Fundamenta Informaticae 1(1996)1-22
IOS Press

Belief Revision in Non-Monotonic Reasoning and Logic Programming*

José Júlio Alferes
DM, U. Évora and
CITIA, U. Nova de Lisboa
2825 Monte da Caparica, Portugal
jja@fct.unl.pt

Luís Moniz Pereira
D. Informática and CITIA
U. Nova de Lisboa
2825 Monte da Caparica, Portugal
lmp@fct.unl.pt

Teodor C. Przymusinski
Department of Computer Science
University of California
Riverside, CA 92521
teodor@cs.ucr.edu

Abstract. In order to be able to explicitly reason about beliefs, we've introduced a non-monotonic formalism, called the Autoepistemic Logic of Beliefs, AEB, obtained by augmenting classical propositional logic with a belief operator, $\mathcal{B}$. For this language we've defined the static autoepistemic expansions semantics. The resulting non-monotonic knowledge representation framework turned out to be rather simple and yet quite powerful. Moreover, it has some very natural properties which sharply contrast with those of Moore's AEL.

While static expansions seem to provide a natural and intuitive semantics for many belief theories, and, in particular, for all affirmative belief theories (which include the class of all normal and disjunctive logic programs), they often can lead to inconsistent expansions for theories in which (subjective) beliefs clash with the known (objective) information or with some other beliefs. In particular, this applies to belief theories (and to logic programs) with strong or explicit negation.

In this paper we generalize AEB to avoid the acceptance of inconsistency provoking beliefs. We show how such AEB theories can be revised to prevent belief originated inconsistencies, and also to introduce declarative language level control over the revision level of beliefs, and apply it to the domains of diagnosis and declarative debugging. The generality of our AEB framework can capture and justify the methods that have been deployed to solve similar revision problems within the logic programming paradigm.

Keywords: Belief Revision, Logics of Knowledge and Beliefs, Non-Monotonic Reasoning, Logic Programming.

*The work of the first two authors was partially supported by JNICT-Portugal and ESPRIT project Compulog 2. The work of the third author was partially supported by the National Science Foundation grant #IRI-9313061.

1. Introduction

Logic programs, deductive databases, and, more generally, non-monotonic theories, use various forms of *default negation*, $not\,A$, whose major distinctive feature is the fact that $not\,A$ is assumed in the absence of *"sufficient evidence"* supporting the formula A. The meaning of *"sufficient evidence"* depends on the specific semantics used. For example, in Reiter's original *Closed World Assumption*, CWA [18], $not\,A$ is assumed if A is not provable, or, equivalently, if there is a minimal model in which A is false. On the other hand, in Minker's *Generalized Closed World Assumption*, $GCWA$ [10, 7], or in McCarthy's Circumscription, $CIRC$ [9], $not\,A$ is assumed only if A is false in *all* minimal models. In Clark's original *Predicate Completion Semantics* [5] for logic programs, this form of negation is called *negation-by-failure* because $not\,A$ is derivable whenever attempts to prove A finitely fail.

For example, the clause:

$$Runs(x) \leftarrow not\, Broken(x)$$

is intended to say that in the absence of "sufficient evidence" that the car is broken, we can use the default belief that it is not broken and thus conclude that it runs. Consequently, if we don't have any additional information (or "evidence") we infer that the car works fine.

While default negation is an inherent part of any commonsense reasoning system and, in particular, it constitutes an important feature of all logic programs and deductive databases, it also often leads to *contradictory information*. This occurs when (subjective) default beliefs clash with the known (objective) information or with some other default beliefs.

Suppose that given the above clause:

$$Runs(x) \leftarrow not\, Broken(x)$$

we find out that the car in fact does not run:

$$\neg Runs(MyCar).$$

The resulting knowledge base seems to be contradictory. On the one hand, given our default assumption that the car is not broken, we should conclude that it runs and yet, on the other hand, we know that it does not run. What should we conclude?

