) Let $X \models_{\mathrm{p}} \phi' \rightarrow \psi'$. From the definition of p-entailment (Definition 3.6), for all $\varepsilon > 0$ there exists a $\delta > 0$ such that for all $P \in \mathscr{P}_{X,\delta}$ which are proper for X and $\phi' \rightarrow \psi'$, $P(\neg\psi' | \phi') \leqslant \varepsilon$. This means that for all proper probability assignments P for X and $\phi' \rightarrow \psi'$,[20] the sentence $\phi' \rightarrow \neg\psi'$ gets an arbitrarily low probability whenever all defeasible sentences in X can be assigned arbitrarily high probability and all strict sentences in X can be assigned probability equal to 1. Thus $\phi' \rightarrow \neg\psi'$ is substantively inconsistent with respect to X.

We now prove the if part. (If $\phi' \rightarrow \neg\psi'$ is substantively inconsistent with respect to X, then X p-entails $\phi' \rightarrow \psi'$.) Let $\phi' \rightarrow \neg\psi'$ be substantively inconsistent with respect to X. From Theorem 3.3, we know that there must be a subset X' of $X \cup \{\phi' \rightarrow \neg\psi'\}$ that is unconfirmable. Furthermore, since X is p-consistent, $X' = X'' \cup \{\phi' \rightarrow \neg\psi'\}$. Let $\mathscr{P}_S$ stand for the set of probability distributions that are proper for X and $\phi' \rightarrow \neg\psi'$ such that if $P \in \mathscr{P}_S$, then $P(\sigma|\varphi) = 1$ for all $\varphi \Rightarrow \sigma$ in X.[21] We will consider two cases depending on the structure of X'':

Case 1: *X'' does not include any defeasible sentences.* From Theorem 3.3, we know that $\phi' \rightarrow \neg\psi'$ cannot be tolerated by X'' for otherwise X' wouldn't be inconsistent. It follows from equation (4a) (probability assignment) that $P(\neg\psi' | \phi') = 0$ for all $P \in \mathscr{P}_S$. Thus, $P(\psi' | \phi') = 1$ in all $P \in \mathscr{P}_S$ and since any probability distribution that is in $\mathscr{P}_{X,\varepsilon}$ must also belong to $\mathscr{P}_S$, it follows from the definition of p-entailment that $X \models_{\mathrm{p}} \phi' \rightarrow \psi'$.

Case 2: *X'' includes defeasible and a possible empty set of strict sentences.* Since $X'' \cup \{\phi' \rightarrow \neg\psi'\}$ is unconfirmable, we have from the proof of Theorem 3.3, that for all probability distributions $P \in \mathscr{P}_S$:

$$\sum_{\phi \rightarrow \psi \in X''} U_p(\phi \rightarrow \psi) + U_p(\phi' \rightarrow \neg\psi') \geqslant 1 , \tag{A.8}$$

which implies that

$$\sum_{\phi \rightarrow \psi \in X} U_p(\phi \rightarrow \psi) \geqslant 1 - U_p(\phi' \rightarrow \neg\psi') = U_p(\phi' \rightarrow \psi') . \tag{A.9}$$

[20] Note that from the definition of p-entailment there must exist at least one P proper for X and $\phi' \rightarrow \psi'$.

[21] We know that $\mathscr{P}_S$ is not empty since $X \cup \{\phi' \rightarrow \mathit{True}\}$ must be p-consistent according to Definition 3.5. In the case where X does not contain any strict sentences, $\mathscr{P}_S$ simply denotes all probability distributions that are proper for $X \cup \{\phi' \rightarrow \mathit{True}\}$.

Since $U_p(\phi \rightarrow \psi) = 1 - P(\phi \rightarrow \psi)$ and $U_p(\phi' \rightarrow \psi') = 1 - P(\phi' \rightarrow \psi')$, equation (A.9) says that $1 - P(\phi' \rightarrow \psi')$ can be made arbitrarily small by requiring the values $1 - P(\phi \rightarrow \psi)$ for $\phi \rightarrow \psi \in D$ to be sufficiently small and the values of $P(\sigma|\varphi)$ to be 1 for all $\varphi \Rightarrow \sigma \in S$. This is equivalent to saying that $X \models_p \phi' \rightarrow \psi'$. □

Theorem 3.9. *If $X = D \cup S$ is p-consistent, X strictly p-entails $\varphi' \Rightarrow \sigma'$ iff $S \cup \{\varphi' \rightarrow True\}$ is p-consistent and there exists a subset S' of S such that $\varphi' \Rightarrow \neg\sigma$ is not tolerated by S'.*

