Evaluating violations of expectations to find exceptional information

Emma Byrne and Anthony Hunter*
Department of Computer Science
University College London
Gower Street
London WC1E 6BT, UK

September 23, 2004

Keywords: Analysing inconsistency; Measuring inconsistency; Expectations; Contradiction; Exceptions; Defeasible information; Default information.

Abstract

Much useful new information (e.g. information in news reports) is often that which is surprising or unexpected. In other words, we harbour many expectations about the world, and when any of these expectations are violated (i.e. made inconsistent) by new information, we have a strong indicator that the information is interesting for us. An expectation can be compared with an integrity constraint. Both an expectation and integrity constraint can be represented by an implicational formula in classical logic, and every time we get new information, we compare it with the implicational formula. However, with an integrity constraint, we are primarily seeking information that is consistent with the implicational formula. In constrast, with an expectation, we are primarily seeking information that is inconsistent with the implicational formula. In this paper, we present a framework for representing and analysing expectations. We consider for an application language the syntax of expectations, the accuracy and validity of expectations, and we explore relationships between these issues. We also consider representing and reasoning with expectations as part of an application in merging information.

1 Introduction

Expectations are an interesting, and potentially valuable, form of default information. We can view each expectation as a rule that describes how if the condition is true, then the consequent is expected to be true. So whilst an expectation is not 100% accurate, it is normally correct, and so when it is violated, we may regard the information that causes the violation as exceptional. Furthermore much new information that is useful can often be described as being unexpected, and hence exceptional. In other words, we harbour many expectations about the world, and when any of these expectation are violated (i.e. made inconsistent) by new information, we have a strong indicator that this new information is interesting for us. However, the majority of new information is either consistent with expectations, or confirms expectations. We can see this with news reports, particularly in the case of business, technical, and scientific news, where the information in news reports normally continues existing trends or reinforces expected information.

*Corresponding author

To illustrate, an expectation arising with business news reports is “when a company makes a takeover bid for another company, the company has sufficient financial resources to pay for the bid”. As another example, “when a company is in administration, the consolidated accounts show that the company is making a net loss”. Later, we will consider expectations arising in biomedical research.

An expectation can be compared with an integrity constraint. Both an expectation and intregrity constraint can be represented by an implicational formula in classical logic, and every time we get new information, we can compare it with the implicational formula. However, with an integrity constraint, we are primarily seeking information that is consistent with the implicational formula. In constrast, with an expectation, we are primarily seeking information that is inconsistent with the implicational formula.

We can also compare expectations and integrity constraints in terms of the relative “correctness” of the information involved. An integrity constraint is assumed to be always correct, and any data that is inconsistent with the integrity constraint is assumed to be incorrect. In contrast, an expectation is assumed to be normally correct, and any data that is inconsistent with the expectation is assumed to be always correct.

An expectation can also be compared with the commonly considered usage of a default (or defeasible) rule. It is noteworthy that human practical reasoning relies much more on exploiting default information than on a myriad of individual facts. Default information tends to be less than 100% accurate, and so has exceptions. Nevertheless it is intuitive and advantageous to resort to such defaults and therefore allow the inference of useful conclusions, even if it does entail making some mistakes as not all exceptions to these defaults are necessarily known. Furthermore, it is often necessary to use default rules when we do not have sufficient information to allow us to specify or use universal laws. So with the common discussed usage of default information, the aim is to use the default information to find the unexceptional situations. This contrasts with the usage of expectations where the aim is to find the abnormal (i.e. exceptional) situations. In this way, this paper raise an opportunity to revisit the notion of default information as explored in the non-monotonic reasoning literature (for a review see [BDK97]), and consider a different role for it.

In a previous paper [BH04], we presented a framework for identifying interesting information in structured news reports by finding interesting inconsistencies. Each structured news report is a news report in the form of an XML document with each textentry being restricted to being a proper noun, a number, a unit, or individual word or short phrase from a domain-dependent ontology. An implemented system based on the framework (1) accepts structured news reports as inputs, (2) translates each report into a logical literal, (3) identifies the story of which the report is a part, and thereby identifying the relevant background facts pertaining to the report, (4) looks for inconsistencies between the report, the background knowledge, and a set of expectations, (5) classifies and evaluates those inconsistencies, and (6) outputs news reports of interest to the user together with associated explanations of why they are interesting. This explanation is in terms of the expectation violated, its accuracy and the background facts used in generating the inconsistency.

Whilst this previous paper introduced the idea of expectations, the emphasis of the paper was on practical aspects of representing and reasoning with the news reports and the background knowledge. The logic of expectations, and measures of accuracy and validity, were left unexplored. So in this paper, we focus on expectations with the aim of generalizing and clarifying the nature of the logic, and the associated measures of accuracy and validity. The net result of this paper is that we can better understand the meaning of expectations, we can develop better automated approaches to obtaining and using expectations, and we can employ them in wider roles for analysing new information.

In summary, in this paper, we provide a logic-based framework for evaluation of violations of expectations. We proceed as follows: (Section 2) We define the language of expectations; (Section 3) We formalize the violation of an expectation by new information in the form of a report; (Section 4) We formalize the accuracy and validity of expectations; (Section 5) We consider how to select the best expectations to use based on syntactic form, and a combination of accuracy and validity; (Section 6) We consider how to generate a set of expectations given a set of reports; and (Section 7) We consider an application of the framework in information fusion.

2 Expectation spaces

In this section, we define expectations as formulae of classical logic. For the language of expectations, we assume a set of predicate symbols P, and a set of variable symbols X. We assume that both of these sets are finite.

Definition 2.1. *Let* $\mathsf{FreeAtoms}(P, X, n)$ *be the set of atoms of the form* $p(x_1, .., x_k)$ *where* p *is a predicate symbol in* P, $x_1, .., x_k$ *are variable symbols in* X*, and* n *is the maximum arity of the atoms, and* $1 \leq k \leq n$*. Let* $\mathsf{FreeLiterals}(P, X, n)$ *be the set of literals defined as follows where*

$$\mathsf{FreeLiterals}(P, X, n) = \mathsf{FreeAtoms}(P, X, n) \cup \{\neg p(x_1, .., x_n) \mid p(x_1, .., x_n) \in \mathsf{FreeAtoms}(P, X, n)\}$$

Let $\mathsf{FreeFormulae}(P, X, n)$ *be the set of formulae formed from* $\mathsf{FreeAtoms}(P, X, n)$ *using the logical connectives* $\neg$, $\wedge$*, and* $\vee$*. For* $\alpha \in \mathsf{FreeFormulae}(P, X, n)$*, let* $\mathsf{FreeVariables}(\alpha)$ *be the free variables in* α*.*

So the formulae in the set FreeFormulae only have variables that are free. In other words, there are no quantifiers in these formulae.

In our framework, expectations are implicational formulae in classical logic. Furthermore, each variable is in the scope of a universal quantifier, and the universal quantifiers are all outermost.

Definition 2.2. *Let* $\mathsf{Expectations}(P, X, n)$ *be the set of expectations of the form* $\forall \bar{x}(\alpha \rightarrow \beta)$ *where* $\alpha, \beta \in \mathsf{FreeFormulae}(P, X, n)$ *and* $\bar{x}$ *is a tuple formed from all the variables in* $\mathsf{FreeVariables}(\alpha) \cup \mathsf{FreeVariables}(\beta)$*.*

So whilst each expectation is represented by a classical formula, we will see that the way we use them allows us to capture default information with them.

Definition 2.3. *For* $\forall \bar{x}(\alpha \rightarrow \beta) \in \mathsf{Expectations}(P, X, n)$*, let* $\bar{y}$ *be a tuple of all variables in* $\mathsf{FreeVariables}(\alpha)$*. and* $\bar{z}$ *be a tuple of all variables in* $\mathsf{FreeVariables}(\beta)$*.*

$$\mathsf{Antecedent}(\forall \bar{x}\ (\alpha \rightarrow \beta)) = \forall \bar{y}\ \alpha$$
$$\mathsf{Consequent}(\forall \bar{x}\ (\alpha \rightarrow \beta)) = \forall \bar{z}\ \beta$$
$$\mathsf{ContraConsequent}(\forall \bar{x}\ (\alpha \rightarrow \beta)) = \forall \bar{z}\ \neg\beta$$

Example 2.1. *Let* ϕ *be the expectation* $\forall x, y, z(p(x) \wedge q(x, z) \rightarrow r(y, z))$*, for which we get the following*

$$\mathsf{Antecedent}(\phi) = \forall x, z\ (p(x) \wedge q(x, z))$$
$$\mathsf{Consequent}(\phi) = \forall y, z\ (r(y, z))$$
$$\mathsf{ContraConsequent}(\phi) = \forall y, z\ (\neg r(y, z))$$

Definition 2.4. *Let* $\phi, \psi \in \mathsf{Expectations}(P, X, n)$*. The* **antecedent ordering** $\geq_a$ *and the* **consequent ordering** $\geq_c$ *are defined as follows.*

$$\phi \geq_a \psi \textit{ iff } \{\mathsf{Antecedent}(\phi)\} \vdash \mathsf{Antecedent}(\psi)$$

$$\phi \geq_c \psi \textit{ iff } \{\mathsf{Consequent}(\phi)\} \vdash \mathsf{Consequent}(\psi)$$

Let $\phi =_a \psi$ *denote* $\phi \geq_a \psi$ *and* $\psi \geq_a \phi$ *and let* $\phi =_c \psi$ *denote* $\phi \geq_c \psi$ *and* $\psi \geq_c \phi$*. Also let* $\phi >_a \psi$ *denote* $\phi \geq_a \psi$ *and* $\psi \not\geq_a \phi$ *and let* $\phi >_c \psi$ *denote* $\phi \geq_c \psi$ *and* $\psi \not\geq_c \phi$*.*

Example 2.2. *The following illustrate the antecedent and consequent orderings.*

$$\forall x, y, z\ (p(x) \wedge q(x, z) \rightarrow r(y, z)) \geq_a \forall x, y, z\ (p(x) \rightarrow r(y, z))$$

$$\forall x, y, z\ (p(x) \wedge q(x, z) \rightarrow r(y, z)) \geq_c \forall x, y, z, v\ (p(x) \rightarrow r(y, z) \vee s(v))$$

The antecedent ordering and consequent ordering are pre-orderings (i.e. reflexive and transitive but not anti-symmetric).