A common-sense approach suggests that in order to avoid such inconsistencies we should refrain from adopting default beliefs that contradict the existing factual information. In this particular case, we could conclude that our initial belief (assumption, hypothesis) that the car is not broken must have been incorrect and thus has to be *revised* or *rejected*. This form of common-sense reasoning is akin to the logical principle of reasoning known as *"reductio ad absurdum"*.

Consider now the following program clause:

$$\neg Broken(x) \leftarrow not\, FlatTire(x), not\, BadBattery(x)$$

which is intended to say that in the absence of any indication that something is wrong with the tires or with the battery we can conclude that the car is not broken. Assuming that this is all that we know about the car, we are likely to conclude that it is not broken because we have no indication that would make us believe that there is any problem with either its battery or tires. In other words, both $FlatTire$ and $BadBattery$ are believed to be false by default.

Suppose, however, that upon inspection we learn that our car is in fact broken, i.e., suppose that we add $Broken(MyCar)$ to our knowledge base. Again, the resulting theory turns out to be inconsistent because we still have no indication of any problem with either battery or tires and thus $not\, FlatTire$ and $not\, BadBattery$ continue to hold true.

As in the previous case, the common-sense approach suggests that in order to avoid such inconsistencies we should refrain from adopting beliefs that contradict the existing factual

information or are mutually contradictory. In this particular case, we could conclude that at least one of our initial default beliefs (assumptions, hypotheses) that the car does not have a flat tire and does not have a bad battery must have been incorrect and thus it has to be *revised* or *rejected*.

However, standard non-monotonic formalisms, such as circumscription, autoepistemic logic and major semantics proposed for logic programs and deductive databases, do not provide any mechanisms for revising or rejecting contradictory beliefs and thus, when faced with similar inconsistencies, they end up in a *contradiction*. In order to remedy this situation, in this paper we investigate the issue of *belief revision*, i.e., the problem of reconciling beliefs with conflicting facts by an appropriate revision of beliefs, and we propose a rather general *belief revision framework* for non-monotonic reasoning. As a byproduct, we obtain a precise description of the nature of the mismatch between facts and beliefs which is shown to have an important application to *diagnosis*.

While rejection of contradictory beliefs may prevent us from deducing contradictory conclusions, simply refraining from believing in certain facts may not be enough as it does not take into account all the consequences of withholding of such beliefs. In order to produce such consequences we must revise the theory itself by adding to it statements that result in the elimination of contradictory beliefs. In other words, we must compile into the theory additional knowledge that prevents the occurrence of the detected belief inconsistency. Accordingly, in this paper we also propose a rather general mechanism for belief revision by means of *theory change*.

Instead of confining our discussion to some narrow class of non-monotonic theories, such as the class of logic programs with some specific semantics, we conduct our study so that it is applicable to a broad class of non-monotonic formalisms. They include the well-known formalisms of circumscription, autoepistemic logic and all the major semantics recently proposed for logic programs, including stable, well-founded, stationary and other semantics.

Specifically, we conduct our study of belief revision within the broad knowledge representation framework of the *AutoEpistemic logic of Beliefs*, *AEB*, introduced by Przymusinski in [15, 17]. *AEB* constitutes a powerful and yet simple unifying framework for non-monotonic reasoning formalisms which was shown to isomorphically contain all of the above mentioned formalisms as special cases[1].

Autoepistemic Logic of Beliefs, *AEB*, has some very natural properties which sharply contrast with those of Moore's Autoepistemic Logic, *AEL*. In particular, *every* belief theory T in *AEB* has the *least* (in the sense of inclusion) static expansion $T^\diamond$ which has an *iterative* definition as the *least fixed point* of a monotonic belief closure operator. Moreover, least static expansions are always consistent in the broad class of *affirmative* belief theories defined later in the paper.

