This paper was selected by a process of
anonymous peer reviewing for presentation at

COMMONSENSE 2007

8th International Symposium on Logical Formalizations of Commonsense Reasoning

Part of the AAAI Spring Symposium Series, March 26-28 2007,
Stanford University, California

Further information, including follow-up notes for some of the
selected papers, can be found at:

www.ucl.ac.uk/commonsense07

A Coherent and Paraconsistent Variant of the Default Logic [*]

Anbu Yue and **Zuoquan Lin**
Department of Information Sciences, Peking University, Beijing 100871, China
{yueanbu, lz}@is.pku.edu.cn

Abstract

Reiter's default logic is supposed to reasoning on consistent knowledge; when inconsistencies or contradictions are present in a default theory, no useful conclusions can be extracted. In the past years, fragments and variants of the default logic are proposed to avoid or handle inconsistencies or contradictions. Unfortunately, the expressive and reasoning power of these fragments are strictly weaker than the full version of Reiter's default logic, and the semantics of the variants are changed even when the default theory is consistent and contradiction-free. In this paper, we propose a paraconsistent Annotated Default Logic, in which, the existence of non-trivial annotated extensions is guaranteed. In addition, the same conclusions are extracted as Reiter's default logic does, as long as the default theory has non-trivial extensions in Reiter's default logic. As a consequent, the intended meaning of the default theories are kept unchanged when shifting to our method. As a by product, the extra information presented in the annotated extensions can help the users in analyzing and modifying their knowledge representations.

Introduction

Reiter's default logic (Reiter 1980) was studied widely for its clarity in syntax as well as strong power in knowledge representation. However, Reiter's default logic is supposed to reason with consistent knowledge: even a single inconsistence or contradiction presented in the premise will lead to the triviality or non-existence of extensions.

One might take the viewpoint that default theories without extensions are illegal or problematic, and the user has the responsibility of providing a "good" one. They hope to find out and/or characterize those coherent fragments (i.e. the existence of extensions is guaranteed) of default logic. Among them there are normal (Reiter 1980), ordered (Papadimitriou & Sideri 1994), and strongly stratified (Cholewinski 1995) default logics. Not surprisingly, such fragments are strictly less expressive than the default logic.

Another view regards all default theories as legal and provide extensions for them. This viewpoint is supported by the trend of heterogeneous data sources and the fact that it is time-consuming to point out whether a default theory is a "good" one. If the premise in not so "good", that is, no extensions exists in Reiter's default logic, a "robust" logic can draw some conclusions. In order to get a "robust" default logic, some researchers try to modify the definition of extension in such a way that all default theories have at least one extensions, such as constraint default logic (Schaub 1992) and justified default logic (Lukaszewicz 1984). Regarding the inconsistences among formulas which lead to the trivial extension, some researchers refer to paraconsistent logics (daCosta 1974; Lin 1996), multi-valued logics in particular (Belnap 1977; Arieli & Avron 1998; Ginsberg 1988). The bi-default logic (Han 2004) and the four-valued default logic (Yue & Lin 2005; Yue, Ma, & Lin 2006) are two examples. Unfortunately, the semantics of these logics are different from that of Reiter's default logic, even when the default theories are consistent and contradiction-free. As a direct consequent, the intended meaning of the default theories are changed under these semantics. This is arguably an important reason why these variants are not accepted and used so widely.

Different from all the above, we try to preserve the intended meaning of default theories as well as to preserve the strong expressive and reasoning power of Reiter's default logic simultaneously. Every default theory has non-trivial annotated extensions under the semantics defined in this paper. Further more, when a default theory have extensions in Reiter's default logic, our method draws the same conclusions as Reiter's default logic does. In addition, inconsistencies among formulas and contradictions among defaults are all annotated out, which provide meaningful hints for why the default theories are not so "good" as the user wanted.

We also visualize, by some examples, how our method can be used to analyze the default theories, when inconsistences and/or contradictions are present. As a by product, our method can be used as a diagnostic tools to help in correcting their representation or enhancing the performance of the system. The formal study on this topic is out of the range of this paper, and we will further the discussions in future work.