Proposition 2.1. *Let* $\phi \in \mathsf{Expectations}(P, X, n)$*,*

1. $\phi \vdash \bot$ *iff (*ϕ *is minimal in the* $\geq_a$ *ordering and* ϕ *is maximal in the* $\geq_c$ *ordering).*
2. $\top \vdash \phi$ *iff (*ϕ *is maximal in the* $\geq_a$ *ordering and* ϕ *is minimal in the* $\geq_c$ *ordering).*

Proof. (1) $\phi \vdash \bot$ iff $\vdash \phi \leftrightarrow \forall \bar{x}(\alpha \vee \neg\alpha \rightarrow \alpha \wedge \neg\alpha)$ iff for all expectations ψ in $\mathsf{Expectations}(P, X, n)$, $\{\mathsf{Consequent}(\forall \bar{x}(\alpha \vee \neg\alpha \rightarrow \alpha \wedge \neg\alpha))\} \vdash \mathsf{Consequent}(\psi)$ and $\{\mathsf{Antecedent}(\psi)\} \vdash \mathsf{Antecedent}(\forall \bar{x}(\alpha \vee \neg\alpha \rightarrow \alpha \wedge \neg\alpha))$ iff ϕ is minimal in the $\geq_a$ ordering and ϕ is maximal in the $\geq_c$ ordering. (2) $\top \vdash \phi$ iff $\vdash \phi \leftrightarrow \forall \bar{x}(\alpha \wedge \neg\alpha \rightarrow \alpha \vee \neg\alpha)$ iff for all $\psi \in \mathsf{Expectations}(P, X, n)$, $\{\mathsf{Antecedent}(\forall \bar{x}(\alpha \wedge \neg\alpha \rightarrow \alpha \vee \neg\alpha))\} \vdash \mathsf{Antecedent}(\psi)$ and $\{\mathsf{Consequent}(\psi)\} \vdash \mathsf{Consequent}(\forall \bar{x}(\alpha \wedge \neg\alpha \rightarrow \alpha \vee \neg\alpha))$ iff ϕ is maximal in the $\geq_a$ ordering and ϕ is minimal in the $\geq_c$ ordering. $\square$

We will see that not all expectations are useful to us. The next definition specifies four particular types of expectation that we will argue, in Section 4, as not being useful.

Definition 2.5. *Let* $\phi \in \mathsf{Expectations}(P, X, n)$*.*

ϕ *is* **self-reinforcing** *iff* $\{\mathsf{Antecedent}(\phi)\} \vdash \mathsf{Consequent}(\phi)$

ϕ *is* **self-defeating** *iff* $\{\mathsf{Antecedent}(\phi)\} \vdash \mathsf{ContraConsequent}(\phi)$

ϕ *is* **non-firing** *iff* $\{\mathsf{Antecedent}(\phi)\} \vdash \bot$

ϕ *is* **non-commiting** *iff* $\top \vdash \mathsf{Consequent}(\phi)$

The simple examples below indicate why these kinds of expectation are not useful. In Section 4, we will be able to formalise why they are not useful.

Example 2.3. *The expectation* $\forall x(a(x) \wedge b(x) \rightarrow a(x))$ *is self-reinforcing. The expectation* $\forall x(a(x) \wedge b(x) \rightarrow \neg a(x))$ *is self-defeating. The expectation* $\forall x(a(x) \wedge \neg a(x) \rightarrow b(x))$ *is non-firing. The expectation* $\forall x(b(x) \rightarrow a(x) \vee \neg a(x))$ *is non-commiting.*

When we select a set of expectations, we will also want to minimize various kinds of redundancy. One kind of redundancy we will consider in Section 5 comes from logically equivalent expectations that are defined as follows.

Definition 2.6. *For all* $\phi, \psi \in \mathsf{Expectations}(P, X, n)$*,*

ϕ *is* **logically equivalent** *to* ψ *iff* $\{\phi\} \vdash \psi$ *and* $\{\psi\} \vdash \phi$

Example 2.4. *The expectation* $\forall x(a(x) \wedge b(x) \rightarrow c(x))$ *is logically equivalent to the expectation* $\forall x(b(x) \wedge a(x) \rightarrow c(x))$*.*

An expectation class is a maximal subset of $\mathsf{Expectations}(P, X, n)$ that contains no logically equivalent formulae.

Definition 2.7. *For* $\mathsf{Expectations}(P, X, n)$*, the set of* **expectation classes** *is defined as follows.*

$$\begin{aligned}&\mathsf{ExpectationClasses}(P, X, n) = \\ &\quad \{\Lambda \subseteq \mathsf{Expectations}(P, X, n) \mid \forall \phi \in \mathsf{Expectations}(P, X, n)\ \exists \psi \in \Lambda \\ &\qquad \textit{s.t. } \phi \textit{ is logical equivalent to } \psi \textit{ and } \forall \phi', \psi' \in \Lambda, \phi' \textit{ is not logically equivalent to } \psi'\}\end{aligned}$$

Logically equivalent expectations are not always comparable using the $\geq_a$ and the $\geq_c$ orderings as illustrated in the next example.

Example 2.5. *Consider the expectations ϕ_1 and ϕ_2 which are logically equivalent. For these, $\phi_1 \not\geq_a \phi_2$, $\phi_2 \not\geq_a \phi_1$, $\phi_1 \not\geq_c \phi_2$, and $\phi_2 \not\geq_c \phi_1$.*

$$\phi_1 = \forall x(\neg a(x) \wedge \neg b(x) \rightarrow c(x))$$
$$\phi_2 = \forall x(\neg a(x) \wedge \neg c(x) \rightarrow b(x))$$

In order to further compare expectations, we combine the antecedent ordering and the consequent ordering to give the expectation ordering over expectations.

Definition 2.8. *Let $\phi, \psi \in \mathsf{Expectations}(P, X, n)$. The* **expectation ordering** *$\geq_e$ is defined as follows.*

$$\phi \geq_e \psi \textit{ iff } \phi \geq_a \psi \textit{ and } \psi \geq_c \phi$$

Let $\phi =_e \psi$ denote $\phi \geq_e \psi$ and $\psi \geq_e \phi$.

Example 2.6. *Let ϕ be $\forall x(a(x) \wedge b(x) \rightarrow c(x) \vee d(x))$ and let ψ be $\forall x(a(x) \rightarrow c(x))$. So $\phi \geq_a \psi$ and $\psi \geq_c \phi$, and therefore $\phi \geq_e \psi$.*

The set of expectations in $\mathsf{Expectations}(P, X, n)$ can be compared using the expectation ordering.

Definition 2.9. *An* **expectation space** *is a pair $(\mathsf{Expectations}(P, X, n), \geq_e)$ where $\geq_e$ is the expectation ordering.*

The expectation ordering provides a partial ordering over expectations. The anti-symmetry means that if $\phi =_e \psi$ holds, then ϕ and ψ are interchangeable. Furthermore, the $=_e$ relationship is a type of logical equivalence as shown in the following result.

Proposition 2.2. *For $\phi, \psi \in \mathsf{Expectations}(P, X, n)$, if $\phi =_e \psi$, then ϕ is logically equivalent to ψ.*

Proof. For $\phi, \psi \in \mathsf{Expectations}(P, X, n)$, assume $\phi =_e \psi$. Therefore, $\phi =_a \psi$ and $\phi =_c \psi$. Therefore, $\vdash \mathsf{Antecedent}(\phi) \leftrightarrow \mathsf{Antecedent}(\psi)$ and $\vdash \mathsf{Consequent}(\phi) \leftrightarrow \mathsf{Consequent}(\psi)$. Therefore, $\{\phi\} \vdash \psi$ and $\{\psi\} \vdash \phi$. Therefore, ϕ is logically equivalent to ψ. $\square$

We can show that an expectation space is a Boolean lattice. Furthermore, using the definition of a bilattice [Gin86, Fit89], we can present an expectation space as a bilattice: For an expectation space $(\Lambda, \geq_e)$, we can show that $(\Lambda, \geq_a)$ and $(\Lambda, \geq_c)$ are Boolean lattices, we can show they are isomorphic, and we can show that the meet and join operations for one lattice is monotone with respect to the other lattice [Byr05].

3 Expectation violation

In the framework in this paper, we assume new information is input in the form of reports. Each report is a consistent set of literals. Each time a report is input, it is compared with a set of expectations to identify which are violated by the report. An expectation is violated by a report when the union of the expectation and the report is inconsistent.

In this section, we consider the language of reports, based on a finite set of predicate symbols P and finite set of constant symbols C, and then we consider the definition of an expectation violation.

Definition 3.1. *Let* $\mathsf{GroundAtoms}(P, C, n)$ *be the set of ground atoms of the form* $p(c_1, .., c_k)$ *where* p *is a predicate symbol in* P*,* $c_1, .., c_k$ *are constant symbols in* C*, and* n *is the maximum arity of the atoms and* $1 \leq k \leq n$*. Let* $\mathsf{GroundLiterals}(P, C, n)$ *be the set of ground literals defined as follows.*

$$\begin{aligned}&\mathsf{GroundLiterals}(P, C, n) = \\ &\qquad \mathsf{GroundAtoms}(P, C, n) \cup \{\neg p(c_1, .., c_k) \mid p(c_1, .., c_k) \in \mathsf{GroundAtoms}(P, C, n)\}\end{aligned}$$

Whilst a report is a consistent set of literals. In practice, this is a minimal requirement, and further constraints are likely to be required for particular applications.