In order to deal with contradictory beliefs, we first introduce the notion of a *careful autoepistemic expansion*, a simple and yet powerful extension of the notion of a static expansion of belief theories, which enables us to incorporate *belief revision* into the framework of *AEB*. When applied to the above example involving a bad battery and flat tires, the proposed approach results in two consistent careful autoepistemic expansions. In one of them we believe that the battery is fine but possibly the tires are not, and, in the other we believe that the tires are fine but possibly the battery is dead. When taken together, the two expansions imply that most likely either the tires or the battery, but not both, are to blame for the car's trouble. They represent therefore an intuitively appealing approach of rejecting those beliefs that contradict factual information, while keeping all the remaining ones intact.

We prove that every consistent belief theory has a consistent careful expansion. This result demonstrates that we can always assign a reasonable set of revised beliefs to any

[1]For simplicity, the class of belief theories considered in this paper does not use the epistemic operator L and thus it does not include Moore's autoepistemic logic, *AEL*, as a special case. However, a simple extension of the discussed framework, described in [15, 17], isomorphically contains *AEL*.

belief theory and underscores the important role played by belief revision in commonsense reasoning. We also show that every consistent static expansion of a belief theory T is also a careful autoepistemic expansion of T and therefore the class of careful expansions extends the class of consistent static expansions. Moreover, for a broad class of affirmative belief theories, defined below, careful expansions coincide with static expansions.

Belief revision based on the notion of a careful autoepistemic expansion can be applied to various reasoning domains. In this paper we illustrate its natural application to the domains of *diagnosis* and *declarative debugging* of logic programs. Here the fact that all consistent theories have consistent careful autoepistemic expansions plays a crucial role because it is imperative that we should be able to derive a reasonable set of conclusions (diagnoses, bugs) from any given knowledge base T even though the observable facts may appear to contradict beliefs resulting from default assumptions contained in T.

Careful autoepistemic expansions represent a form of belief revision in which the rational epistemic agent abstains from believing formulae which, when believed, would lead to a contradiction. However, as we mentioned before, simply refraining from believing in certain formulae does not eliminate the contradictory information present in the knowledge base and it also does not take into account all the consequences of withholding of such beliefs. For example, faced with the fact that the car does not run we may decide to revise our belief that the car is not broken (cf. Example 4.1.). However, that should also compel us to refrain from believing in the related fact that the car does not need to be fixed.

We propose a natural solution to this problem using the previously introduced notion of a careful autoepistemic expansion. The proposed approach is based on the appropriate *revision* of the *original theory* itself instead of just the revision of our beliefs about it. Specifically, we change the theory by adding to it new information that results in the elimination of contradictory beliefs. In other words, we compile into the theory the knowledge that prevents the same belief inconsistencies from occurring again.

In some application domains, beliefs may logically depend on other beliefs, which may be viewed as more basic and sometimes considered to be non-revisable. For example, this is true when diagnosing faults in a device: causally deeper component faults are sometimes preferred over surface faults, that are simply consequences of the former. In such cases, one may want to control the level at which diagnosis is performed, by eliminating diagnoses which do not focus on the causally deeper faults. In declarative debugging, one may know in advance that some predicates are specified correctly (e.g., those that are part of a previously debugged program) so that any observable bugs involving these predicates must necessarily be caused by the incorrect specification of the remaining predicates. More generally, any revision of beliefs should comply with any given specification of mutual dependency of beliefs. We illustrate how one can express such dependencies in AEB by means of the so called *Belief Completion Clauses*. These clauses essentially state that a revision of some beliefs requires a revision of beliefs on which they logically depend.

Because of its generality, this method of specifying the logical level of revision in belief theories can be employed to explain and justify, via embedding of logic programs into AEB, the meta-linguistic devices used for controlling abduction, view updates, declarative debugging and contradiction removal in logic programs. Moreover, the fact that it is expressible in the object-language, rather than in some meta-language, leads to a computationally simpler solution. In particular, in [3] we prove that the *contradiction removal semantics* for non-disjunctive extended logic programs, introduced by the first two authors in [13, 12], can be isomorphically embedded into the more general framework of the Autoepistemic Logic of Beliefs.

