Paraconsistent Inference from Data using Existential Ω-Entailment

Michael Kassoff and Michael R. Genesereth
Logic Group, Stanford University
{*mkassoff, genesereth*}*@cs.stanford.edu*

***Abstract*—Existential Ω-entailment is a paraconsistent entailment relation designed to show the consequences of data which is inconsistent with a set of integrity constraints Ω. In this paper, we prove semantic properties of existential Ω-entailment and give an algorithm for computing it.**

***Keywords*-deductive databases; logic**

I. Introduction

Existential Ω-entailment (sometimes referred to as "strict entailment") is a paraconsistent entailment relation that was designed specifically for the purpose of reasoning with data that is inconsistent with a set of constraints. Existential Ω-entailment was first used in logical spreadsheets to show the consequences of the data even when it was inconsistent with the spreadsheet formulae [1]. It has also been used to propagate values in Web forms that allow form values to be inconsistent with validation constraints [2], and to query databases in which the data is inconsistent with the integrity constraints [3]. In the case of Web forms and spreadsheets, temporary inconsistencies are occasionally required to move from one consistent state to another, as only one value may be changed at a time. In the case of databases, inconsistencies arise, for example, due to changing integrity constraints, or in databases consisting of data gathered from multiple conflicting sources.

While existential Ω-entailment was defined in [1], no algorithm was given for computing the entailment relation, and the naïve algorithm is intractable even for small amounts of data and constraints. While [3] gives an algorithm for reformulating existential Ω-entailment queries into Datalog queries, the reformulation can be expensive, and may not be worthwhile unless the cost is amortized. In this paper, we give the first proof theory for directly computing the existentially Ω-entailed facts for a set of data.

The rest of this paper is organized as follows. In Section II we define existential Ω-entailment. In Section III we give an illustrative example of its use. In Section IV we examine its semantic properties. In Section V we compare existential Ω-entailment to resolution. In Section VI we discuss how to compute the existentially Ω-entailed data from a set of logical constraints and a set of data. In Section VII we discuss related work.

II. Definitions

We assume a standard definition of first-order logic and a standard definition of resolution theorem proving [4].

Existential Ω-entailment is defined as follows:

Definition 1 *Let* Λ *and* Ω *be sets of sentences. We say that a sentence* ϕ *is* existentially Ω-entailed *by* Λ *iff there is some subset* $\lambda \subseteq \Lambda$ *such that* $\lambda \cup \Omega$ *is consistent and* $\lambda \cup \Omega$ *logically entails* ϕ*. We will use the notation* $\Lambda \mid\!\approx_\Omega \phi$ *to indicate that* Λ *existentially* Ω*-entails* ϕ*.*

Existential Ω-entailment is a generalization of existential entailment [5], [6], [7] to allow for a set of nondefeasible constraints Ω. A set of formulae Λ *existentially entails* a sentence ϕ iff there is a consistent subset $\lambda \subseteq \Lambda$ such that λ logically entails ϕ. Existential entailment corresponds to existential Ω-entailment where $\Omega = \emptyset$.

We can equivalently define existential Ω-entailment using *maximal* Ω*-consistent subsets of* Λ, which are those Ω-consistent subsets of Λ not strictly contained in any other Ω-consistent subset of Λ.

Proposition 1 (Maximality)**.** *Say that* $\Lambda \mid\!\approx_\Omega^M \phi$ *iff there is a* maximal Ω*-consistent subset* $\lambda \subseteq \Lambda$ *such that* $\lambda \cup \Omega$ *logically entails* ϕ*. Then* $\Lambda \mid\!\approx_\Omega \phi$ *iff* $\Lambda \mid\!\approx_\Omega^M \phi$*.*

Proof If $\Lambda \mid\!\approx_\Omega^M \phi$, then there is an Ω-consistent subset λ of Λ such that $\lambda \cup \Omega \models \phi$, and thus $\Lambda \mid\!\approx_\Omega \phi$. If $\Lambda \mid\!\approx_\Omega \phi$, then there is an Ω-consistent subset λ of Λ such that $\lambda \cup \Omega \models \phi$. Let λ' be a maximal Ω-consistent subset of Λ that contains λ. By the monotonicity of logical entailment, since $\lambda \cup \Omega \models \phi$ and $\lambda \subseteq \lambda'$ then $\lambda' \cup \Omega \models \phi$. Since is λ' is maximal, $\Lambda \mid\!\approx_\Omega^M \phi$. □

A *fact* is a ground literal. A *datum* is a positive fact. In this paper, we will be focusing on the application of existential Ω-entailment to databases. We will refer to the consequence relation *existential* Ω*-entailment restricted to data* defined by restricting the domain of existential Ω-entailment to data and the range of existential Ω-entailment to facts.