Definition 3.2. *Let* $\mathsf{Reports}(P, C, n)$ *be the set of reports where for each* $\rho \in \mathsf{Reports}(P, C, n)$*,* $\rho \not\vdash \bot$ *and* $\mathsf{Reports}(P, C, n) \subset \wp(\mathsf{GroundLiterals}(P, C, n))$*.*

A complete report is a report that by set inclusion is maximal in $\mathsf{Reports}(P, C, n)$. Therefore, for every ground atom in the language, a complete report contains either the atom or its negation.

Definition 3.3. *For* $\rho \in \mathsf{Reports}(P, C, n)$*,*

ρ *is a* **complete report** *iff for every* $\delta \in \mathsf{GroundAtoms}(P, C, n), \rho \vdash \delta$ *or* $\rho \vdash \neg\delta$

In some applications, the reports available would be complete reports. In some other applications, complete reports can be generated (if appropriate) from reports using the closed world assumption [Rei78].

Definition 3.4. *A* **grounding assignment** *is an equality predicate where the first argument is a variable symbol and the second argument is a constant symbol. A* **grounding** *is a set of grounding assignments. For a formula* α *and a grounding* Φ*,* $\mathsf{Ground}(\alpha, \Phi)$ *gives the result of substituting each occurrence of the variable symbol* x *in* α *with the constant symbol* c *for each* $x = c \in \Phi$*.*

Example 3.1. *Let* ϕ *be the expectation* $\forall x, y\ (p(x, y) \rightarrow q(y))$ *and let the grounding* Φ *be* $\{x = c, y = d\}$*. For this,* $\mathsf{Ground}(\phi, \Phi) = p(c, d) \rightarrow q(d)$*.*

We now consider some important relationships that may hold between an expectation and a report.

Definition 3.5. *For all* P*,* C*,* X*, and* n*, if* $\rho \in \mathsf{Reports}(P, C, n)$*, and* $\phi \in \mathsf{Expectations}(P, X, n)$*, then*

ϕ *is* **fired** *by* ρ *iff there is grounding* Φ *such that* $\rho \vdash \mathsf{Ground}(\mathsf{Antecedent}(\phi), \Phi)$

ϕ *is* **attacked** *by* ρ *iff there is grounding* Φ *such that* $\rho \vdash \neg\mathsf{Ground}(\mathsf{Consequent}(\phi), \Phi)$

ϕ *is* **violated** *by* ρ *iff there is grounding* Φ
such that $\rho \vdash \mathsf{Ground}(\mathsf{Antecedent}(\phi), \Phi)$
and $\rho \vdash \neg\mathsf{Ground}(\mathsf{Consequent}(\phi), \Phi)$

ϕ *is* **confirmed** *by* ρ *iff there is grounding* Φ
such that $\rho \vdash \mathsf{Ground}(\mathsf{Antecedent}(\phi), \Phi)$
and $\rho \vdash \mathsf{Ground}(\mathsf{Consequent}(\phi), \Phi)$

Clearly, if ϕ is confirmed by ρ, then ϕ is fired by ρ. Also if ϕ is violated by ρ, then ϕ is fired by ρ and ϕ is attacked by ρ.

Example 3.2. *Let* ϕ *be* $\forall x, y(p(x) \wedge q(y) \rightarrow r(x, y))$ *and let* $\rho_1, .., \rho_6$ *be the following reports. Hence,* ϕ *is confirmed by* ρ_1*,* ϕ *is violated by* ρ_2*,* ϕ *is fired by* ρ_3*,* ϕ *is attacked by* ρ_4*,* ϕ *is violated by* ρ_5*, and* ϕ *is attacked by* ρ_6*.*

$$\begin{array}{ll}\rho_1 = \{p(a), q(b), r(a, b)\} & \rho_2 = \{p(c), q(d), \neg r(c, d)\} \\ \rho_3 = \{p(a), q(c)\} & \rho_4 = \{\neg r(a, b)\} \\ \rho_5 = \{p(d), q(c), \neg r(d, c), q(d)\} & \rho_6 = \{p(a), \neg r(d, e)\}\end{array}$$

If ϕ is confirmed by ρ, then $\rho \cup \{\phi\} \not\vdash \bot$. However, $\rho \cup \{\phi\} \not\vdash \bot$ does not necessarily imply that ϕ is confirmed by ρ. In contrast, the violated relationship can be rephrased in terms of inconsistency as follows.

Proposition 3.1. *Let $\rho \in \mathsf{Reports}(P, C, n)$, and let $\phi \in \mathsf{Expectations}(P, X, n)$.*

$$\phi \textit{ is violated by } \rho \textit{ iff } \rho \cup \{\phi\} \vdash \bot$$

Proof. ϕ is violated by ρ iff there is grounding Φ such that $\rho \vdash \mathsf{Ground}(\mathsf{Antecedent}(\phi), \Phi)$ and $\rho \vdash \neg\mathsf{Ground}(\mathsf{Consequent}(\phi), \Phi)$ iff there is grounding Φ such that $\rho \vdash \neg\mathsf{Ground}(\phi, \Phi)$ iff $\rho \cup \{\phi\} \vdash \bot$. □

If an expectation in $\mathsf{Expectations}(P, X, n)$ is violated by a report ρ, then there are other expectations in $\mathsf{Expectations}(P, X, n)$ that are violated by ρ, and these other expectations can be identified using the consequent and antecedent orderings.

Proposition 3.2. *Let $\rho \in \mathsf{Reports}(P, C, n)$, and let $\phi \in \mathsf{Expectations}(P, X, n)$, such that ϕ is violated by ρ. For $\psi \in \mathsf{Expectations}(P, X, n)$, if $\phi \geq_e \psi$ then ψ is violated by ρ.*

Proof. Let Φ be a grounding s.t. $\rho \vdash \mathsf{Ground}(\mathsf{Antecedent}(\phi), \Phi)$ and $\rho \vdash \neg\mathsf{Ground}(\mathsf{Consequent}(\phi), \Phi)$. So if $\phi \geq_a \psi$, then $\mathsf{Ground}(\mathsf{Antecedent}(\phi), \Phi) \vdash \mathsf{Ground}(\mathsf{Antecedent}(\psi), \Phi)$. If $\psi \geq_c \phi$, there is a grounding Φ' such that $\Phi \subseteq \Phi'$ and $\neg\mathsf{Ground}(\mathsf{Consequent}(\phi), \Phi') \vdash \neg\mathsf{Ground}(\mathsf{Consequent}(\psi), \Phi')$. Hence, $\rho \vdash \neg\mathsf{Ground}(\mathsf{Consequent}(\psi), \Phi')$, and $\rho \vdash \mathsf{Ground}(\mathsf{Antecedent}(\psi), \Phi')$. Therefore ψ is violated by ρ. □

Given a report ρ, we may want to consider all the information in the report in any violation. In other words, given ρ, we seek expectations ϕ such that ϕ is minimally violated by ρ as defined next.

Definition 3.6. *Let $\rho \in \mathsf{Reports}(P, C, n)$, and let $\phi \in \mathsf{Expectations}(P, X, n)$.*

ϕ is **minimally violated** *by ρ*
iff ϕ is violated by ρ and there is no $\rho' \subset \rho$ such that ϕ is violated by ρ'

Example 3.3. *Continuing Example 3.2, we have ϕ is minimally violated by ρ_2, but ϕ is not minimally violated by ρ_5.*

For a report ρ, we can justify the use of expectations ϕ such that ϕ is minimally violated by ρ by appealing to the Principle of Total Evidence. This principle has been established for reasoning with conditional probability statements. It says that when we have two or more conditional probability statements all of whose conditions are satisfied, then we should use the conditional probability statement with the most specific condition [Gil96]. The idea here is that when assessing the probability of a given event, we should calculate its probability value conditional on all the available evidence. The probability-theoretic Principle of Total Evidence is analogous to the notion of specificity as used in non-monotonic reasoning with default or defeasible rules.

Proposition 3.3. *Let $\rho \in \mathsf{Reports}(P, C, n)$, and let $\phi, \psi \in \mathsf{Expectations}(P, X, n)$.*

If ϕ is minimally violated by ρ, and $\phi \geq_e \psi$, then ψ is not necessarily minimally violated by ρ.

Proof. Suppose $\phi \geq_e \psi$, then ψ is violated by ρ, but there may be a $\rho' \subset \rho$ s.t. ψ is violated by ρ'. □

So now we can define a satisfactory way of taking a set of expectations and a report and determining the subset of expectations that are violated.

Definition 3.7. *Let $\rho \in \mathsf{Reports}(P, C, n)$, and let $\Delta \subseteq \mathsf{Expectations}(P, X, n)$. The set of violated expectations and the set of minimally violated expectations are defined as follows.*

$$\mathsf{ViolatedExpectations}(\Delta, \rho) = \{\phi \in \Delta \mid \phi \textit{ is violated by } \rho\}$$

$$\mathsf{MinViolatedExpectations}(\Delta, \rho) = \{\phi \in \Delta \mid \phi \textit{ is minimally violated by } \rho\}$$

Clearly, $\mathsf{MinViolatedExpectations}(\Delta, \rho) \subseteq \mathsf{ViolatedExpectations}(\Delta, \rho)$. So according to the Principle of Total Evidence, it may be preferrable to use $\mathsf{MinViolatedExpectations}(\Delta, \rho)$. However, if our reports contain information that is not relevant, then we may prefer to use $\mathsf{ViolatedExpectations}(\Delta, \rho)$. We return to these issues later.

Given a set of expectations Δ, and a report ρ, the set of violated expectations may be large. To discriminate between them we use measures of accuracy and validity as defined in the next section. So for example, using accuracy, the most accurate expectations that are violated could be presented to a user first. The definition for violated expectations (Definition 3.7) assumes a set of expectations Δ. Whilst this could be an arbitrarily chosen set, in Section 5, we will discuss how such a set of expectations could be chosen to meet some important criteria.

4 Accuracy and validity of expectations

Given an expectation ϕ, and set of reports Π, we can evaluate the number of reports such that ϕ is fired by each of them. We can get similar scores for attacked, violated, and confirmed. We use these to get values for the accuracy and validity of the expectation.