2. Autoepistemic Logic of Beliefs

We first briefly recall the definition and basic properties of the *Autoepistemic Logic of Beliefs*, AEB. The language of AEB, is a propositional modal language, $\mathcal{K}_\mathcal{B}$, with standard connectives ($\vee$, $\wedge$, $\supset$, $\neg$), the propositional letter $\bot$ (denoting *false*) and a modal operator $\mathcal{B}$, called the *belief* operator. The atomic formulae of the form $\mathcal{B}F$, where F is an arbitrary formula of $\mathcal{K}_\mathcal{B}$, are called *belief atoms*. The formulae of $\mathcal{K}_\mathcal{B}$ in which $\mathcal{B}$ does not occur are called *objective* and the set of all such formulae is denoted by $\mathcal{K}$. Any theory T in the language $\mathcal{K}_\mathcal{B}$ is called an *autoepistemic theory of beliefs*, or, briefly, a *belief theory*.

Definition 2.1. [Belief Theory] By an *autoepistemic theory of beliefs*, or just a *belief theory*, we mean an arbitrary theory in the language $\mathcal{K}_\mathcal{B}$, i.e., a (possibly infinite) set of arbitrary clauses of the form:

$$B_1 \wedge ... \wedge B_k \wedge \mathcal{B}G_1 \wedge ... \wedge \mathcal{B}G_l \wedge \neg\mathcal{B}F_1 \wedge ... \wedge \neg\mathcal{B}F_n \supset A_1 \vee ... \vee A_m$$

where $k, l, m, n \geq 0$, A_is and B_is are objective atoms and F_is and G_is are arbitrary formulae of $\mathcal{K}_\mathcal{B}$. Such a clause says that if the B_is are true, the G_is are believed, and the F_is are not believed then one of the A_is is true.

By an *affirmative belief theory* we mean any belief theory all of whose clauses satisfy the condition that $m > 0$. In other words, affirmative belief theories are precisely those belief theories that satisfy the condition that all of their clauses contain at least one *objective* atom in their heads[2]

Observe that arbitrarily deep level of *nested beliefs* is allowed in belief theories. We assume the following two simple axiom schemata and one inference rule describing the arguably obvious properties of belief atoms:

(D) Consistency Axiom:

$$\neg\mathcal{B}\bot \tag{1}$$

(K) Normality Axiom: For any formulae F and G:

$$\mathcal{B}(F \supset G) \supset (\mathcal{B}F \supset \mathcal{B}G) \tag{2}$$

(N) Necessitation Rule: For any formula F:

$$\frac{F}{\mathcal{B}F} \tag{3}$$

The first axiom states that tautologically false formulae are *not* believed. The second axiom states that if we believe that a formula F implies a formula G and if we believe that F is true then we believe that G is true as well. The necessitation inference rule states that if a formula F has been proven to be true then F is believed to be true.

Definition 2.2. [Formulae Derivable from a Belief Theory]
For any belief theory T, we denote by $Cn_*(T)$ the smallest set of formulae of the language $\mathcal{K}_\mathcal{B}$ which contains the theory T, all the (substitution instances of) the axioms (K) and (D) and is closed under standard propositional consequence and under the necessitation rule (N).

We say that a formula F is *derivable* from theory T in the logic AEB if F belongs to $Cn_*(T)$. We denote this fact by $T \vdash_* F$. We call a belief theory T *consistent* if the theory $Cn_*(T)$ is consistent. Consequently, $Cn_*(T) = \{F : T \vdash_* F\}$. Moreover, T is consistent if and only if $T \nvdash_* \bot$.

[2]More precisely, we require that all clauses contain at least one *positive objective* atom in their heads. Later, we introduce *negative objective atoms*, namely, the so called "strong negation" and "explicit negation" atoms.

Remark 2.1. It is easy to see that, in the presence of the axiom (K), the axiom (D) is equivalent [17] to the axiom:

$$\mathcal{B}F \supset \neg\mathcal{B}\neg F. \tag{4}$$

stating that if we believe in a formula F then we do *not* believe in $\neg F$.