III. example

Traditional logical entailment is *explosive*, i.e. an inconsistent theory logically entails all sentences. In contrast, existential Ω-entailment is *paraconsistent*, meaning that it does not entail all conclusions when an inconsistency is present. Intuitively, existential Ω-entailment only entails conclusions that are justified by an Ω-consistent subset of Λ. We illustrate this concept with an example. Consider the following constraints Ω:

$\neg r(X) \vee \neg d(X)$	(One cannot be both Republican and Democrat.)
$r(X) \Rightarrow g(X)$	(Republicans like George.)
$d(X) \Rightarrow b(X)$	(Democrats like Bill.)
$b(X) \wedge g(X) \Rightarrow u(X)$	(People who like Bill and George are undecided.)

And data $\Lambda = \{r(Robert), d(David), r(Ed), d(Ed)\}$

Then the maximal Ω-consistent sets of Λ are:

s_1: $\{r(Robert), d(David), r(Ed)\}$
s_2: $\{r(Robert), d(David), d(Ed)\}$

The data entailed by $s_1 \cup \Omega$ is:

$$c_1 = \{r(Robert), d(David), r(Ed), g(Robert), b(David), g(Ed)\}$$

The data entailed by $s_2 \cup \Omega$ is:

$$c_2 = \{r(Robert), d(David), r(Ed), g(Robert), b(David), b(Ed)\}$$

And so the existentially Ω-entailed data are:

$$c_1 \cup c_2 = \{r(Robert), d(David), r(Ed), g(Robert), b(David), g(Ed), d(Ed)\}$$

Note that, although both $g(Ed)$ and $d(Ed)$ are existentially Ω-entailed, $u(Ed)$ is not, due to the constraint $\neg r(X) \vee \neg d(X)$.

IV. Properties

To help shed light on the behavior of Existential Ω-entailment, in this section we investigate its formal properties. We first note that existential Ω-entailment depends on the syntax of the theory it is applied to, not just the semantics. To make this notion formal, we first need the following definitions.

Definition 1 (Logical Equivalence)**.** *Two theories Φ and Ψ are* logically equivalent *iff for all $\phi \in \Phi$, $\Psi \models \phi$ and conversely for all $\psi \in \Psi$, $\Phi \models \psi$.*

Definition 2 (Syntax Dependence)**.** *A consequence relation $\mid\!\approx$ is* syntax independent *if, for any logically equivalent theories Φ and Ψ and any sentence γ, $\Phi \mid\!\approx \gamma$ iff $\Psi \mid\!\approx \gamma$. Otherwise it is* syntax-dependent.

Now we are ready to prove our assertion.

Proposition 2 (Syntax Dependence)**.** *Existential Ω-entailment is syntax dependent.*

Proof Let $\Phi = \{p(a) \wedge q(a)\}$, let $\Psi = \{p(a), q(a)\}$ and let $\Omega = \{\neg q(a)\}$. Then Φ and Ψ are logically equivalent, but $\Psi \mid\!\approx_\Omega p(a)$ and $\Phi \not\mid\!\approx_\Omega p(a)$. □

This is discouraging. However, in practice, we will only be using Existential Ω-entailment applied to data.

Proposition 3 (Syntax Independence)**.** *Existential Ω-entailment restricted to data is syntax independent.*

Proof This trivially arises from the fact that two data are logically equivalent iff they have the same data. □

Also importantly, the logical form of the constraints does not matter.