Definition 4.1. *For $\phi \in \mathsf{Expectations}(P, X, n)$ and for $\Pi \subseteq \mathsf{Reports}(P, C, n)$.*

$$\mathsf{Fired}(\phi, \Pi) = |\{\rho \in \Pi \mid \phi \textit{ is fired by } \rho\}|$$

$$\mathsf{Attacked}(\phi, \Pi) = |\{\rho \in \Pi \mid \phi \textit{ is attacked by } \rho\}|$$

$$\mathsf{Violated}(\phi, \Pi) = |\{\rho \in \Pi \mid \phi \textit{ is violated by } \rho\}|$$

$$\mathsf{Confirmed}(\phi, \Pi) = |\{\rho \in \Pi \mid \phi \textit{ is confirmed by } \rho\}|$$

Let $\mathsf{Unaffected}(\phi, \Pi) = \mathsf{Fired}(\phi, \Pi) - (\mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi))$.

Proposition 4.1. *For $\phi \in \mathsf{Expectations}(P, X, n)$, and for $\Pi \subseteq \mathsf{Reports}(P, C, n)$, it follows from Definition 3.5 that if all $\rho \in \Pi$ are complete reports, then $\mathsf{Fired}(\phi, \Pi) = \mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi)$, otherwise $\mathsf{Fired}(\phi, \Pi) > \mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi)$.*

The fired and attacked values are related to the antecedent and consequent orderings as follows.

Proposition 4.2. *For $\phi, \psi \in \mathsf{Expectations}(P, X, n)$ and for $\Pi \subseteq \mathsf{Reports}(P, C, n)$, it follows from Definition 3.5 that*

1. $\mathsf{Fired}(\phi, \Pi) \leq \mathsf{Fired}(\psi, \Pi)$ *iff* $\phi \geq_a \psi$
2. $\mathsf{Attacked}(\phi, \Pi) \leq \mathsf{Attacked}(\psi, \Pi)$ *iff* $\psi \geq_c \phi$

The accuracy of an expectation is dependent on the number of times it has been violated divided by the number of times it has been fired. We can regard the ratio of $\mathsf{Violated}(\phi, \Pi)/\mathsf{Fired}(\phi, \Pi)$ as the error ratio for ϕ. So to minimize error, we need to maximize accuracy.

Definition 4.2. *For* $\phi \in \mathsf{Expectations}(P, X, n)$ *and for* $\Pi \subseteq \mathsf{Reports}(P, C, n)$*, if* $\mathsf{Fired}(\phi, \Pi) = 0$*, then* $\mathsf{Accuracy}(\phi, \Pi) = 0$*, otherwise*

$$\mathsf{Accuracy}(\phi, \Pi) = 1 - \frac{\mathsf{Violated}(\phi, \Pi)}{\mathsf{Fired}(\phi, \Pi)}$$

The validity of an expectation is the number of times it has been confirmed divided by the number of times it has been fired.

Definition 4.3. *For* $\phi \in \mathsf{Expectations}(P, X, n)$ *and for* $\Pi \subseteq \mathsf{Reports}(P, C, n)$*, if* $\mathsf{Fired}(\phi, \Pi) = 0$*, then* $\mathsf{Validity}(\phi, \Pi) = 0$*, otherwise*

$$\mathsf{Validity}(\phi, \Pi) = \frac{\mathsf{Confirmed}(\phi, \Pi)}{\mathsf{Fired}(\phi, \Pi)}$$

Example 4.1. *Let* Π *be the set of reports* $\{\rho_1, .., \rho_{10}\}$ *defined as follows.*

$$\begin{array}{ll} \rho_1 = \{p(a), q(b), t(a, b)\} & \rho_2 = \{p(c), q(b), r(c, b)\} \\ \rho_3 = \{p(a), q(b), r(a, b)\} & \rho_4 = \{p(c), q(c), r(c, c)\} \\ \rho_5 = \{p(e), s(b), r(e, b)\} & \rho_6 = \{p(a), q(d), r(a, d)\} \\ \rho_7 = \{p(a), s(b), r(a, b)\} & \rho_8 = \{p(b), q(b), r(b, b)\} \\ \rho_9 = \{p(a), q(c), r(a, c)\} & \rho_{10} = \{p(a), q(b), \neg r(a, b)\} \end{array}$$

If $\phi = \forall x, y(p(x) \wedge q(y) \rightarrow r(x, y))$*, then*

$$\begin{array}{l} \mathsf{Fired}(\phi, \Pi) = |\{\rho_1, \rho_2, \rho_3, \rho_4, \rho_6, \rho_8, \rho_9, \rho_{10}\}| = 8 \\ \mathsf{Attacked}(\phi, \Pi) = |\{\rho_{10}\}| = 1 \\ \mathsf{Violated}(\phi, \Pi) = |\{\rho_{10}\}| = 1 \\ \mathsf{Confirmed}(\phi, \Pi) = |\{\rho_2, \rho_3, \rho_4, \rho_6, \rho_8, \rho_9\}| = 6 \end{array}$$

So $\mathsf{Accuracy}(\phi, \Pi) = 0.875$ *and* $\mathsf{Validity}(\phi, \Pi) = 0.750$

The $\mathsf{Accuracy}$ and $\mathsf{Validity}$ functions each return a value in the $[0, 1]$ range. These values can be used to discriminate between a number of expectations that have been fired by a report. Validity is a more skeptical measure than accuracy. They can be used as "upper" and "lower" bounds on quality of expectations.

Proposition 4.3. *For* $\phi \in \mathsf{Expectations}(P, X, n)$*, and for* $\Pi \subseteq \mathsf{Reports}(P, C, n)$*, if all* $\rho \in \Pi$ *are complete reports, then* $\mathsf{Validity}(\phi, \Pi) = \mathsf{Accuracy}(\phi, \Pi)$*, otherwise* $\mathsf{Validity}(\phi, \Pi) \leq \mathsf{Accuracy}(\phi, \Pi)$.

Proof. If all $\rho \in \Pi$ are complete reports, then $\mathsf{Fired}(\phi, \Pi) = \mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi)$. So $\mathsf{Validity}(\phi, \Pi) = \frac{\mathsf{Confirmed}(\phi, \Pi)}{\mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi)}$, and $\mathsf{Accuracy}(\phi, \Pi) = 1 - \frac{\mathsf{Violated}(\phi, \Pi)}{\mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi)}$. Hence, $\mathsf{Validity}(\phi, \Pi) = \mathsf{Accuracy}(\phi, \Pi)$. If there is a $\rho \in \Pi$ that is not a complete report, then $\mathsf{Fired}(\phi, \Pi) \geq \mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi)$. So we get,

$$\mathsf{Validity}(\phi, \Pi) = \frac{\mathsf{Confirmed}(\phi, \Pi)}{\mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi) + \mathsf{Unaffected}(\phi, \Pi)}$$

$$\mathsf{Accuracy}(\phi, \Pi) = 1 - \frac{\mathsf{Violated}(\phi, \Pi)}{\mathsf{Violated}(\phi, \Pi) + \mathsf{Confirmed}(\phi, \Pi) + \mathsf{Unaffected}(\phi, \Pi)}$$

Hence, in this case $\mathsf{Validity}(\phi, \Pi) \leq \mathsf{Accuracy}(\phi, \Pi)$. □

We now return to the types of expectation introduced in Definition 2.5. These four types of expectation are notable in that their accuracy and validity values can be determined entirely from their syntax as demonstrated in the following result. From this, it is clear that each of these four types of expectation offers no useful information.

Proposition 4.4. *For $\phi \in \mathsf{Expectations}(P, X, n)$, and for $\Pi \subseteq \mathsf{Reports}(P, C, n)$, such that $\Pi \neq \emptyset$.*

1. *If ϕ is self-defeating, then $\mathsf{Accuracy}(\phi, \Pi) = 0$ and $\mathsf{Validity}(\phi, \Pi) = 0$*
2. *If ϕ is self-reinforcing, and $\mathsf{Fired}(\phi, \Pi) \neq 0$, then $\mathsf{Accuracy}(\phi, \Pi) = 1$ and $\mathsf{Validity}(\phi, \Pi) = 1$*
3. *If ϕ is non-firing, $\mathsf{Accuracy}(\phi, \Pi) = 0$ and $\mathsf{Validity}(\phi, \Pi) = 0$*
4. *If ϕ is non-commiting, and $\mathsf{Fired}(\phi, \Pi) \neq 0$, then $\mathsf{Accuracy}(\phi, \Pi) = 1$ and $\mathsf{Validity}(\phi, \Pi) = 1$*

Proof. If ϕ is self-defeating, then $\{\mathsf{Antecedent}(\phi)\} \vdash \mathsf{ContraConsequent}(\phi)$. Therefore, $\mathsf{Violated}(\phi, \Pi)$ = $\mathsf{Fired}(\phi, \Pi)$, and $\mathsf{Confirmed}(\phi, \Pi) = 0$. Hence, $\mathsf{Accuracy}(\phi, \Pi) = 0$ and $\mathsf{Validity}(\phi, \Pi) = 0$. If ϕ is self-reinforcing, then $\{\mathsf{Antecedent}(\phi)\} \vdash \mathsf{Consequent}(\phi)$. Therefore, $\mathsf{Confirmed}(\phi, \Pi)$ = $\mathsf{Fired}(\phi, \Pi)$, and $\mathsf{Violated}(\phi, \Pi) = 0$. So from $\mathsf{Fired}(\phi, \Pi) \neq 0$, we get $\mathsf{Accuracy}(\phi, \Pi) = 1$ and $\mathsf{Validity}(\phi, \Pi) = 1$. If ϕ is non-firing, then $\{\mathsf{Antecedent}(\phi)\} \vdash \perp$. So, $\mathsf{Fired}(\phi, \Pi) = 0$. Hence, $\mathsf{Accuracy}(\phi, \Pi) = 0$ and $\mathsf{Validity}(\phi, \Pi) = 0$. If ϕ is non-commiting, then $\top \vdash \mathsf{Consequent}(\phi)$. Therefore, $\mathsf{Confirmed}(\phi, \Pi)$ = $\mathsf{Fired}(\phi, \Pi)$, and $\mathsf{Violated}(\phi, \Pi) = 0$. Hence, from $\mathsf{Fired}(\phi, \Pi) \neq 0$, we get $\mathsf{Accuracy}(\phi, \Pi) = 1$ and $\mathsf{Validity}(\phi, \Pi) = 1$. □

The accuracy and validity values are related to the extreme positions in the expectation ordering as follows.