For readers familiar with modal logics it should be clear by now that we are, in effect, considering here a *normal* modal logic with one modality $\mathcal{B}$ which satisfies the consistency axiom (D) [8]. The axiom (K) is called "normal" because all normal modal logics satisfy it [8].

2.1. Intended Meaning of Belief Atoms

In general, belief atoms $\mathcal{B}F$ can be given different intended meanings. In this paper, the intended meaning of belief atoms $\mathcal{B}F$ is based on Minker's *GCWA* (see [10, 7]) or McCarthy's *Predicate Circumscription* [9], and is described by the principle of *predicate minimization*:

$$\mathcal{B}F \equiv F \text{ is minimally entailed} \equiv F \text{ is true in all minimal models.}$$

Accordingly, beliefs considered in this paper can be called *minimal beliefs*.

We now give a precise definition of minimal models and minimal entailment. Throughout the paper we represent *models* as (consistent) *sets of literals*. An atom A is *true* in a model M if and only if A belongs to M. An atom A is *false* in a model M if and only if $\neg A$ belongs to M. A model M is *total* if for every atom A either A or $\neg A$ belongs to M. Otherwise, the model is called *partial*. Unless stated otherwise all models are assumed to be total. A (total) model M is *smaller* than a (total) model N if it contains fewer positive literals (atoms). For convenience, when describing models we usually list *only* those of their members that are *relevant* to our considerations, typically those whose predicate symbols appear in the theory that we are currently discussing.

Definition 2.3. [Minimal Models][15, 17] By a *minimal model* of a belief theory T we mean a model M of T with the property that there is *no* smaller model N of T which coincides with M on belief atoms $\mathcal{B}F$. If a formula F is true in all minimal models of T then we write: $T \models_{\min} F$ and say that F is *minimally entailed* by T.

For readers familiar with *circumscription*, this means that we are considering predicate circumscription $CIRC(T;\mathcal{K})$ of the theory T in which atoms from the objective language $\mathcal{K}$ are minimized while the belief atoms $\mathcal{B}F$ are fixed:

$$T \models_{\min} F \equiv CIRC(T;\mathcal{K}) \models F.$$

In other words, minimal models are obtained by first assigning *arbitrary* truth values to the belief atoms and then *minimizing* objective atoms.

2.2. Static Autoepistemic Expansions

Like in Moore's Autoepistemic Logic, also in the Autoepistemic Logic of Beliefs we introduce sets of beliefs that an ideally rational and introspective agent may hold, given a set of premises T. We do so by defining *static autoepistemic expansions* $T^\diamond$ of T, which constitute plausible sets of such rational beliefs.

Definition 2.4. [Static Autoepistemic Expansion] [15, 17] A belief theory $T^\diamond$ is called a *static autoepistemic expansion* of a belief theory T if it satisfies the following fixed-point equation:

$$T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F\}),$$

where F ranges over all formulae of $\mathcal{K}_\mathcal{B}$.

The definition of static autoepistemic expansions is based on the idea of building an expansion $T^\diamond$ of a belief theory T by closing it with respect to: (i) the derivability in the logic AEB, and, (ii) the addition of belief atoms $\mathcal{B}F$ satisfying the condition that the formula F is minimally entailed by $T^\diamond$. Consequently, the definition of static expansions *enforces* the intended meaning of belief atoms described above. Note that negations $\neg\mathcal{B}F$ of the remaining belief atoms are not *explicitly* added to the expansion although some of them will be forced in by the Normality and Consistency Axioms (2) and (1).

Definition 2.5. [Static Semantics] By the *(skeptical) static semantics* of a belief theory T we mean the set of all formulae that belong to all static autoepistemic expansions $T^\diamond$ of T.

Every belief theory T in AEB has the *least* (in the sense of set-theoretic inclusion) static expansion $T^\diamond$ which has an *iterative* definition as the *least fixed point* of the monotonic belief closure operator Ψ_T defined below.

Definition 2.6. [Belief Closure Operator] [15, 17] For any belief theory T define the *belief closure* operator Ψ_T by the formula:

$$\Psi_T(S) = Cn_*(T \cup \{\mathcal{B}F : S \ \models_{\min} F\}),$$

where S is an arbitrary belief theory and the F's range over all formulae of $\mathcal{K}_\mathcal{B}$.