Proposition 4 (Equivalence)**.** *Let Ω and Θ be two logically equivalent theories. Then for any theory Λ and sentence ϕ, $\Lambda \mid\!\approx_\Omega \phi$ iff $\Lambda \mid\!\approx_\Theta \phi$.*

Proof Say that $\Lambda \mid\!\approx_\Omega \phi$. Then there is an Ω-consistent subset λ of Λ such that $\lambda \cup \Omega \models \phi$. Since Ω and Θ are logically equivalent, $\lambda \cup \Theta \models \phi$, and thus $\Lambda \mid\!\approx_\Theta \phi$. The other case is symmetric. □

We now turn towards examining properties of existential Ω-entailment. First we define some notation to denote a generic consequence relation.

Definition 3 (Consequences)**.** *Let Λ be a theory and let $\mid\!\approx$ be a consequence relation.*
$Cn_{\mid\!\approx}(\Lambda) =_{def} \{\phi \mid \Lambda \mid\!\approx \phi\}$

Table 1 lists properties (and non-properties) of existential Ω-entailment. The list of properties studied is adapted from [7] and [8]. Proofs of the properties can be found in [7], [9].

Table 1 shows that existential Ω-entailment behaves quite differently than classical logical entailment, for which all of the properties hold.[1] For the properties that do not hold, they do not hold even when existential Ω-entailment is restricted to data.

V. Comparison to Resolution

The need for a paraconsistent entailment relation to show the consequences of data arises due to the explosive nature of classical logic. However, it is worth noting that resolution can be used to construct a paraconsistent entailment relation. A *clause* is a set of literals and represents a disjunction of the literals in the set. A clause Φ *subsumes* a clause Ψ if and only if there exists a substitution σ such that $\Phi\sigma \subseteq \Psi$.

Definition 4 (Resolution Entailment)**.** *Let Ω be a set of clausal formulae, and let ϕ be a literal. We say that Ω-*resolution entails *ϕ (written $\Omega \mid\!\approx_R \phi$) iff ϕ is subsumed by a non-empty clause that has a resolution derivation from Ω.*

[1] Ω-Subclassicality and Ω-Consistent Classicality do not hold for logical entailment, though Subclassicality and Consistent Classicality do. Subclassicality is defined as $\Lambda \models \phi$ *if* $\Lambda \mid\!\approx \phi$ and Consistent Classicality is defined as *if* $\Lambda \not\models \bot$, $\Lambda \mid\!\approx \phi$ *iff* $\Lambda \models \phi$, where $\mid\!\approx$ is a consequence relation.

Property	Name	?
$\{\psi, \neg\psi\} \not\approx_\Omega \phi$	(Explosiveness)	N
$\Lambda \cup \Omega \models \phi$ if $\Lambda \not\approx_\Omega \phi$	(Ω-Subclassicality)	Y
if $\Lambda \cup \Omega \not\models \bot$, $\Lambda \not\approx_\Omega \phi$ iff $\Lambda \cup \Omega \models \phi$	(Ω-Consistent Classicality)	Y
$\Lambda \cup \{\phi\} \not\approx_\Omega \psi$ if $\Lambda \not\approx_\Omega \psi$	(Monotonicity)	Y
$\Lambda \not\approx_\Omega \phi$ if $\phi \in \Lambda$	(Reflexivity)	N
$\Lambda \cup \{\phi\} \not\approx_\Omega \gamma$ if $\Lambda \cup \{\psi\} \not\approx_\Omega \gamma$ and $\models \phi \Leftrightarrow \psi$	(Left Logical equivalence)	Y
$\Lambda \not\approx_\Omega \phi$ if $\Lambda \not\approx_\Omega \psi$ and $\not\approx_\Omega \psi \Rightarrow \phi$	(Right Weakening)	Y
$\Lambda \cup \{\neg\psi\} \not\approx_\Omega \neg\phi$ if $\Lambda \cup \{\phi\} \not\approx_\Omega \psi$	(Contraposition)	N
$\Lambda \not\approx_\Omega \phi \wedge \psi$ if $\Lambda \not\approx_\Omega \phi$ and $\Lambda \not\approx_\Omega \psi$	(And)	N
$\Lambda \cup \{\phi \vee \psi\} \not\approx_\Omega \gamma$ if $\Lambda \cup \{\phi\} \not\approx_\Omega \gamma$ and $\Lambda \cup \{\psi\} \not\approx_\Omega \gamma$	(Or)	N
$\Lambda \not\approx_\Omega \psi$ if $\Lambda \not\approx_\Omega \phi$ and $\Lambda \cup \{\phi\} \not\approx_\Omega \psi$	(Cut)	N
$\Lambda \not\approx_\Omega \phi \Rightarrow \psi$ if $\Lambda \cup \{\phi\} \not\approx_\Omega \psi$	(Conditionalization)	Y
$\Lambda \cup \{\phi\} \not\approx_\Omega \psi$ if $\Lambda \not\approx_\Omega \phi \Rightarrow \psi$	(Deduction)	N
$Cn_{\not\approx_\Omega}(\Lambda) = Cn_{\not\approx_\Omega}(Cn_{\not\approx_\Omega}(\Lambda))$	(Idempotence)	N