The approach of annotating formulas has been broadly studied in logic programming (e.g. (Subrahmanian 1987; Damasio, Pereira, & Swift 1999; Vennekens, Verbaeten, & Bruynooghe 2004) among others). But when classical nega-

[*]Partially supported by NSFC (grant numbers 60373002 and 60496322) and by NKBRPC (2004CB318000).

tion is present, the annotated logic programs are not necessarily have model(s). In contract, in the annotated default logic, non-trivial annotated extensions exist for any default theory.

The rest of the paper is organized as follows. In the next section, we briefly review Reiter's default logic, and discuss the incoherence and multi-extensions problem of the default logic and then we present the underlying logic of the annotated default logic. In Section 3, we introduce the annotated default logic in detail. Then in Section 4, we compare the annotated default logic with Reiter's default logic in detail, and visualize how to analyze the default theories with the annotated default logic by some examples in Section 5. After discussing some related works in Section 6, we conclude this paper in Section 7.

Default Logic

Default Logic

In this paper, we denote $\mathcal{L}$ as a propositional language. A theory is a set of formulas in $\mathcal{L}$, and Th is the consequence operator.

A *default* is an inference rule of the form $\frac{\alpha:\beta_1,\cdots,\beta_n}{\gamma}$, $(n \geq 1)$ where $\alpha, \beta_1, \cdots, \beta_n$ and γ are formulas in $\mathcal{L}$. α is called the *prerequisite*, $\beta_1, \cdots, \beta_n$ the *justifications*, and γ the *consequence* of the default. A default theory is a pair $T = (W, D)$, where W is a set of formulas, and D is a set of defaults. A default is *normal* if it is of the form $\frac{\alpha:\beta}{\beta}$, and a default is *semi-normal* if it is of the form $\frac{\alpha:\beta\wedge\gamma}{\gamma}$. A default theory T is semi-normal if all defaults in T are semi-normal. Janhunen proved that Reiter's default logic and its semi-normal fragment are of equally expressive power (Janhunen 2003).

A default theory may have none, a single or multiple *(default) extensions* defined by:

Definition 1 ((Reiter 1980)) *Let E be a set of formulas, and $T = (W, D)$ be a default theory. Define*

$$E_0 = W$$

and for all $i \geq 0$

$$E_{i+1} = Th(E_i) \cup \left\{ \gamma \,\middle|\, \frac{\alpha : \beta_1, \cdots, \beta_n}{\gamma} \ where\ \alpha \in E_i \ and\ \neg\beta_1 \notin E, \cdots, \neg\beta_n \notin E \right\}$$

Then E is an (default) extension of T iff $E = \bigcup_{i=0}^{\infty} E_i$

A default theory is called *coherent* if it has at least one extension, otherwise, it is called *incoherent*. A default theory may also have a *trivial* extension, which contains all formulas of $\mathcal{L}$.

Theorem 1 ((Reiter 1980)) *A default theory $T = (W, D)$ has a trivial extension iff W is inconsistent.*

Inconsistences and Contradictions

In the previous subsection, we have seen that a default theory may have none or a trivial extension.

The triviality of the default logic originates from its underlying logic, which can not handle *inconsistences* among formulas, and the incoherence of default logic comes from the fact that it can not handle *contradictions* brought up by defaults D (with W involved).

Example 1 *Consider the following four default theories $T_i = (W_i, D_i), i = 1, 2, 3, 4$ where*

- $W_1 = \emptyset$, $D_1 = \{\frac{:p}{q}, \frac{p}{\neg q}\}$.
- $W_2 = \{q\}$, $D_2 = \{\frac{:p}{\neg q}\}$.
- $W_3 = \emptyset$, $D_3 = \{\frac{:p}{\neg p}\}$.
- $W_4 = \emptyset$, $D_4 = \{\frac{:p}{q}, \frac{q:r}{\neg p}\}$.

It is easy to verify that none of T_1, T_2, T_3, T_4 has extensions.