Proposition 4.5. *For $\phi, \psi \in \mathsf{Expectations}(P, X, n)$, and for $\Pi \subseteq \mathsf{Reports}(P, C, n)$, such that $\Pi \neq \emptyset$.*

1. *If ϕ is maximal in $\geq_a$, then $\mathsf{Accuracy}(\phi, \Pi) = 0$ and $\mathsf{Validity}(\phi, \Pi) = 0$*
2. *If ϕ is minimal in $\geq_c$, and $\mathsf{Fired}(\phi, \Pi) \neq 0$, then $\mathsf{Accuracy}(\phi, \Pi) = 1$ and $\mathsf{Validity}(\phi, \Pi) = 1$*

Proof. If ϕ is maximal in $\geq_a$, then ϕ is non-firing, and so $\mathsf{Accuracy}(\phi, \Pi) = 0$ and $\mathsf{Validity}(\phi, \Pi) = 0$. If ϕ is minimal in $\geq_c$, then ϕ is non-commiting, and so $\mathsf{Accuracy}(\phi, \Pi) = 1$ and $\mathsf{Validity}(\phi, \Pi) = 1$. □

Expectations that have logically equivalent antecedents and logically equivalent consequents, have the same accuracy and the same validity. Whereas expectations that differ in consequent but not in antecedent are such that the expectation with the weakest consequent will be the most accurate and the most valid, as the following result demonstrates.

Proposition 4.6. *For $\phi, \psi \in \mathsf{Expectations}(P, X, n)$, and for $\Pi \subseteq \mathsf{Reports}(P, C, n)$. If $\phi =_a \psi$ and $\phi \geq_c \psi$, then $\mathsf{Accuracy}(\phi, \Pi) \leq \mathsf{Accuracy}(\psi, \Pi)$ and $\mathsf{Validity}(\phi, \Pi) \leq \mathsf{Validity}(\psi, \Pi)$.*

Proof. If $\phi =_a \psi$, then $\mathsf{Fired}(\phi, \Pi) = \mathsf{Fired}(\psi, \Pi)$. If $\phi \geq_c \psi$, then $\mathsf{Violated}(\phi, \Pi) \geq \mathsf{Violated}(\psi, \Pi)$ and $\mathsf{Confirmed}(\phi, \Pi) \leq \mathsf{Confirmed}(\psi, \Pi)$. Hence, the results follow by the definitions for Accuracy and Validity. □

Note, if $\phi \geq_a \psi$ and $\phi =_c \psi$, then it is not necessarily the case that $\mathsf{Accuracy}(\phi, \Pi) \leq \mathsf{Accuracy}(\psi, \Pi)$ nor $\mathsf{Accuracy}(\psi, \Pi) \leq \mathsf{Accuracy}(\phi, \Pi)$. Also it is not necessarily the case that $\mathsf{Validity}(\phi, \Pi) \leq \mathsf{Validity}(\psi, \Pi)$ nor $\mathsf{Validity}(\psi, \Pi) \leq \mathsf{Validity}(\phi, \Pi)$.

Logically equivalent expectations may have different accuracy values and different validity values. For example, the expectations $\forall x(a(x) \land b(x) \rightarrow c(x))$ and $\forall x(a(x) \rightarrow \neg b(x) \lor c(x))$ are equivalent but the different antecedents and consequents can lead to different values for the Fired function, and different values for the Confirmed function, for the two expectations, though they will have the same value for the Violated function.

Proposition 4.7. *For* $\Pi \subseteq \mathsf{Reports}(P, C, n)$*, and for* $\phi, \psi \in \mathsf{Expectations}(P, X, n)$*, if* $\phi \geq_e \psi$*, and* ϕ *is logically equivalent to* ψ*, then* $\mathsf{Accuracy}(\phi, \Pi) \leq \mathsf{Accuracy}(\psi, \Pi)$.

Proof. If ϕ is logically equivalent to ψ, then $\mathsf{Violated}(\phi, \Pi) = \mathsf{Violated}(\psi, \Pi)$. Also if $\phi \geq_e \psi$, then $\mathsf{Fired}(\phi, \Pi) \leq \mathsf{Fired}(\psi, \Pi)$. Therefore, $\mathsf{Accuracy}(\phi, \Pi) \leq \mathsf{Accuracy}(\psi, \Pi)$. □

The expectation ordering therefore allows for the selection of the most accurate expectations from a class of equivalent expectations.

So far we have not discussed where expectations come from. If we assume the expectations come from a machine learning algorithm, we could obtain the accuracy and validity values indirectly from the data used by the machine learning. Alternatively, we can generate a set of expectations from a set of reports. We investigate this latter option in Section 6.

5 Best expectations

If expectations are to be used to identify interesting information in reports, then in practice the set of expectations should be small enough to be searched effectively on receipt of a report. For this, we want to select expectations according to some criteria. In this section, we consider how we can select the "best" expectations based on the syntax and relative accuracy and validity of the expectations. The overall aim of this section is to address the question raised at the end of Section 3: How do we select a subset $\Delta \subseteq \mathsf{Expectations}(P, X, n)$ for comparison with a ρ (i.e. for use in Definition 3.7)? The idea here is that Δ is chosen in advance, and then as each report ρ arrives the set $\mathsf{ViolatedExpectations}(\Delta, \rho)$ is evaluated. Next we motivate why, in general, we want $\Delta \subset \mathsf{Expectations}(P, X, n)$ to hold.

When a report violates an expectation ϕ, then a user may be interested to know. But the user is unlikely to want to know that the report violates all the expectations that are logically equivalent to ϕ. From the previous section, it is apparent that given an arbitrarily chosen set of expectations Δ, there may be a number of logically equivalent expectations and these may have differing accuracy. Since logically equivalent expectations provide redundant information, it would be preferable to have just one of the logically equivalent expectations as a representative of its logical equivalents. In other words, ideally we should select Δ to be an expectation class, though this is not always possible.

However, this does raise the question of how we choose amongst the logically equivalent expectations for inclusion in the chosen expectation class. Now recall the main role of measuring accuracy is to differentiate between different expectations that have been violated by a given report. In other words, if $\rho \in \mathsf{Reports}(P, C, n)$, and $\Delta \subseteq \mathsf{Expectations}(P, X, n)$, then $\mathsf{ViolatedExpectations}(\Delta, \rho)$ may be a large set of expectations. We therefore need to differentiate between them.

To illustrate this discussion, consider the set of expectations $\{\phi_0, \phi_1, \phi_2, ..., \phi_n\}$ which are defined below. These expectations are logically equivalent.

$$
\begin{array}{ll}
\phi_0 & = \forall \bar{x}(\top \rightarrow \alpha_1 \vee ... \vee \alpha_n) \\
\phi_1 & = \forall \bar{x}(\neg\alpha_1 \rightarrow \alpha_2 \vee ... \vee \alpha_n) \\
\phi_2 & = \forall \bar{x}(\neg\alpha_1 \wedge \neg\alpha_2 \rightarrow \alpha_3 \vee ... \vee \alpha_n) \\
\phi_3 & = \forall \bar{x}(\neg\alpha_1 \wedge \neg\alpha_2 \wedge \neg\alpha_3 \rightarrow \alpha_4 \vee ... \vee \alpha_n) \\
: & \\
: & \\
\phi_{n-1} & = \forall \bar{x}(\neg\alpha_1 \wedge ... \wedge \neg\alpha_{n-1} \rightarrow \alpha_n) \\
\phi_n & = \forall \bar{x}(\neg\alpha_1 \wedge ... \wedge \neg\alpha_n \rightarrow \bot)
\end{array}
$$

Following Proposition 4.7, we have the following.

$$\mathsf{Accuracy}(\phi_0, \Pi) \geq \mathsf{Accuracy}(\phi_1, \Pi) \geq \mathsf{Accuracy}(\phi_2, \Pi) \geq .... \geq \mathsf{Accuracy}(\phi_n, \Pi)$$

For all ϕ_i, where $1 \leq i \leq n$, the value for $\mathsf{Violated}(\phi_i, \Pi)$ is the same. The difference in the accuracy value comes from the difference in the value for the fired value.

$$\mathsf{Fired}(\phi_0, \Pi) \geq \mathsf{Fired}(\phi_1, \Pi) \geq \mathsf{Fired}(\phi_2, \Pi) \geq .... \geq \mathsf{Fired}(\phi_n, \Pi)$$

At the one extreme, $\mathsf{Fired}(\phi_0, \Pi) = |\Pi|$, and furthermore, for any subsequent report, expectations of the form ϕ_0 will always be fired, because for any report ρ, we have $\rho \vdash \mathsf{Antecedent}(\phi_0)$. At the other extreme, since the antecedent of ϕ_n is the most specific out of these expectations, the firing of this expectation is the one that is most dependent on the context of the report. However, we cannot choose ϕ_n because $\mathsf{Accuracy}(\phi_n, \Pi) = 0$ for any Π. Of the expectations with non-zero accuracy, the expectation with the most specific context is of the form of ϕ_{n-1}.

Let us now compare ϕ_0 and ϕ_{n-1}. Suppose we have adopted a threshold for accuracy τ. So if the accuracy of a violated expectation is below τ, we do not consider it further. Adopting a threshold is a straightforward, and justifiable, way of reducing the number of violated expectations we have to consider. Now suppose $\mathsf{Accuracy}(\phi_0, \Pi) > \tau$ and $\mathsf{Accuracy}(\phi_{n-1}, \Pi) > \tau$. So neither ϕ_0 nor ϕ_{n-1} is rejected, but because of the syntax of ϕ_{n-1}, the Fired value of ϕ_{n-1} is likely to be lower and so it more unlikely *a priori* for ϕ_{n-1} to exceed the threshold, and so in this sense, ϕ_{n-1} is a more forceful statement, and we can suppose a more interesting statement. In other words, implicitly ϕ_{n-1} has a higher hurdle to jump over to exceed the threshold, and so if it does exceed the threshold, it is more exceptional. On this basis, we prefer the syntactic form of ϕ_{n-1}.