Table I
PROPERTIES OF EXISTENTIAL Ω-ENTAILMENT. THE FINAL COLUMN SHOWS WHETHER THE PROPERTY HOLDS (Y) OR NOT (N).

Resolution entailment corresponds to quasi-classical logic [10], restricted to clauses and literals.

Example 1 (Paraconsistency of Resolution Entailment)**.** *Consider the following theory:*

$$\{p \vee q, r, \neg r\}$$

The theory is inconsistent due to the presence of r and $\neg r$. However, the only clause in the resolution closure of the theory that subsumes p is the empty clause, and so p is not resolution entailed.

The following proposition says that the conclusions of existential Ω-entailment are contained in the conclusions of resolution entailment.

Proposition 5. *Let Λ be a set of data, let Ω be a set of clauses, and let ϕ be a literal. Then $\Lambda \cup \Omega \not\approx_R \phi$ if $\Lambda \not\approx_\Omega \phi$.*

Proof of Proposition 5. Since $\Lambda \not\approx_\Omega \phi$, there is an Ω-consistent subset λ of Λ such that $\lambda \cup \Omega \models \phi$. By the Subsumption Theorem [11], there is a clause δ that is resolution derivable from Ω such that δ subsumes ϕ. Since $\lambda \cup \Omega \not\models \bot$, we see that δ cannot be the empty clause. Thus $\lambda \cup \Omega \not\approx_R \phi$, and by monotonicity, $\Lambda \cup \Omega \not\approx_R \phi$. □

However, the opposite is not the case, i.e. the conclusions of resolution entailment are not contained in the conclusions of existential Ω-entailment.

Proposition 6. *There exists a set of data Λ, a set of clauses Ω, and a literal ϕ such that $\Lambda \cup \Omega \not\approx_R \phi$ but $\Lambda \not\approx_\Omega \phi$.*

Proof Let $\Omega = \{\neg p \vee \neg q, \neg p \vee \neg q \vee r\}$, let $\Lambda = \{p, q\}$, and let $\phi = r$. □.

VI. COMPUTING EXISTENTIAL Ω-ENTAILMENT

We now turn to the question of how to compute whether a fact is existentially Ω-entailed from a set of data.

Going straight from the definition, we can reduce the problem to resolution as follows. First, we define a procedure for determining if $\Lambda \cup \Omega$ is consistent with a literal ϕ. For this, we use the fact that resolution is refutation complete [4]. While this procedure is undecidable in general, we note that it is decidable for the special cases of propositional logic [4], monadic logic [12] and Finite Herbrand Logic [13]:

Algorithm 1 IS-CONSISTENT[Ω]

```
if Ω resolves to the empty clause then
    return false
else
    return true
end if
```

Now, we are able to obtain an algorithm for computing whether a fact is existentially Ω-entailed by a set of data Λ and a set of constraints Ω.

Algorithm 2 NAÏVE-ALGORITHM[Λ, Ω, ϕ]

```
for all subsets λ of Λ do
    if IS-CONSISTENT(λ ∪ Ω) then
        if ¬IS-CONSISTENT(λ ∪ Ω ∪ {¬φ}) then
            return true
        end if
    end if
end for
return false
```

It is easy to see that the algorithm is correct, as it translates quite directly from the definition of existential Ω-entailment. Unfortunately, even in the best case, this algorithm requires iterating through all subsets of Λ, the number of which is exponential in the size of Λ.

We can do better in general by realizing that, to determine whether a fact that is entailed by *some* subset of data, there is no need to do brute-force search through *every* subset of data. Intuitively, we should intelligently choose what subsets to look through, ignoring other "irrelevant" subsets altogether.