In another hand, any two extensions of a default logic can not be put together to obtain another extension, because some defaults can not be applied at the same time:

Example 2 $D = \{\frac{:\neg p}{q}, \frac{:\neg q}{p}\}$, *and $W = \emptyset$. Default theory $T = (W, D)$ has two nontrivial extensions, $Th(\{p\})$ and $Th(\{q\})$. In this example, one default says that p is true unless q is ture, while the other says that q is true unless p is true. Surely, these two defaults are incompatible. Therefore, they are split into two extensions, one contains the first default and the other contains the second.*

In this example, it is easy to point out why the default theory have multiple answers, but when the default theory is complex, it is not that easy anymore. So, maybe some unwanted extensions are given but we do not know exactly why.

In knowledge representation, some useful things can be expressed utilizing the non-existence and/or multi-extensions. One of the most important things is to express constraints, which can be expressed as $\frac{p\wedge q:r\vee\neg r}{r\wedge\neg r}$. It is claimed that p and q should not hold at the same time. When both of p and q are present in the premise, no extensions exists (except for a trivial one if W is inconsistent). Inversely, when a default theory lacks extensions, it is likely that some constraints are violated. But, in the practice, knowing that *some* constraints are possibly unsatisfied is not enough; we would like to know *which* constraints are broken. Another example is multi-extensions, which is useful especially in representing multi-goal problem; one extension contains some goals and the corresponding executable planning or schedule. Sometimes, we want to know how to modify the system to achieve more goals simultaneously. In this situation, we need to find out which part of the premises make the goals distributed into different extensions. Thus, in order to enhance the performance, only that part should be updated.

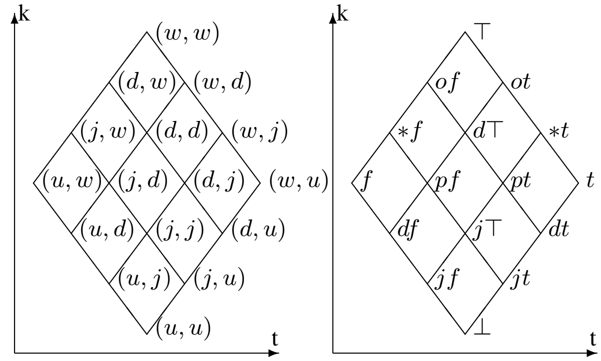

Figure 1: The bilattice $\mathcal{BL}$ that has 16 truth values and its alternative representation, where (w, w) is also referred as $\top$, and (d, w) is also referred as of, etc.

Bilattice and Annotated logic

In this paper, we propose an annotated logic to serve as the underlying logic to handle inconsistency.

The family of annotated logics (Kifer & Lozinskii 1992) is a generalization of annotated logic programs introduced by Subrahmanian (Subrahmanian 1987). In Kifer's annotated logic, a belief semi lattice is required (see (Kifer & Lozinskii 1992) for detail), while in this paper, we define a bilattice as the math structure. *Bilattice* is presented by Ginsberg by generaliing Belnap's four-valued logic (Belnap 1977) for unifying several existing formalisms of logical reasoning in artificial intelligence (Ginsberg 1988). Bilattice-valued logics and their preferential reasoning have some nice properties, see (Ginsberg 1988; Arieli & Avron 1998; 2000; Fitting 2002) for details.

In Reiter's default logic, beliefs in an extension can be divided into four levels of uncertainty: unknown, satisfiable, revisable, and believable. The conclusions that can not be revised are believable; the conclusions drawn by applying some defaults are revisable; the justifications of applicable defaults are satisfiable, although they are not included in the extension; and the remains are unknown. The four levels of uncertainty are denoted as u, j, d, w respectively in this paper, and they are ordered by $u < j < d < w$, which forms a lattice. We can define a bilattice $\mathcal{BL}$ (Fig 1) with 16-values by:

$$
\begin{array}{l}
(x_1, y_1) \leq_t (x_2, y_2) \text{ iff } x_1 \leq x_2, y_1 \geq y_2 \\
(x_1, y_1) \leq_k (x_2, y_2) \text{ iff } x_1 \leq x_2, y_1 \leq y_2 \\
\neg(x, y) = (y, x)
\end{array}
$$

where x_1, x_2, y_1, y_2, x, y are one of u, j, d, w.