So, in general, our solution is to select expectations that conform to a particular syntactic form that we call key expectations and we formalize this next.

Definition 5.1. *The set of* **key expectations***, is defined as follows.*

$$\begin{aligned}\mathsf{KeyExpectations}(P, X, n) = \{\forall \bar{x}\ (\alpha_1 \wedge .. \wedge \alpha_n \rightarrow \beta) \in \mathsf{Expectations}(P, X, n) \mid \\ \alpha_1, .., \alpha_n, \beta \in \mathsf{Literals}(P, X, n) \textit{ and } \alpha_1 \neq ... \neq \alpha_n \neq \beta\}\end{aligned}$$

The condition $\alpha_1 \neq ... \neq \alpha_n \neq \beta$ in the definition of key expectations ensures that the following proposition holds.

Proposition 5.1. *For all $\phi \in \mathsf{KeyExpectations}(P, X, n)$, ϕ is not self-reinforcing and ϕ is not non-commiting.*

Restricting the choice of expectations to just key expectations allows us to eliminate many syntactic variants that are logically equivalent. But it does not eliminate all logically equivalent expectations: For each key expectation, there may be a number of logically equivalent expectations that are also key expectations. These logically equivalent key expectations are obtained by contraposition. Furthermore, these contrapositive variations on a key expectation may also have differing accuracy values. So for example, it is not necessarily the case that $\mathsf{Accuracy}(\forall \bar{x}(\alpha \rightarrow \beta), \Pi)$ = $\mathsf{Accuracy}(\forall \bar{x}(\neg\beta \rightarrow \neg\alpha), \Pi)$. A solution to this second question is to choose the contrapositive variants of a key expectation with highest accuracy, or the highest validity, or preferably both.

We summarise this in the following definition of best expectation classes. Furthermore, we incorporate a threshold for accuracy τ_a and a threshold for validity τ_v to further limit the violated expectations that we consider per report.

Definition 5.2. *For* $\mathsf{Expectations}(P, X, n)$*, the set of* **best expectations** *is defined as follows, where* $\Pi \subseteq \mathsf{Reports}(P, X, n)$*, and* τ_a *is the threshold for accuracy, and* τ_v *is the threshold for validity.*

$$\begin{array}{rl} \mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v) = \{\phi \in \mathsf{KeyExpectation}(P, X, n) \mid & \\ \mathsf{Accuracy}(\phi, \Pi) > \tau_a \textit{ and } \mathsf{Validity}(\phi, \Pi) > \tau_v & \\ \textit{and there is no } \psi \in \mathsf{KeyExpectations}(P, X, n) & \\ \textit{such that } \psi \textit{ is logically equivalent to } \phi & \\ \textit{and } \mathsf{Accuracy}(\psi, \Pi) > \mathsf{Accuracy}(\phi, \Pi) & \\ \textit{and } \mathsf{Validity}(\psi, \Pi) > \mathsf{Validity}(\phi, \Pi)\} & \end{array}$$

Ideally, the set of expectations obtained by $\mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v)$ *is the set of expectations to be used in Definition 3.7.*

By using only key expectations, we are not loosing any violations modulo logical equivalence. But we are restricting consideration to the more forceful expectations. These are captured as the set of best expectations. Though, it is not necessarily the case that $\mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v)$ is an expectation class.

Proposition 5.2. *For any expectation class* Λ*, if* $\phi \in \Lambda$*, there is at most two expectations* ϕ_1 *and* ϕ_2 *in* $\mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v)$ *such that* ϕ_2 *is logically equivalent to* ϕ_1 *and* ϕ_1 *is logically equivalent to* ϕ_2*.*

Proof. There are three cases to consider for $\phi \in \mathsf{KeyExpectations}(P, X, n)$: (1) It may be that ϕ and all its logically equivalent key expectations have an accuracy and validity below the respective thresholds; (2) It may be there is a ϕ such that ϕ has accuracy and validity above the respective thresholds, and above all its logically equivalent expectations; or (3) It may be that there are $\phi_1, \phi_2 \in \mathsf{KeyExpectations}(P, X, n)$ with accuracy and validity above the respective thresholds and ϕ_1 is logically equivalent to ϕ_2 and ϕ_1 is maximal for accuracy amongst its logically equivalent key expectations and ϕ_2 is maximal for validity amongst its logically equivalent key expectations. In this case, both ϕ_1 and ϕ_2 are in the set of best expectations. □

So whilst it is not necessarily the case that $\mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v)$ is an expectation class, it can be regarded as a good approximation of an expectation class. Clearly, there are alternative criteria for selecting expectations, and the choice of criteria depends on the application. Elsewhere, we have considered taking the coverage of the expectation into account, which appears to be an important criteria in analysing news reports [Byr05].

6 Viable expectations

Given a set $\mathsf{FreeAtoms}(P, X, n)$, we can potentially construct a set of best expectations. However for all but a small set of atoms, we would need to construct an enormous set of expectations, and often this would actually be unnecessary. In this section, we consider contructing best expectations up to a maximum number of literals, and then using some of the general results considered in Section 4 to infer a lower bound on the accuracy and validity for expectations that involve more literals. The overall aim therefore is to render the approach of generating and maintaining expectations more viable.

Candidates for best expectations can be generated from sets of reports, and tested against other reports to determine accuracy and validity. For a set of complete reports Π, the least generalization of Π can be generated by an induction algorithm such as Plotkin's algorithm [Plo70].

Definition 6.1. *Let* Π *be a set of complete reports. An expectation* ϕ *is a least generalization of* Π *iff for each* $\rho \in \Pi$*, there is an* $\alpha \in \rho$*, such that* $\{\phi\} \not\vdash \alpha$ *and* $\{\phi\} \cup \rho \setminus \{\alpha\} \vdash \alpha$*.*

So an expectation is a least generalization of a set of reports Π if it implies part of each report in Π given the rest of that report.

Example 6.1. *Consider the following set of reports.*

$$\rho_1 = \{p(a,b), q(a,c), r(a,c)\}$$
$$\rho_2 = \{p(a,c), q(a,b), r(a,b)\}$$
$$\rho_3 = \{p(d,a), q(d,c), r(d,c)\}$$

Least generalizations of $\{\rho_1, \rho_2, \rho_3\}$ *that include the following.*

$$\forall x, y\ (p(x,y) \land q(y,z) \rightarrow r(x,z))$$
$$\forall x, y\ (p(x,y) \land r(y,z) \rightarrow q(x,z))$$
$$\forall x, y\ (r(x,y) \land q(y,z) \rightarrow p(x,z))$$

The candidate expectations are obtained from the least generalizations of facts taken from the reports.

Definition 6.2. *The set of* **candidate expectations** *is defined as follows, where* $\Pi \subseteq \mathsf{Reports}(P, X, n)$, $k \in \mathbb{N}$, *and for all* $\rho \in \Pi$, $|\rho| = k$.

$$\begin{aligned}\mathsf{CandidateExpectations}(\Pi, k) = \{&\forall \bar{x}(\alpha_1 \land .. \land \alpha_i \rightarrow \alpha_{i+1}) \mid \\ &\textit{and } \forall \bar{x}(\alpha_1 \land .. \land \alpha_i \rightarrow \alpha_{i+1}) \textit{ is a least generalization of } \Pi \\ &\textit{and } i+1 \leq k\}\end{aligned}$$

The candidate expectations obtained are then evaluated for accuracy and validity.

Definition 6.3. *The set of* **generated expectations** *is defined as follows, where* $\Pi \subseteq \mathsf{Reports}(P, X, n)$, τ_a *is the threshold for accuracy,* τ_v *is the threshold for validity, and* $k \in \mathbb{N}$.

$$\begin{aligned}&\mathsf{GeneratedExpectations}(P, X, n, \Pi, \tau_a, \tau_v, k) \\ &\quad = \{\phi \in \mathsf{CandidateExpectations}(\Pi, k) \mid \mathsf{Accuracy}(\phi, \Pi) > \tau_a \textit{ and } \mathsf{Validity}(\phi, \Pi) > \tau_v \\ &\qquad \textit{and there is no } \psi \in \mathsf{CandidateExpectations}(\Pi, k) \\ &\qquad\quad \textit{such that } \psi \textit{ is logically equivalent to } \phi \\ &\qquad\quad \textit{and } \mathsf{Accuracy}(\psi, \Pi) > \mathsf{Accuracy}(\phi, \Pi) \\ &\qquad\quad \textit{and } \mathsf{Validity}(\psi, \Pi) > \mathsf{Validity}(\phi, \Pi)\}\end{aligned}$$

The generated expectations do not include the problematical expectations that we considered in Section 2: Because self-defeating expectations and non-firing expectations have zero accuracy, they do not appear in the set of generated expectations; because the generated expectations are key expectations, they are not non-commiting; and because the generated expectations are least generalizations, they are not self-reinforcing.

Proposition 6.1. *If* $\phi \in \mathsf{GeneratedExpectations}(P, X, n, \Pi, \tau_a, \tau_v, k)$, *then* ϕ *is not self-defeating, and* ϕ *is not non-firing, and* ϕ *is not non-commiting, and* ϕ *is not self-reinforcing.*

So $\mathsf{GeneratedExpectations}(P, X, n, \Pi, \tau_a, \tau_v, k)$ is a set of key expectations each of which have up to k literals. Suppose ϕ is an expectation in this set, and suppose we have a report ρ such that ϕ is violated by ρ. However, if $|\rho| > k$, it is not the case that ϕ is minimally violated by ρ. In this case, we can appeal to Proposition 4.6. Using this, we can find a non-key expectation ψ such that ψ is minimally violated by ρ. Suppose ϕ is of the form $\alpha_1 \land .. \land \alpha_{k-1} \rightarrow \alpha_k$, then ψ is of the form $\alpha_1 \land .. \land \alpha_{k-1} \rightarrow \alpha_k \lor \beta_1 \lor .. \lor \beta_j$ where $\beta_1, .., \beta_j$ are literals negated by ρ. To get a lower bound on the accuracy for ψ, we can use the accuracy of the expectation ϕ.