Proof. It follows from the proof of Theorem 3.7 (see Case 1 of the *if* part). □

Lemma A.1. *TEST_CONSISTENCY constitutes a decision procedure for testing the p-consistency of a set X of conditional sentences.*

Proof. If the procedure stops at either line 4 or line 9 an unconfirmable subset is found, and by Theorem 3.3 the set of sentences is p-inconsistent. If, on the other hand, the procedure reaches line 10, the order in which the sentences are tolerated can be used to build a high probability model for X using the construction (of the "nested decreasing sequence") in the proof of Theorem 3.3, and X must therefore be p-consistent. □

Theorem 4.1. *The worst-case complexity of testing consistency (or entailment) is bounded by $[\mathcal{PS} \times (\frac{1}{2}|D|^2 + |S|)]$ where $|D|$ and $|S|$ are the number of defeasible and strict sentences respectively, and $\mathcal{PS}$ is the complexity of propositional satisfiability for the material counterpart of the sentences in the database.*

Proof. Given that TEST_CONSISTENCY constitutes a decision procedure for p-consistency (see Lemma A.1 above), a complexity bound for this procedure will be an upper bound for the problem of deciding p-consistency. To assess the time complexity of TEST_CONSISTENCY, note that the WHILE-loop of line 6 will be executed $|S|$ times in the worst case, and each time we must do at most $\mathcal{PS}$ work to test the satisfiability of $S - \{s\}$; thus, its complexity is $|S| \times \mathcal{PS}$. In order to *find* a tolerated sentence $d : \phi \rightarrow \psi$ in D', we must test at most $|D'|$ times (once for each sentence $d \in D'$) for the satisfiability of the conjunction of $\phi \wedge \psi$ and the material counterparts of the sentences in $S \cup D' - \{d\}$. However, the size of D' is decremented by at least one sentence in each iteration of the WHILE-loop in line 2, therefore the number of times that we test for satisfiability is

$$|D| + |D| - 1 + |D| - 2 + \cdots + 1$$

which is bounded by $\frac{1}{2}|D|^2$. Thus, the overall time complexity is $O[\mathcal{PS} \times (\frac{1}{2}|D|^2 + |S|)]$. □

Theorem 6.10. *If the set X is acyclic and of* Horn *form, wp_2-implication can be decided in polynomial time.*

Proof. The proof of this theorem requires a short review of some results from [6], since the procedure for deciding wp_2-implication is based on one of the algorithms presented in that paper. Given a set $\mathcal{H}$ of Horn clauses, Dowling and Gallier define an auxiliary graph $G_{\mathcal{H}}$ to represent the set $\mathcal{H}$, and reduce the problem of finding a truth assignment satisfying the sentences in $\mathcal{H}$ to that of finding a *pebbling* on the graph using a breadth-first strategy. We first describe these concepts more precisely and then apply them to the problem at hand:

Definition A.2 (Dowling and Gallier [6]). Given a set $\mathcal{H}$ of Horn clauses, $G_{\mathcal{H}}$ is labeled directed graph with $N+2$ nodes (a node for each propositional letter occurring in $\mathcal{H}$, a node for **true** and a node for **false**) and a set of labels $[I]$. It is constructed with i taking values in $[I]$ as follows depending on the form of the ith Horn formula in $\mathcal{H}$:

(1) If it is a positive literal q, there is an edge from **true** to q labeled i.
(2) If it is of the form $\neg p_1 \vee \cdots \vee \neg p_n$, there are n edges from $p_1, \ldots, p_n$ to **false** labeled i.
(3) If it is of the form $\neg p_1 \vee \cdots \vee \neg p_n \vee q$, there are n edges from $p_1, \ldots, p_n$ to q labeled i.

A node q in $G_{\mathcal{H}}$ can be *pebbled* if and only if for some label i, all sources of incoming edges labeled i are pebbled. The node **true** is considered to be pebbled. A pebbled path is a path on the graph such that all its nodes are pebbled. Given the correspondence between a Horn rule h_i and the set of i-labeled edges in the graph we are going to use both terms (edge and rule) indistinctively. Thus, eliminating a rule h_i should be understood as removing the set of i-labeled edges from the graph. Similarly a pebbled rule will indicate that the associated nodes in the graph are pebbled etc. A graph $G_{\mathcal{H}}$ is considered to be *completely* pebbled, if and only if all nodes that remain unpebbled have at least one incoming edge with a source that cannot be pebbled; i.e., there cannot be a pebbled path from **true** to that node.