In the bilattice, we use $\wedge$ and $\vee$ for meet and join which correspond to the $\leq_t$ lattice. And those correspond to the $\leq_k$ lattice are not used in this paper.

A valuation v maps each atom in $\mathcal{L}$ to an element in $\mathcal{BL}$. Any valuation is extended to complex formulas in the obvious way. In this paper we extend the valuation to the annotated formulas. An *annotated formula* is of the form $\phi : \mu$, where ϕ is a formula of the propositional language $\mathcal{L}$ and μ is an element from the bilattice $\mathcal{BL}$. In this paper, we denote $(\Sigma : \mu)$ as the set $\{\phi : \mu \mid \phi \in \Sigma\}$. A valuation v is a model $(\phi : \mu)$ iff $v(\phi) \geq_k \mu$.

Preferential reasonings can be defined in bilattice-valued logics to improve their reasoning power, see (Ginsberg 1988; Arieli & Avron 1998) for more details.

The elements in $\mathcal{BL}$ can be divided into four subsets: $\mathcal{F}_t = \{dt, pt, d\top, of, t, \star t, ot, \top\}$, $\mathcal{F}_f = \{df, pf, d\top, ot, f, \star f, of, \top\}$, $\mathcal{F}_\top = \mathcal{F}_t \cap \mathcal{F}_f$, and $\mathcal{F}_\bot = \mathcal{BL} \setminus (\mathcal{F}_t \cup \mathcal{F}_f)$.

Definition 2 *A valuation v is called more classical than μ, denoted as $\mu \prec v$, if $\mu(p) \in \mathcal{F}_\top \cup \mathcal{F}_\bot$ whenever $v(p) \in \mathcal{F}_\top \cup \mathcal{F}_\bot$, for every atom p in $\mathcal{L}$.*

Definition 3 *Let E be a set of annotated formulas. Then we write $E \models^A_\prec \phi : \mu$ iff any most classical model of E is a model of $(\phi : \mu)$.*

In this paper, we denote $Th^A_\prec(E) = \{(\phi : \mu) \mid E \models^A_\prec (\phi : \mu)\}$, where E is a set of annotated formulas. It is worth noting that if $(\phi : \mu) \in Th^A_\prec(E)$ and $\nu \leq_k \mu$ then $(\phi : \nu) \in Th^A_\prec(E)$.

Following the work in (Arieli & Avron 1998), we have that:

Corollay 1 *Let E be a set of consistent formulas and E' be a set of annotated formulas, s.t. $\phi \in E$ iff $(\phi : \mu) \in E'$ for some $\mu \in \{dt, t\}$. Then $E \models_{cl} \psi$ iff $E' \models^A_\prec (\psi : dt)$, where $\models_{cl}$ is the consequence relation of classical logic.*

According to (Arieli & Avron 2000), it is easy to prove that the consequent relation $\models^A_\prec$ satisfies *cautious monotonicity* and *cautious cut* property.

Annotated Default Logic

The Logic

In Reiter's default logic, a justification of a default is satisfiable in a context S, which is a set of formulas, if its negation is not derivable in S. Since the underlying logic is paraconsistent, a formula and its negation can presented simultaneously, the satisfiability of the justifications should be redefined:

Definition 4 *Let E be a set of annotated formulas, and let ϕ be a formula. Denote $E \models_j \phi$ if*

- $E \models^A_\prec (\phi : jt)$ *and* $E \not\models^A_\prec (\phi : f)$, *or*
- $E \not\models^A_\prec (\phi : jf)$

If E does not contain any information contradict with ϕ, i.e. $E \not\models^A_\prec (\phi : jf)$, then ϕ is satisfiable in the context of E. In another hand, $E \models^A_\prec (\phi : jt)$ explicitly state that ϕ is satisfiable in the context of E, and ϕ is not satisfiable if it is definitely false in the context of E, i.e. $E \models^A_\prec (\phi : f)$.