In general, $\mathsf{GeneratedExpectations}(P, X, n, \Pi, \tau_a, \tau_v, k) \subset \mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v)$ does not hold, since the expectations are restricted to those that can be generated by the least generalization of the reports. Nevertheless, $\mathsf{GeneratedExpectations}(P, X, n, \Pi, \tau_a, \tau_v, k)$ can be regarded as a good approximation of $\mathsf{BestExpectations}(P, X, n, \Pi, \tau_a, \tau_v)$ for the expectations containing up to k literals.

The definition for $\mathsf{GeneratedExpectations}(P, X, n, \Pi, \tau_a, \tau_v, k)$ provides a simple way of generating viable expectations. Some refinements of this have been developed [Byr05]. As an alternative, this definition can be used as a specification of the required expectations, and a general machine learning approach such as inductive logic programming [Mug99a, Mug99b] can be used.

7 Analysing expectations when merging information

In information fusion the aim is to take various sources of information and form an aggregation of the information that reflects what is in common, and what is different, between the reports.

The information in common may be regarded as reflecting redundancy. For example, if we have a number of news reports that all agree on the facts, then we could just take one of the news reports to represent all the other news reports. In this way, we would cut out repetition. However, information in common may also be regarded as reflecting confirmation. So if a number of news reports agree on some fact, then the greater the number of reports that agree, the greater the confidence we have that this fact is indeed true.

The information that is different between the reports, may be information that some reports have that other reports do not have. Merging the reports can then take the union of such information. However, the information that is different may also be in conflict. For example one report may say α and the other may say $\neg\alpha$. In this case, we need to consider the conflicts in the aggregation process. This may involve some conflict resolution strategy, and/or taking the relative confidence of the aggregated information. In the merged report, this confidence can be represented either qualitatively (e.g. all the reports agree on the fact, or the majority of the reports agree on the fact, or the majority of the reports do not negate the facts, etc) or quantitatively using a statistical evaluation.

From the above discussion there are clearly a number of factors to consider in merging, and there are a number of logic-based proposals that we may consider for supporting this. For some possibilities consider [BKMS92, BCVB01, CGL$^+$98, CM01, CH97, Hun02a, KP98, LS98, PHG$^+$99]. The emphasis in this section is purely on showing how one factor that we incorporate in the merged report is information on the reports that violate some expectations. Often when merging information, there are some expectations about the reports. So if the reports meet the expectations, for example the values fall between an upper and lower bound, then the reports can be merged. Reports that do not meet the expectations are either rejected, or if too many violate the expectation, then the merging is aborted. Yet reports that violate an expectation are perhaps the most interesting reports in diverse applications such as intelligence gathering or biomedical research.

In the rest of this section, we use an extended example to illustrate how we may use expectations in fusion. In the example, Π is the set of reports used to generate the expectations. Π may be a substantial repository collected over a protracted perion. In contrast, Γ is a smaller set of reports that are of particular interest. They may, for example, have been collected over a shorter time period, perhaps recently.

So first we consider the set of reports Γ with the aim of producing a summary of them. Suppose the set of reports is $\Gamma = \{\rho_1, .., \rho_n\}$, where $|\Gamma| = 1000$. Each report provides information pertaining to an individual patient in a drug trial. It includes information on the age, placebo or drug taken, symptoms, and degree of improvement in the patients condition. We give a few simple examples from Γ below.

$$\rho_1 = \{treatment(MrJones, placebo), age(MrJones, 57), \\ symptom(MrJones, TypeA), improvement(MrJones, 0\%)\}$$

$$\rho_2 = \{treatment(MrSmith, placebo), age(MrSmith, 52), \\ symptom(MrSmith, TypeB), improvement(MrSmith, 0\%)\}$$

$$\rho_3 = \{treatment(MrPatel, drug), age(MrPatel, 67), \\ symptom(MrPatel, TypeA), improvement(MrPatel, 50\%)\}$$

$$\rho_4 = \{treatment(MrSingh, drug), age(MrSingh, 59), \\ symptom(MrSingh, TypeB), improvement(MrSingh, 100\%)\}$$

In addition, we need to consider a set of key expectations Δ where

$$\mathsf{ViolatedExpectations}(\Delta, \Gamma) = \{\phi_1, \phi_2, \phi_3, \phi_4\}$$

and

$$\phi_1 = treatment(X, placebo) \wedge symptom(X, TypeA) \rightarrow improvement(X, 0\%)$$
$$\phi_2 = treatment(X, placebo) \wedge symptom(X, TypeB) \rightarrow improvement(X, 0\%)$$
$$\phi_3 = treatment(X, drug) \wedge symptom(X, TypeA) \rightarrow improvement(X, 50\%)$$
$$\phi_4 = treatment(X, drug) \wedge symptom(X, TypeB) \rightarrow improvement(X, 100\%)$$

Now we need to consider what to do with the expectation violation distribution presented in Table 1. We are merging information, and so we may treat a violation of an expectation as

- violation indicates report contains something exceptional
- violation indicates expectation is unreliable

To select amongst these two options, we need to consider the "difference" between the long term past (Π) used to generate the accuracy of ϕ and the recent history (Γ). A small difference indicates that the reports that violate contain exceptional information, whereas a large difference indicates that the expectation is unreliable. To formalise, we assume a threshold σ, so if the difference is less than σ, then we treat the expectation as reliable, and treat the violating reports as exceptional.

$$(|\mathsf{Accuracy}(\phi, \Pi) - \mathsf{Accuracy}(\phi, \Gamma)|) < \sigma$$

We can view each of $\mathsf{Accuracy}(\phi, \Pi)$ and $\mathsf{Accuracy}(\phi, \Gamma)$ as a conditional probability assignment for the conditional probability statement defined by ϕ. If Π and Γ are sufficiently large sets of reports selected by some unbiased process, then we would expect the difference between $\mathsf{Accuracy}(\phi, \Pi)$ and $\mathsf{Accuracy}(\phi, \Gamma)$ to be relatively small. Indeed, we can attempt to estimate the probability that the difference between $|\mathsf{Accuracy}(\phi, \Pi)|$ and $|\mathsf{Accuracy}(\phi, \Gamma)|$ is less than σ. Possible approaches to estimating this probability include Bayesian and Pseudo-Bayesian techniques [CHS93].

Returning to the example, we see the accuracy values for the expectations ϕ_1 to ϕ_4 with the set of reports Π and Γ as follows. We assume the figures in the left column (below) have been given for some Π. For the right column (below), we have used the data in Table 1 for the relevant information we require about Γ. For example, $\mathsf{Accuracy}(\phi_1, \Gamma)$ is $1 - \frac{\mathsf{Violated}(\phi_1, \Gamma)}{\mathsf{Fired}(\phi_1, \Gamma)}$ which can be calculated as $1 - \frac{5}{995}$ since from the table, we get $\mathsf{Fired}(\phi_1, \Gamma)$ as the sum of all the reports considered in the table (i.e. 995 is the sum of the right-hand column), and we get $\mathsf{Violated}(\phi_1, \Gamma)$ as the sum of the reports that violate ϕ_1 in the table (i.e. 5 is the sum of the reports that violate ϕ_1),

$$
\begin{array}{ll}
\mathsf{Accuracy}(\phi_1, \Pi) = 0.997 & \mathsf{Accuracy}(\phi_1, \Gamma) = 0.995 \\
\mathsf{Accuracy}(\phi_2, \Pi) = 0.998 & \mathsf{Accuracy}(\phi_2, \Gamma) = 0.998 \\
\mathsf{Accuracy}(\phi_3, \Pi) = 0.938 & \mathsf{Accuracy}(\phi_3, \Gamma) = 0.983 \\
\mathsf{Accuracy}(\phi_4, \Pi) = 0.925 & \mathsf{Accuracy}(\phi_4, \Gamma) = 0.797
\end{array}
$$

So if we let σ be 0.05, then we would regard ϕ_1, ϕ_2, and ϕ_3 as reliable, and ϕ_4 as unreliable. Any reports that violate ϕ_1, ϕ_2, or ϕ_3 are treated as exceptions, and are interesting as such. Whereas ϕ_4 is treated as unreliable, and is treated as uninteresting as such.

We now turn to the information that may be presented to the user. To illustrate some possibilities, we represent the merged information as an XML document. Let us suppose the basic structure for the merged report is as follow.

⟨ClinicalTrial⟩
    ⟨AverageAge⟩ · · · ⟨/AverageAge⟩
    ⟨Period⟩ · · · ⟨/Period⟩
    ⟨TrialCentre⟩ · · · ⟨/TrialCentre⟩
    ⟨Drug⟩ · · · ⟨/Drug⟩
    :
    :
⟨/ClinincalTrial⟩

When providing the merged report, one aspect of the merged report is a summary of the expectations violated. This information can be represented as part of the XML document as follows.

⟨ExpectationAnalysis⟩
    ⟨ViolationCount expectation = "ϕ_1"⟩5⟨/ViolationCount⟩
    ⟨ViolationCount expectation = "ϕ_2"⟩2⟨/ViolationCount⟩
    ⟨ViolationCount expectation = "ϕ_3"⟩16⟨/ViolationCount⟩
    ⟨ViolationCount expectation = "ϕ_4"⟩202⟨/ViolationCount⟩
⟨/ExpectationAnalysis⟩

We may also want to give details about what is in common between the reports that violate, or even provide a list of the reports that violate. This information can also be represented as part of the XML document as follows.