Example 2. *Consider $\Omega = \{\neg p(X) \vee \neg q(X) \vee r(X), \neg p(a) \vee \neg q(a)\}$ and $\Lambda = \{p(a), p(b), q(a), q(b), s(b)\}$. Intuitively, when querying for $r(b)$, $s(b)$ is irrelevant, in the sense that there is no need to consider subsets of Λ that contain $s(b)$. Only subsets that contain $p(b)$, $q(b)$, or $r(b)$ are relevant.*

How can we make this intuitive notion of relevance a precise concept? A proof-theoretic view is of use here. In particular, note that one can prove fact ϕ from data Λ and constraints Ω iff there is a resolution refutation of $\neg\phi$ from the clausal form of $\Lambda \cup \Omega$. Thus, we only need to consider subsets of Λ that are used in some minimal-length resolution refutation of $\neg\phi$. Thus, instead of iterating blindly through all consistent subsets and testing whether they entail ϕ, we can build up a resolution refutation of $\neg\phi$, ensuring that all the premises of the refutation are Ω-consistent.

A *clause with lineage* is a 2-tuple $\langle \phi, \Delta \rangle$, where ϕ is a clause and Δ is a set of clauses. We will use clauses with lineage to represent the fact that ϕ was derived using facts in Δ.

Using clauses with lineage, we can modify the traditional definition of resolution to derive only Ω-consistent conclusions. If a subset of the literals in a clause Φ has a most general unifier γ, then the clause Φ' obtained by applying γ to Φ is called a *factor* of Φ. Suppose that $\langle \Phi, \Lambda \rangle$ and $\langle \Psi, \Delta \rangle$ are two clauses with lineage. If (1) there is a literal ϕ in some factor Φ' of Φ and a literal $\neg\psi$ in some factor Ψ' of Ψ such that ϕ and ψ have a most general unifier γ and (2) $\Lambda \cup \Delta \cup \Omega$ is consistent, then we say that the two clauses Φ and Ψ Ω-*resolve* and that the new clause with lineage $\langle ((\Phi' - \{\phi\}) \cup (\Psi' - \{\neg\psi\}))\gamma, \Lambda \cup \Delta \rangle$ is a Ω-*resolvent* of the two clauses.

A Ω-*resolution derivation* of a clause ϕ from a set of clauses with lineage Δ is a sequence of clauses with lineage terminating in $\langle \phi, \delta \rangle$ for some δ in which each item is either (1) a member of Δ (a *premise*) or (2) the result of applying Ω-resolution to earlier items in the sequence.

A sentence ϕ is *provable* from a set of sentences Δ by Ω-resolution if and only if there is an Ω-resolution derivation of the empty clause from the set $lineage(\Delta, \Omega, \phi)$, where $lineage(\Delta, \Omega, \phi)$ is defined as consisting of the following clauses with lineage:

(1) $\langle \psi, \{\psi\} \rangle$, for each ψ in the clausal form of Δ such that ψ is consistent with Ω;
(2) $\langle \psi, \emptyset \rangle$, for each ψ in the clausal form of Ω;
(3) $\langle \psi, \emptyset \rangle$, for each ψ in the clausal form of $\neg\phi$.

Ω-resolution does what we hope to accomplish - generate all resolution derivations except those whose premises are not Ω-consistent.

Theorem 7 (Soundness and Completeness)**.** $\Lambda \mid\!\approx_\Omega \phi$ *iff* ϕ *is provable from* Λ *by* Ω*-resolution.*

Proof of Theorem 7. Let Λ and Ω be sets of sentences, and let ϕ be a sentence. We have:

$\Lambda \mid\!\approx_\Omega \phi$
iff there is an Ω-consistent subset λ of Λ such that $\lambda \cup \Omega \models \phi$
iff there is a resolution derivation $D = \langle \psi_1, \ldots, \psi_k \rangle$ from the clausal form of $\lambda \cup \Omega \cup \{\neg\phi\}$ such that $\psi_k = \emptyset$
iff there is an Ω-resolution derivation D' of $\emptyset$ from $lineage(\lambda, \Omega, \phi)$, where D' is exactly D except each resolvent ψ_i, $1 \leq i \leq k$ has been replaced by a corresponding clause with lineage $\Psi_i = \langle \psi_i, \delta \rangle$ such that the lineage δ is (1) $\emptyset$, if the ψ_i is in the clausal form of Ω or $\neg\phi$, (2) ψ_i, if ψ_i is in the clausal form of λ, or (3) the union of the lineages of the clauses that resolve to ψ_i, otherwise.
iff ϕ is Ω-derivable from $lineage(\Lambda, \Omega, \phi)$
iff ϕ is provable from Λ by Ω-resolution. $\square$

We illustrate Ω-resolution with a few examples. We begin by revisiting Example 2.