Definition 5 *Let E be a set of annotated formulas, and let $T = (W, D)$ be a default theory.*

Define

$$\begin{aligned} E_0 &= (W:t) \\ E_{i+1} &= Th^A_{\prec}(E_i) \cup \{(\gamma : dt), (\beta_1 : jt), \ldots, (\beta_n : jt) \mid \\ &\quad \frac{\alpha : \beta_1, \ldots, \beta_n}{\gamma} \in D, (\alpha : dt) \in E_i, \text{ and} \\ &\quad E \models_j \beta_1, \ldots, E \models_j \beta_n)\} \end{aligned}$$

E is an annotated extension of T iff $E = \bigcup_{i=0}^{\infty} E_i$.

In the annotated extensions, only those formulas with annotation greater than dt in the partial order $\leq_k$ are considered as true. In Section , we will show that an extension in Reiter's default logic can be viewed as a simplification of an annotated extension.

In Definition 5, $Th^A_{\prec}(E_i)$ is presented in constructing E_{i+1}, which means that all conclusions followed by E_i are all added into E_{i+1}. Recall that a logic is cautious monotonic means that added conclusions inferred by a theory into the theory itself will not revised any conclusions, and a logic satisfies cautious cut means that add conclusions inferred by a theory into itself will never introduce new conclusions. So, it is sufficient and necessary to require the underlying logic to satisfy the cautious monotonicity and cautious cut property.

The following example shows how the annotated default logic works:

Example 3 *Let $T = (\{p, p \to q\}, \{\frac{q:r}{\neg p}\})$ be a default theory. T has no extensions in Reiter's default logic, but it has a unique annotated extension $E = Th^A_{\prec}(p : ot, q : t, r : jt)$.*

By analyzing the annotated extension E of $T = (W, D)$, we can obtain the inconsistent formulas by $Inc(E) = \{\phi \mid (\phi : d\top) \in E\}$ and the contradictory defaults by $GD^T_{\top}(E) = \{\frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma} \in D \mid (\alpha : dt) \in E, (\beta_i : pf) \in E \; for\; some\; 1 \leq i \leq n\}$. Therefore, the pair $(Inc(E), GD^T_{\top}(E))$ can be used to characterize the degree of plausibility of the annotated extension E of T. An annotated extension E_1 is called more plausible than E_2 iff $Inc(E_1) \subseteq Inc(E_2)$ and $GD^T_{\top}(E_1) \subseteq GD^T_{\top}(E_2)$.

Existence of Annotated Extensions

The annotated default logic is non-trivial because the underlying annotated logic is paraconsistent. In this subsection, we will show that the annotated default logic is coherent, i.e. any default theory has at least one annotated extensions.

At first, any semi-normal default theories has annotated extensions:

Theorem 2 *Let $T = (W, D)$ be any semi-normal default theory. $SQ = [d_1, d_2, \cdots, d_n]^{\infty}$ is a sequence, in which $[d_1, d_2, \cdots, d_n]$ is a permutation of D.*

Define

$$E_0 = (W : t)$$

and for $i \geq 0$

$$\begin{aligned} E_{i+1} &= Th^A_{\prec}(E_i) \cup \{(\gamma : dt), (\gamma \wedge \beta : jt) \mid \\ &\quad d_i = \frac{\alpha : \beta \wedge \gamma}{\gamma} \in D, \text{where} \\ &\quad (\alpha : dt) \in E_i, E_i \models_j \gamma \wedge \beta\} \end{aligned}$$

then $E = \bigcup_{i=0}^{\infty} E_i$ is an annotated extension of T.

In the above theorem, $[d_1, d_2, \cdots, d_n]^{\infty}$ is the infinite sequence $[d_1, d_2, \ldots, d_n, d_1, d_2, \ldots, d_n, \ldots]$.