Lemma A.3 (Dowling and Gallier [6]). *Let $\mathcal{H}$ be a set of Horn clauses and let $G_{\mathcal{H}}$ be its associated graph, $\mathcal{H}$ is unsatisfiable iff there is a pebbling in $G_{\mathcal{H}}$ from* **true** *to* **false**.

This lemma and the existence of an $O(N^2)$ algorithm for deciding satisfiability are proven in [6] (N represents the number of occurrences of literals in the set of clauses). We now prove a couple of lemmas regarding a polynomial procedure for deciding whether a conditional sentence x is weakly inconsistent with respect to a set X. Recall that by the definition of wp_2-implication

(Definition 6.8) once we have identified a sentence as weakly inconsistent, its negation is wp_2-implied. The lemma below shows a simple test for deciding whether a particular Horn sentence h is essential for the unsatisfiability of some set $\mathcal{H}$:

Lemma A.4. *Let $G_{\mathcal{H}}$ be an acyclic graph representing the set $\mathcal{H}$ of Horn clauses. Assume that $\mathcal{H}$ is unsatisfiable and that $G_{\mathcal{H}}$ is completely pebbled. Let $h \in \mathcal{H}$ be a Horn clause such that both the antecedent and consequent of h are pebbled in $G_{\mathcal{H}}$, and assume that there is a pebbled path from the consequent of h to* **false**. *Then there exists a nonempty subgraph $G'_{\mathcal{H}}$ of $G_{\mathcal{H}}$ containing h such that $G'_{\mathcal{H}}$ is unsatisfiable but $G'_{\mathcal{H}} - \{h\}$ is satisfiable.*

We show the correctness of this lemma by constructing the graph $G'_{\mathcal{H}}$. The idea is to eliminate from $G_{\mathcal{H}}$ all the alternative pebbled paths to **false**, and leave $G'_{\mathcal{H}}$ with only the path that goes through the rule h, together with those necessary to render this path pebbled. First, we select one pebbled path from **true** to **false** that goes through h (by the assumptions of Lemma A.4, we know that there is at least one). Next, we eliminate any rule that reaches **false** directly (i.e., of form (2) in Definition A.2) that is not in the selected path. We now traverse the selected path "backwards" from **false** to the node representing the consequent of h, and remove any incoming edges that are not necessary to render this path pebbled. Note that we can guarantee to have eliminated alternative paths to **false**. The only possibility for this construction to fail is if we would have removed some paths that pebble the antecedents of h (in which case $G'_{\mathcal{H}}$ would be satisfiable), but this can only happen if there is a cycle in the graph involving h, and this possibility is ruled out by the assumptions of acyclicity. Since to complete the pebbling of a graph is no worse than testing for satisfiability, and searching for a pebbled path from one node to another can also be done by a breadth-first search algorithm it follows that the test outlined in Lemma A.4 can be performed in time polynomial in N. This test constitutes the basis of a procedure for deciding weakly inconsistency:

Lemma A.5. *Given a set X which is of Horn form and acyclic, to decide whether a sentence is weakly inconsistent with respect to X requires polynomial time.*

Given a set X and a sentence x, we first apply the consistency test of Section 4 to $X \cup \{x\}$ in order to find an unconfirmable subset X_u. If none can be found or the sentence x does not belong to X_u, we can assert that x is *not* weakly inconsistent with respect to X. In the first case X is consistent, and in the second case x does not belong to any inconsistent subset of $X \cup \{x\}$. Once X_u is found (and $x \in X_u$), we systematically complete the pebbling of the associated graph G_{X_u} starting from each one of the antecedents of the sentences in

X_u. If in one of these pebblings, the sentence x complies with the requirements of the test outlined in Lemma A.4, then x is weakly inconsistent. Note that all the steps involved require polynomial time with respect to N (i.e., the number of occurrences of literals in the set of clauses), and since once we have a procedure for deciding whether a sentence is weakly inconsistent we have a procedure for wp_2-implication (see Definition 6.8), we have essentially proven Theorem 6.10. □

We remark that these results are relevant not only to nonmonotonic reasoning but to any application involving propositional entailment.

Acknowledgement

Many of the proofs, techniques and notation are extensions of those presented in [2]. We thank E. Adams, P. Eggert, H. Geffner, J. Horty, K. Konolige, D. Lehmann, M. Magidor, D. Makinson, P. Morris, and two anonymous reviewers for useful discussions and comments. We are indebted to Kurt Konolige for pointing out a mistake in an earlier draft of Section 6, and to Charles Elkan for suggesting the term "tolerate".