Thus $\Psi_T(S)$ augments the theory T with all those belief atoms $\mathcal{B}F$ with the property that F is minimally entailed by S. It is easy to see that a theory $T^\diamond$ is a static autoepistemic expansion of the belief theory T in AEB if and only if $T^\diamond$ is a fixed point of the operator Ψ_T, i.e. if $T^\diamond = \Psi_T(T^\diamond)$.

Theorem 2.1. [Least Static Expansion][15, 17] *Every belief theory T in AEB has the* least *static expansion, namely, the least fixed point $T^\diamond$ of the monotonic belief closure operator Ψ_T.*

Moreover, the least static expansion $T^\diamond$ of a belief theory T can be constructed as follows. Let $T^0 = Cn_*(T)$ and suppose that T^α has already been defined for any ordinal number $\alpha < \beta$. If $\beta = \alpha + 1$ is a successor ordinal then define:

$$T^{\alpha+1} = \Psi_T(T^\alpha) = Cn_*(\ T \cup \{\mathcal{B}F : T^\alpha \ \models_{\min} \ F\}\ ),$$

where F ranges over all formulae in $\mathcal{K}_\mathcal{B}$. Else, if β is a limit ordinal then define $T^\beta = \bigcup_{\alpha<\beta} T^\alpha$.

The sequence $\{T^\alpha\}$ is monotonically increasing and has a unique fixed point $T^\diamond = T^\lambda = \Psi_T(T^\lambda)$, for some ordinal λ. For finite theories T the fixed point $T^\diamond$ is reached after finitely many steps.

Observe that the *least* static autoepistemic expansion $T^\diamond$ of T contains therefore those and only those formulae which are true in *all* static autoepistemic expansions of T and therefore it always coincides with the static semantics of T. It is easy to verify that a belief theory T either has a *consistent* least static expansion $T^\diamond$ or it does *not* have any consistent static expansions at all. Moreover, least static expansions of *affirmative* belief theories are always consistent [15, 17].

Example 2.1. Consider the following belief theory T:

$$Car$$
$$Car \wedge \mathcal{B}\neg Broken \supset Runs$$

For simplicity, when describing static expansions of this and other examples we list only those elements of the expansion that are "relevant" to our discussion. In particular, we usually omit nested beliefs. In order to iteratively compute the least static expansion $T^\diamond$ of

T we first let $T^0 = Cn_*(T)$. Let us observe that $T^0 \models Car$ and $T^0 \models_{min} \neg Broken$. Indeed, in order to find minimal models of T^0 we need to assign an *arbitrary* truth value to the only belief atom $\mathcal{B}\neg Broken$, and then *minimize* the objective atoms $Broken$, Car and $Runs$. We easily see that T^0 has the following two minimal models (truth values of the remaining belief atoms are irrelevant and are therefore omitted):

$$M_1 = \{\mathcal{B}\neg Broken, Car,\ Runs,\ \neg Broken\};$$
$$M_2 = \{\neg\mathcal{B}\neg Broken, Car,\ \neg Runs,\ \neg Broken\}.$$

Since in both of them Car is true, and $Broken$ is false, we deduce that $T^0 \models_{min} Car$ and $T^0 \models_{min} \neg Broken$. Consequently, since $T^1 = \Psi_T(T^0) = Cn_*(T \cup \{\mathcal{B}F : T^0 \models_{\min} F\})$, we obtain:

$$T^1 = Cn_*(T \cup \{\mathcal{B}Car, \mathcal{B}\neg Broken\}).$$

Since $T^1 \models Runs$ and $T^2 = \Psi_T(T^1) = Cn_*(T \cup \{\mathcal{B}F : T^1 \models_{\min} F\})$, we obtain:

$$T^2 = Cn_*(T \cup \{\mathcal{B}Car, \mathcal{B}\neg Broken, \mathcal{B}Runs\}).$$

It is easy to check that $T^2 = \Psi_T(T^2)$ is a fixed point of Ψ_T and therefore $T^\diamond = T^2 = Cn_*(T \cup \{\mathcal{B}Car, \mathcal{B}\neg Broken, \mathcal{B}Runs\})$ is the least static expansion of T. The static semantics of T asserts our belief that the car is not broken and thus runs fine. One easily verifies that T does not have any other (consistent) static expansions.