It is worth noting that a semantic equivalent transformation from default theory to semi-normal ones are provided in (Janhunen 2003), and extensions of the semi-normal default theories can be reduced to the extensions of the original ones. By using the same transformation and reducing operators, we can prove that[1]:

Corollay 2 (Existence of Annotated Extensions) *For any default theory $T = (W, D)$, T has at least one annotated default extensions.*

Thus, inconsistencies among formulas and contradictions involving defaults can be resolved, as shown in the following examples:

Example 4 *Consider the default theories given in Example 1, T_i has a unique annotated extension E_i, s.t.*

- $E_1 = Th^A_{\prec}(\{p : jt, q : d\top\})$, *in which, we do not know whether p is true of false, but it is likely to be true. q is regarded as both true and false, but such inconsistent conclusion can be revised when new information is achieved.*
- $E_2 = Th^A_{\prec}(\{p : jt, q : ot\})$. *$q$ is also regarded as an inconsistence, but it can not be revised to false.*
- $E_3 = Th^A_{\prec}(\{p : pf\})$. *$p$ is regarded as consistently false, but such conclusion is problematic since it is regarded as likely to be true at the same time.*
- $E_4 = Th^A_{\prec}(\{p : pf, q : dt, r : jt\})$. *The conclusion is drawn by firs assume that p is not false and then infer that p is false by default, so, the statement about p is problematic.*

Compare with Reiter's Default logic

In the annotated extensions, conclusions (formulas) are annotated with different levels of uncertainties. So, we expect that the annotated extensions provid more detailed characterizations than Reiter's default extensions do. We also expect that these extra information help the users in knowledge representation as well as analysis of the default theories.

Definition 6 *Let E be set of annotated formulas, denote $Cls(E) = \{\phi \mid (\phi : \mu) \in E, \mu \geq_k dt\}$.*

[1] In fact, we proved that, under the transformation $\mathcal{T}rans(\cdot)$, by omitting auxiliary symbols, the annotated extensions of the default theory $\mathcal{T}rans(T)$ are annotated extensions of the default theory T.

Informally speaking, $Cls(E)$ contains the same statements as E does, since only those statements with the annotation greater than dt in the partial order k are regarded as true.

Proposition 1 *Let E be a set of annotated formulas. Then $Inc(E) = \emptyset$ iff $Cls(E)$ is consistent.*

Theorem 3 (Classicalness) *Suppose that E is an annotated extension of default theory $T = (W, D)$. If $Inc(E) = \emptyset$ and $GD_{\top}^{T}(E) = \emptyset$, then $Cls(E)$ is an nontrivial extension of (W, D) in Reiter's default logic.*

In another hand, by adding the information about the justifications of the defaults that generate the (Reiter's default) extension, we can obtain all we need in an annotated extension:

Theorem 4 *If E is an extension of a default theory (W, D) in Reiter's default logic, then E' is an $\leq_k$ minimal annotated extension of (W, D), and E' is classical consistent, where $E' = Th_{\prec}^{A}((W : t) \cup \{(\gamma : dt), (\beta_1 : jt), \ldots, (\beta_n : jt) \mid \frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma} \in GD^T(E)\})$, where $GD^T(E) = \{\frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma} \in D \mid \alpha \in E, \neg\beta_1 \notin E, \ldots, \neg\beta_n \notin E\}$.*

Since a subset of annotated extensions are identical to extensions in Reiter's default logic in the sense of the conclusions contained by them, the annotated default logic is an extension of Reiter's default logic.

Knowledge Representation

In annotated default logic, conclusions are annotated with different annotations, from which useful information can be extracted to help us to analyze the default theories. In this section, we will show how this can be done by some examples. We will further our work on this interesting topic in the future.

Example 5 *Let $T = (\{p, q\}, \{\frac{p,q:c_1 \vee \neg c_1}{c_1 \wedge \neg c_1}, \frac{q,r:c_2 \vee \neg c_2}{c_2 \wedge \neg c_2}\})$. The two constraints in T are referenced by c_1 and c_2 respectively. Obviously, constraint c_1 is broken but c_2 is not. T has no extension in Reiter's default logic, and thus we can not infer why it lacks extensions. Contrarily, T has an annotated extension $E = Th_{\prec}^{A}(\{p : t, q : t, c_1 : d\top, c_2 : \bot\})$, in which the annotation $d\top$ of c_1 indicate that it is broken and $\bot$ indicate that c_2 is not, just as expected.*

Some times, we need to express things which may have more than one possible answers:

Example 6 *Let $T = (\emptyset, D_1 \cup D_2)$, where $D_1 = \{\frac{:\neg(process_{job1} \wedge process_{job2})}{\neg(process_{job1} \wedge process_{job2})}, \frac{:process_{job1}}{process_{job1}}, \frac{:process_{job2}}{process_{job2}}\}$, and $D_2 = \{\frac{process_{job1}:g_1}{g_1}, \frac{process_{job2}:g_2}{g_2}\}$. The defaults in D_1 make sure that exactly one of $job1$ and $job2$ be processed, but if using classical statement $process_{job1} \vee process_{job2}$ to represent that, we can not get any extension contains $process_{job1}$ or $process_{job2}$. T has two annotated extensions: $E_1 = Th_{\prec}^{A}(process_{job1} : dt, g_1 : dt)$ and $E_2 = Th_{\prec}^{A}(process_{job2} : dt, g_2 : dt)$. What will happen if we want g_1 and g_2 achieved at the same time? T has also another annotated extension $E_3 = Th_{\prec}^{A}(process_{job1} : d\top, process_{job2} : d\top, g_1 : dt, g_2 : dt)$, in which both of g_1 and g_2 are annotated as dt. Notice that in E_3, $process_{job1}$ and $process_{job2}$ are annotated as a inconsistent label $\top$. This fact indicate that, only after when we change the premise to make them consistent, we can achieve more goals. Obviously, by adding more processors we can process more jobs simultaneously, and then get more outcomes. It is worth noting that $Inc(E_3) \neq \emptyset$ and thus is less plausible as E_1 and E_2. So, We can not infer that $g_1 \wedge g_2$ if we only consider the most plausible annotated extensions.*

Even when the default theory is consistent and coherent, our method can extract interesting information:

Example 7 *Let $T = (W, D)$ be a default theory with $W = \{broken\_a \vee broken\_b\}$ and $D = \{\frac{:usable\_a \wedge \neg broken\_a}{usable\_a}, \frac{:usable\_b \wedge \neg broken\_b}{usable\_b}\}$. T also has a single annotated extension: $E' = Th_{\prec}^{A}(\{(broken\_a \vee broken\_b) : \star t, broken\_a : jf, broken\_b : jf, usable\_a : dt, usable\_b : dt\})$. In this annotated extension, we can infer that both a and b are usable, but we do not know precisely which one, just as same as what we can infer in Reiter's default logic. However, the annotation $\star t$ means that some justifications can not be satisfied simultaneously because of the statement $(broken\_a \vee broken\_b)$, and so, the obtained annotated extension E' is doubtable.*

Example 8 *Let $T = (\emptyset, \{\frac{:p}{q}, \frac{:\neg p}{r}\})$. T has only one extension $E = Th(\{q, r\})$, in which $q \wedge r$ is true. But this conclusion is "brittle", i.e. when any new information about p is achieved, it will be revised. This is not a problem in general, but sometimes the users do not want that. E has a counterpart E' which is an annotated extension of T, where $E' = Th_{\prec}^{A}(\{p : j\top, q : dt, r : dt\})$. In E', the annotation $j\top$ indicate that some conclusions are possibly too "brittle".*

Related Works

Coherent variants of Reiter's default logic have been presented in the past years, among them, there are constraint default logic (Schaub 1992), and justified default logic (Lukaszewicz 1984). Also, paraconsistent versions was proposed too, such as the bi-default logic (Han 2004), the four-valued default logic (Yue & Lin 2005; Yue, Ma, & Lin 2006).

In constraint default logic and justified default logic, by preserving justifications of applied defaults, constraint and justified extensions are ensured to exist for any default theory. As a result, the intended meaning of the default theories are correspondingly changed as long as the conclusions are changed. Our method is different from them in that: 1) ours aim at resolving inconsistences and contradictions by applying multi-valued logic; 2) ours is an extension of Reiter's default logic, but constraint default logic and justified default logic are less expressive than Reiter's default logic (Delgrande & Schaub 2003).

Bi-default logic and the four-valued default logic are paraconsistent; both of them are non-trivial. Using the technique of formula transformation, inconsistences and contradictions can be resolved. But some default theories lack bi-extensions and/or k-minimal models. In contract, any default theory has at least one annotated extension, in which, possible inconsistencies and contradictions are annotated if there are any.