⟨ExpectationAnalysis⟩
    ⟨ViolatingReportsSummary expectation = "ϕ_1"⟩
        ⟨AverageAge⟩ · · · ⟨/AverageAge⟩
        ⟨ProportionTreatment = "Placebo"⟩ · · · ⟨/Proportion⟩
        ⟨ProportionTreatment = "Drug"⟩ · · · ⟨/Proportion⟩
        ⟨ProportionSymptom = "TypeA"⟩ · · · ⟨/Proportion⟩
        ⟨ProportionSymptom = "TypeB"⟩ · · · ⟨/Proportion⟩
    ⟨/ViolatingReportsSummary⟩
⟨/ExpectationAnalysis⟩

The use of expectations seems particularly appropriate in merging scientific information. Not only is it often feasible to have sufficient information with which to construct expectations. But also scientists often have expectations explicitly represented, in the form of hypotheses or conjectures, when designing and

ϕ_1	ϕ_2	ϕ_3	ϕ_4	Number of reports
violated	violated	violated	violated	1
violated	violated	violated	unviolated	0
violated	violated	unviolated	violated	0
violated	violated	unviolated	unviolated	1
violated	unviolated	violated	violated	3
violated	unviolated	violated	unviolated	0
violated	unviolated	unviolated	violated	0
violated	unviolated	unviolated	unviolated	0
unviolated	violated	violated	violated	0
unviolated	violated	violated	unviolated	0
unviolated	violated	unviolated	violated	0
unviolated	violated	unviolated	unviolated	0
unviolated	unviolated	violated	violated	5
unviolated	unviolated	violated	unviolated	7
unviolated	unviolated	unviolated	violated	193
unviolated	unviolated	unviolated	unviolated	785

Table 1: An example of an expectation violation distribution obtained from a set of reports: Number of reports that violate particular combinations of the expectations ϕ_1 to ϕ_4. Each row gives a particular combination, so row 1 shows that 1 report violates all the expectations, whereas row 15 shows that 193 reports violate just ϕ_4, and row 16 shows that 785 reports violate no expectation. So this distribution provides an abstraction of a particular set of reports.

conducting experiments. Hence it seems valuable to use such expectations in a logical form in the merging process as outlined in this paper.

In suggesting that a merged report can be constructed from the set of reports, as outlined in this section, we are circumventing many key issues. These issues fall outside the scope of this paper, and we see the proposals in this paper being appropriate in various logic-based approaches to merging such as [Hun02a, HS03, HS04]. Here, we only want to show how evaluating violations of expectations can be an important part of finding exceptional (and hence interesting) information in the overall task of merging information.

8 Conclusions

Unexpected information is often more useful than expected information to the reader. Identifying such information is beyond the current capabilities of traditional information systems technologies. We look for interesting new information based on the expectations we have about the world. If a new information violates an expectation, this indicates the new information is interesting. Furthermore, if we represent each expectation as a classical logic formula, and the information in a new report as a set of ground literals of classical logic, then the violation of an expectation by this new information can be characterized as a type of logical inconsistency.

Both an expectation and intregrity constraint can be represented by an expectation in classical logic, and every time we get new information, we compare it with the expectation. However, with an integrity constraints, we are primarily seeking information that is consistent with the expectation. In constrast, with an expectation, we are primarily seeking information that is inconsistent expectation. This means there is a clear line between the role, and nature, of expectations and of integrity constraints. Nonetheless, there are some important commonalities.

In this paper, we have presented a framework for representing and reasoning with expectations. This has included consideration of the syntax of expectations, definitions for accuracy and validity, results on relationships between logical form of expectations and their relative accuracy and validity. We have argued that the best expectations are the most appropriate for applications, and we have outlined how they can be generated. We have shown how analysing violated expectations can be used as part of a process of information fusion.

One of the advantages of the framework presented in this paper is the potential for using background knowledge together with reports. So instead of looking for reports that violate expectations, we look for reports plus background knowledge that violate expectations. This background knowledge is application dependent, but may include ontological information (e.g. synonyms, hyponyms, and hypernyms) and practical reasoning axioms (e.g. axioms for drawing temporal relationships between information in reports).

As we have shown, a violation of an expectation is a form of inconsistency. So the evaluation of violations of expectations, as presented in this paper, is a form of measuring of inconsistency. The better confirmed an expectation is, the more significant the inconsistency arising when the expectation is violated. This approach has wider applicability in measuring inconsistency in knowledge. The traditional view in logic is that a set of formulae is either consistent or inconsistent. However, for artificial intelligence, we need more than this binary classification. We need to be able to better describe the nature of inconsistency in a set of formulae. Some proposals for measuring inconsistency have been put forward (see for example [Loz94, Hun02b, Hun03, KLM03]), but the field needs further development [HK04].

The framework for expectations presented in this paper, provides another alternative in the space of approaches to measuring and analysis of inconsistency. First, since this framework is based on expectations, we are considering a specific kind of information: Expectations are a form of rule reflecting “normality”, and so implicitly, it is accepted they will be inconsistent in abnormal contexts. Second, there is a record of the accuracy and validity of each expectation: This gives a graded approach to treating violations (inconsistencies). In other words, not all inconsistencies involving expectations are treated equally. This is a result of the framework being more structured than other approaches to analysising inconsistency. Finally, the application of analysing violation of expectations in information fusion (Section 7) indicates the potential utility of this approach to measuring inconsistency.

Acknowledgements

This work has been supported by the EPSRC project (GR/R22551/01) “Logic-based Technology for Measuring Inconsistency of News Reports”.

References

[BCVB01] S Bergamaschi, S Castano, M Vincini, and D Beneventano. Semantic integration of heterogeneous information sources. *Data and Knowledge Engineering*, 36:215–249, 2001.

[BDK97] G Brewka, J Dix, and K Konolige. *Non-monotonic Reasoning: An Overview*. CLSI Publications, 1997.

[BH04] E Byrne and A Hunter. Man bites dog: Looking for interesting inconsistencies in structured news reports. *Data and Knowledge Engineering*, 48:265–285, 2004.

[BKMS92] C Baral, S Kraus, J Minker, and V Subrahmanian. Combining knowledgebases of consisting of first-order theories. *Computational Intelligence*, 8:45–71, 1992.

[Byr05] E Byrne. *Representing and reasoning with expectations*. PhD thesis, UCL Department of Computer Science, 2005. (in preparation).

[CGL+98] D Calvanese, G De Giacomo, M Lenzerini, D Nardi, and R Rosati. Source integration in data warehousing. In *Proceedings of the 9th International Workshop on Database and Expert Systems (DEXA'98)*, pages 192–197. IEEE Computer Society Press, 1998.

[CH97] L Cholvy and A Hunter. Information fusion in logic: A brief overview. In D Gabbay, R Kruse, A Nonnengart, and H-J Ohlbach, editors, *Symbolic and Quantitative Approaches to Uncertainty (ECSQARU'97)*, Lecture Notes in Computer Science, pages 86–95. Springer, 1997.

[CHS93] J Cussens, A Hunter, and A Srinivasan. Generating explicit orderings for non-monotonic logics. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI'93)*, pages 420–425. MIT Press, 1993.

[CM01] L Cholvy and S Moral. Merging databases: Problems and examples. *International Journal of Intelligent Systems*, 10:1193–1221, 2001.

[Fit89] M Fitting. Bilattices and the theory of truth. *Journal of Philosophical Logic*, 18:225–256, 1989.

[Gil96] D Gillies. *Artificial Intelligence and Scientific Method*. Oxford University Press, 1996.

[Gin86] M Ginsberg. Multi-valued logics. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI'86)*, pages 243–247, Philadelphia PA, 1986. Morgan Kaufmann.

[HK04] A Hunter and S Konieczny. Approaches to measuring inconsistent information. In *Inconsistency Tolerance*, volume 3300 of *Lecture Notes in Computer Science*. Springer, 2004, (in press).

[HS03] A Hunter and R Summerton. Propositional fusion rules. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 2711 of *Lecture Notes in Computer Science*, pages 502–514. Springer, 2003.

[HS04] A Hunter and R Summerton. Fusion rules for context-dependent aggregation of structured news reports. *Journal of Applied Non-classical Logic*, 2004. in press.

[Hun02a] A Hunter. Logical fusion rules for merging structured news reports. *Data and Knowledge Engineering*, 42:23–56, 2002.

[Hun02b] A Hunter. Measuring inconsistency in knowledge via quasi-classical models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'02)*, pages 68–73. MIT Press, 2002.

[Hun03] A Hunter. Evaluating the significance of inconsistency. In *Proceedings of the International Joint Conference on AI (IJCAI'03)*, pages 468–473. Morgan Kaufmann, 2003.

[KLM03] S Konieczny, J Lang, and P Marquis. Quantifying information and contradiction in propositional logic through epistemic actions. In *Proceedings of the 18th International Joint Conference on Artificial Intellignce (IJCAI'03)*, pages 106–111, 2003.

[KP98] S Konieczny and R Pino Perez. On the logic of merging. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 488–498. Morgan Kaufmann, 1998.

[Loz94] E Lozinskii. Information and evidence in logic systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:163–193, 1994.

[LS98] P Liberatore and M Schaerf. Arbitration (or how to merge knowledgebases). *IEEE Transactions on Knowledge and Data Engineering*, 10:76–90, 1998.

[Mug99a] S Muggleton. Inductive logic programming: issues, results and the LLL challenge. *Artificial Intelligence*, 114(1–2):283–296, December 1999.

[Mug99b] S Muggleton. Scientific knowledge discovery using Inductive Logic Programming. *Communications of the ACM*, 42(11):42–46, November 1999.

[PHG+99] A Preece, K Hui, A Gray, P Marti, T Bench-Capon, D Jeans, and Z Cui. The KRAFT architecture for knowledge fusion and transformation. In *Expert Systems*. Springer, 1999.

[Plo70] G Plotkin. A note on inductive generalization. In B Meltzer and D Michie, editors, *Machine Intelligence 5*, pages 153–163. Edinburgh University Press, 1970.

[Rei78] R Reiter. On closed world databases. In H Gallaire and J Minker, editors, *Logic and Databases*, pages 55–76. Plenum Press, 1978.