References

[1] E.W. Adams, Probability and the logic of conditionals, in: J. Hintikka and P. Suppes, eds., *Aspects of Inductive Logic* (North-Holland, Amsterdam, 1966).
[2] E.W. Adams, *The Logic of Conditionals* (Reidel, Dordrecht, Netherlands, 1975).
[3] A. Anderson and N. Belnap, *Entailment: The Logic of Relevance and Necessity, Vol.* 1 (Princeton University Press, Princeton, NJ, 1975).
[4] C. Boutilier, Default priorities as epistemic entrenchment, Tech. Report KRR-TR-91-2, University of Toronto, Toronto, Ont. (1991).
[5] J.P. Delgrande, An approach to default reasoning based on a first-order conditional logic: revised report, *Artif. Intell.* **36** (1988) 63–90.
[6] W. Dowling and J. Gallier, Linear-time algorithms for testing the satisfiability of propositional horn formulae, *J. Logic Program.* **3** (1984) 267–284.
[7] P. Gärdenfors, *Knowledge in Flux: Modeling the Dynamics of Epistemic States* (MIT Press, Cambridge, MA, 1988).
[8] H.A. Geffner, On the logic of defaults, in: *Proceedings AAAI-88*, St. Paul, MN (1988) 449–454.
[9] H.A. Geffner, Default reasoning: causal and conditional theories, Tech. Report TR-137, Ph.D. Dissertation, Cognitive Systems Lab., University of California Los Angeles, Los Angeles, CA (1989). Forthcoming, MIT Press (1992).
[10] H.A. Geffner and J. Pearl, A framework for reasoning with defaults, in: H.E. Kyburg, R.P. Loui and G. Carlson, eds., *Knowledge Representation and Defeasible Reasoning* (Kluwer Academic Publishers, London, 1990) 245–265.
[11] M. Goldszmidt, P. Morris and J. Pearl, A maximum entropy approach to nonmonotonic reasoning, in: *Proceedings AAAI-90*, Boston, MA (1990) 646–652.
[12] M. Goldszmidt and J. Pearl, On the relation between rational closure and system Z, in:

Proceedings Third International Workshop on Nonmonotonic Reasoning, South Lake Tahoe, CA (1990) 130–140.

[13] M. Goldszmidt and J. Pearl, System Z^+: a formalism for reasoning with variable strength defaults, in: *Proceedings AAAI-91*, Anaheim, CA (1991) 399–404.

[14] J.F. Horty and R.H. Thomason, Mixing strict and defeasible inheritance, in: *Proceedings AAAI-88*, St. Paul, MN (1988) 427–432.

[15] S. Kraus, D. Lehmann and M. Magidor, Nonmonotonic reasoning, preferential models and cumulative logics, *Artif. Intell.* **44** (1990) 167–207.

[16] D. Lehmann, What does a conditional knowledge base entail? in: *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Ont. (1989) 212–222.

[17] D. Lehmann and M. Magidor, What does a conditional knowledge base entail? Tech. Report TR-90-10, Department of Computer Science, Hebrew University, Jerusalem (1990).

[18] R.P. Loui, Defeat among arguments: a system of defeasible inference, *Comput. Intell.* **3** (3) (1987) 100–106.

[19] J. McCarthy, Applications of circumscription to formalizing common-sense knowledge, *Artif. Intell.* **28** (1986) 89–116.

[20] J. Pearl, Deciding consistency in inheritance networks, Tech. Report TR-96, Cognitive Systems Lab., University of California Los Angeles, Los Angeles, CA (1987).

[21] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann, San Mateo, CA, 1988).

[22] J. Pearl, Probabilistic semantics for nonmonotonic reasoning: a survey, in: *Proceedings First International Conference on Principles of Knowledge Representation and Reasoning*, Toronto, Ont. (1989) 505–516.

[23] J. Pearl, System Z: a natural ordering of defaults with tractable applications to default reasoning, in: M. Vardi, ed., *Proceedings of Theoretical Aspects of Reasoning about Knowledge* (Morgan Kaufmann, San Mateo, CA, 1990) 121–135.

[24] D. Poole, On the comparison of theories: preferring the most specific explanation, in: *Proceedings IJCAI-85*, Los Angeles, CA (1985) 144–147.

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

[26] W. Spohn, Ordinal conditional functions: a dynamic theory of epistemic states, in: W.L. Harper and B. Skyrms, eds., *Causation in Decision, Belief Change, and Statistics* (Reidel, Dordrecht, Netherlands, 1987) 105–134.

[27] D.S. Touretzky, *The Mathematics of Inheritance Systems* (Morgan Kaufmann, San Mateo, CA, 1986).