Example 3. *Let* $\Omega = \{\neg p(X) \vee \neg q(X) \vee r(X), \neg p(a) \vee \neg q(a)\}$ *and* $\Lambda = \{p(a), p(b), q(a), q(b), s(b)\}$*. Then we can prove* $r(b)$ *from* Ω *and* Λ *via* Ω*-resolution:*

(1)	$\langle \neg r(b), \{\} \rangle$	*(goal)*
(2)	$\langle p(b), \{p(b)\} \rangle$	*(premise)*
(3)	$\langle q(b), \{q(b)\} \rangle$	*(premise)*
(4)	$\langle \neg p(X) \vee \neg q(X) \vee r(X), \{\} \rangle$	*(premise)*
(5)	$\langle \neg q(b) \vee r(b), \{p(b)\} \rangle$	*[2,4]*
(6)	$\langle r(b), \{p(b), q(b)\} \rangle$	*[3,5]*
(7)	$\langle \{\}, \{p(b), q(b)\} \rangle$	*[1,6]*

Note that the lineage of the empty clause of the proof is $\{p(b), q(b)\}$*. In particular, note that* $s(b)$ *is not included. This illustrates how* Ω*-resolution avoids considering irrelevant subsets of* Λ*, as discussed above.*

Also, as desired, we cannot prove $r(a)$ *from from* Ω *and* Λ *via* Ω*-resolution:*

(1)	$\langle \neg r(a), \{\} \rangle$	*(goal)*
(2)	$\langle p(a), \{p(a)\} \rangle$	*(premise)*
(3)	$\langle q(a), \{q(a)\} \rangle$	*(premise)*
(4)	$\langle \neg p(X) \vee \neg q(X) \vee r(X), \{\} \rangle$	*(premise)*
(5)	$\langle \neg q(a) \vee r(a), \{p(a)\} \rangle$	*[2,4]*
(6)	$\langle \neg q(a), \{p(a)\} \rangle$	*[1,5]*
~~*(7)*~~	~~$\langle \{\}, \{p(a), q(a)\} \rangle$~~	~~*[1,6]*~~

Step (7) is not allowed since $\{p(a), q(a)\}$ *is inconsistent with* Ω*; in particular it is inconsistent with* $\neg p(a) \vee \neg q(a)$*.*

VII. Related Work

There are a rich variety of paraconsistent logics. The interested reader is referred to [14] for a recent survey. These logics are generally concerned with modifying the inference rules of traditional logic so as to avoid concluding nonsense in the face of inconsistency. There are a few exceptions. In particular, existential entailment, first defined by [5], is defined as the consequences that are logically entailed by some consistent subset of a theory. Existential Ω-entailment can be thought of as existential entailment with a nondefeasible set of constraints Ω.

Existential Ω-entailment was first introduced in [1]. It was later studied in [3], which gives a method for transforming

a set of clausal constraints Ω into an open Datalog program that, together with a set of data, entails the existentially Ω-entailed facts for the data.

Existential entailment was revisited in [6] and [7] which introduced a family of argumentative logics, which are subset based. In addition to existential entailment, argumentative logics include the dual notion of universal entailment, which is defined to be the consequences that are logically entailed by every maximally consistent subset of a theory. [15] gives an algorithm for constructing arguments in propositional logic using connection graphs.

Defeasible Logic Programming [16] is a logic programming language which partitions the logic program into a set of feasible rules and data and a set of nondefeasible rules and data. Conclusions for which the negation is not also a conclusion are considered to be answers.

Consistent query answering is introduced in [17]. Unlike the previously mentioned consequence relations, consistent query answering distinguishes between the constraints and the data. A consistent query answer is an answer that is entailed in all minimal repairs of the data.