Some other semantics are also provided for default theories, such as well-founded semantics (Gelder, Ross, & Schlipf 1991). We do not compare ours with them exhaustively, since the annotated default logic is an extension of Reiter's default logic.

Conclusions and Future Work

Our main contribution in this paper is to provide the annotated default logic, which is paraconsistent and coherent, thus meaningful knowledge can be extracted even when inconsistencies and contradictions present. When a default theory has non-trivial extensions in Reiter's default logic, the same conclusions are extracted in the annotated default logic. As a consequent, the intended meaning of the default theories are keeping unchanged when shifting from Reiter's default logic to the annotated default logic.

In another hand, the extensions in Reiter's default logic can be viewed as simplified annotated extensions which are classical consistent. Therefore the annotated default logic is an extension of Reiter's default logic. In our method, conclusions can be distinguished according to their different level of uncertainty. Therefore, the annotation extensions provide a more detailed characterization, and contain more information, which help the users in evaluating, analyzing and modifying their knowledge representations.

In the future, we will try to implement the annotated default logic as well as consider the applications of the annotated default logic in knowledge representation and reasoning.

References

Arieli, O., and Avron, A. 1998. The value of the four values. *Artificial Intelligence* 102(1):97–141.

Arieli, O., and Avron, A. 2000. General patterns for nonmonotonic reasoning: From basic entailments to plausible relations. *Logic Journal of the IGPL* 8(2):119– 148.

Belnap, N. 1977. A useful four-valued logic. In *Modern uses of multiple-valued logic*. 30–56.

Cholewinski, P. 1995. Stratified default theories. *Computer Science Logic* 933:456–470.

daCosta, N. 1974. Theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic* 15:497–510.

Damasio, C. V.; Pereira, L. M.; and Swift, T. 1999. Coherent well-founded annotated logic programs. *Logic Programming and Nonmonotonic Reasoning* 1730:262–276.

Delgrande, J. P., and Schaub, T. 2003. On the relation between reiter's default logic and its (major) variants. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Proceeding* 2711:452–463.

Fitting, M. 2002. Bilattices are nice things. In *Conference on Self-Reference, proceedings.*

Gelder, A. V.; Ross, K. A.; and Schlipf, J. S. 1991. The wll-founded semantics for general logic programms. *Journal of ACM* 38(3):620–650.

Ginsberg, M. L. 1988. Multivalued logics: a uniform approach to reasoning in artificial intelligence. *Computational Intelligence* 4:265–316.

Han, Q. 2004. Paraconsistent default reasoning. In *10th International Workshop on Non-Monotonic Reasoning*, 197–203.

Janhunen, T. 2003. Evaluating the effect of semi-normality on the expressiveness of defaults. *Artificial Intelligence* 144(1-2):233–250.

Kifer, M., and Lozinskii, E. L. 1992. A logic for reasoning with inconsistency. *Journal of Automated Reasoning* 9(2):179–215.

Lin, Z. 1996. Paraconsistent circumscription. *Journal of Pattern Recognition and Artificial Intelligence* 10.

Lukaszewicz, W. 1984. Considerations on default logic. In *Proceedings of the Non-Monotonic Reasoning Workshop*, 165–193.

Papadimitriou, C. H., and Sideri, M. 1994. Default theories that always have extensions. *Artificial Intelligence* 69(1-2):347 – 357.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.

Schaub, T. 1992. On constrained default theories. In *ECAI*, 304–308.

Subrahmanian, V. S. 1987. On the semantics of quantitative logic programs. In *4th IEEE Symposium on Logic Programming*, 173–182. Computer Society Press, Washington DC.

Vennekens, J.; Verbaeten, S.; and Bruynooghe, M. 2004. Logic programs with annotated disjunctions. *Logic Programming, Proceedings* 3132:431–445.

Yue, A., and Lin, Z. 2005. Default logic based on four valued semantics. *Chinese journal of computer* 28(9):1447–1458.

Yue, A.; Ma, Y.; and Lin, Z. 2006. Four-valued semantics for default logic. In *19th Conference of the Canadian Society for Computational Studies of Intelligence*, volume 4013 of *Lecture Notes in Computer Science*, 195–205.