To understand how the aforementioned consequence relations relate to one another, we can classify them according to following questions. (1) Does the consequence relation distinguish between constraints and data (in particular, is it parameterized by constraints and operate on data)? Or does it treat all sentences equally? (2) Is the consequence relation based on "possible worlds"? In other words, does the consequence relation consider each consistent subset of the theory or the data to produce some results and then combine the results in some way? If so: (a) Does the consequence relation require that a conclusion is true in *every* possible world (it is *universal*) or just *some* possible world (it is *existential*)? (b) Are the possible worlds defined to be *subsets* of the original data, or are they defined to be *minimal repairs* of the original data?

The answers to these questions are summed up in the following table. The table lists the consequence relations most closely related to existential Ω-entailment and describes whether they are universal ($\forall$) or existential ($\exists$), whether they are subset ($\subseteq$) or repair-based (Repair), and whether the relations consider the constraints separately from the data ($\neq$), or whether they are treated together in a uniform fashion (=).

Consequence relation	$\forall/\exists$	$\subseteq$**/Repair**	$\neq$ / =
Existential Ω-entailment	$\exists$	$\subseteq$	$\neq$
Existential entailment	$\exists$	$\subseteq$	$=$
Universal entailment	$\forall$	$\subseteq$	$=$
Consistent query answering	$\forall$	Repair	$\neq$

REFERENCES

[1] M. Kassoff, L.-M. Zen, A. Garg, and M. R. Genesereth, "PrediCalc: A logical spreadsheet management system," in *VLDB*, 2005.

[2] T. L. Hinrichs, "Plato: A compiler for interactive web forms," in *PADL*, 2011, pp. 54–68.

[3] T. L. Hinrichs, J.-Y. Kao, and M. R. Genesereth, "Inconsistency-tolerant reasoning with classical logic and large databases," in *SARA*, 2009.

[4] M. R. Genesereth and N. Nilsson, *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, 1987.

[5] R. Manor and N. Rescher, "On inferences from inconsistent information," *Theory and Decision*, vol. 1, no. 1, pp. 179–219, 1970.

[6] S. Benferhat, D. Dubois, and H. Prade, "Argumentative inference in uncertain and inconsistent knowledge bases," in *UAI'93*. Morgan Kaufmann, 1993, pp. 411–419.

[7] M. Elvang-Gøransson and A. Hunter, "Argumentative logics: Reasoning with classically inconsistent information," *Data Knowledge Engineering*, vol. 16, no. 2, pp. 125–145, 1995.

[8] S. Kraus, D. Lehmann, and M. Magidor, "Nonmonotonic reasoning, preferential models and cumulative logics," *Artif. Intell.*, vol. 44, pp. 167–207, July 1990.

[9] M. Kassoff, *Updating Logical Spreadsheets*. Stanford University, Forthcoming.

[10] P. Besnard and A. Hunter, "Quasi-classical Logic: Non-trivializable Classical Reasoning from Inconsistent Information," in *ECSQARU'95, volume 946 LNCS*, vol. 946. Springer, 1995, pp. 44–51.

[11] S. Nienhuys-cheng and R. de Wolf, "The subsumption theorem in inductive logic programming: Facts and fallacies," in *Advances in Inductive Logic Programming. IOS*. Press, 1995, pp. 265–276.

[12] G. S. Boolos and R. C. Jeffrey, *Computability and logic*. New York, NY, USA: Cambridge University Press, 1989.

[13] T. L. Hinrichs, *Extensional Reasoning*. Stanford University, 2007.

[14] G. Priest and K. Tanaka, "Paraconsistent logic," in *The Stanford Encyclopedia of Philosophy*, summer 2009 ed., E. N. Zalta, Ed., 2009.

[15] V. Efstathioua and A. Hunter, "Algorithms for generating arguments and counterarguments in propositional logic," *International Journal of Approximate Reasoning*, 2011.

[16] A. J. García and G. R. Simari, "Defeasible logic programming: an argumentative approach," *Theory Pract. Log. Program.*, vol. 4, pp. 95–138, January 2004.

[17] M. Arenas, L. Bertossi, and J. Chomicki, "Consistent query answers in inconsistent databases," in *PODS '99*. New York, NY, USA: ACM, 1999, pp. 68–79.