2.3. Logic Programs as Belief Theories

One can easily show that Circumscription is properly embeddable into the Autoepistemic Logic of Beliefs, AEB. In [15, 17] it was also shown that major semantics defined for normal and disjunctive *logic programs* are also embeddable into AEB. In particular, this is true for the well-founded, stable and stationary (or partial stable) semantics of normal logic programs. In the next section we recall an analogous result for the stable semantics of extended logic programs with so called "classical negation" [6].

Suppose that P is a normal logic program consisting of rules:

$$A \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, C_1 \wedge \ldots \wedge not\, C_n$$

The translation of P into the affirmative belief theory $T_{\mathcal{B}\neg}(P)$ is given by the set of the corresponding clauses:

$$B_1 \wedge \ldots \wedge B_m \wedge \mathcal{B}\neg C_1 \wedge \ldots \wedge \mathcal{B}\neg C_n \supset A \tag{5}$$

obtained by replacing the non-monotonic negation $not\, F$ by the belief atom $\mathcal{B}\neg F$, and by replacing the rule symbol $\rightarrow$ by the standard material implication $\supset$.

The translation, $T_{\mathcal{B}\neg}(P)$, gives therefore the following meaning to the non-monotonic negation:

$$not\, F \stackrel{def}{\equiv} \mathcal{B}\neg F \equiv F \text{ is believed to be false} \equiv \neg F \text{ is minimally entailed.} \tag{6}$$

Theorem 2.2. [Embeddability of Stationary and Stable Semantics][15, 17] *There is a one-to-one correspondence between stationary (or, equivalently, partial stable) models $\mathcal{M}$ of the program P and consistent static autoepistemic expansions $T^\diamond$ of its translation $T_{\mathcal{B}\neg}(P)$ into a belief theory. Namely, for any objective atom A we have:*

$$A \in \mathcal{M} \quad iff \quad \mathcal{B}A \in T^\diamond$$
$$\neg A \in \mathcal{M} \quad iff \quad \mathcal{B}\neg A \in T^\diamond.$$

In particular, the well-founded model $\mathcal{M}_0$ of the program P corresponds to the least static expansion of $T_{\mathcal{B}\neg}(P)$. Moreover, (total) stable models (or answer sets) $\mathcal{M}$ of P correspond to those consistent static autoepistemic expansions $T^\diamond$ of $T_{\mathcal{B}\neg}(P)$ that satisfy the condition that for all objective atoms A, either $\mathcal{B}A \in T^\diamond$ or $\mathcal{B}\neg A \in T^\diamond$.

Example 2.2. It is easy to see that the belief theory T considered in Example 2.1. can be viewed as a translation $T_{\mathcal{B}\neg}(P)$ of the logic program P given by:

$$Car$$
$$Runs \leftarrow Car \wedge notBroken$$

The unique consistent static expansion,

$$T^{\diamond} = Cn_{*}(T \cup \{\mathcal{B}Car, \mathcal{B}\neg Broken, \mathcal{B}Runs\})$$

of T corresponds therefore to the unique stationary (or stable) model,

$$M = \{Car, \neg Broken, Runs\}$$

of P, which is also its unique well-founded model [14].

2.4. Strong Negation

Classical negation, $\neg A$, which is part of the propositional language $\mathcal{K}_{\mathcal{B}}$ of the Autoepistemic Logic of Beliefs, AEB, satisfies the so called *law of the excluded middle*, $A \vee \neg A$, which requires that any given property A be known to be either true or false. However, in many commonsense reasoning domains, such a requirement appears undesirable. In particular, this is the case in logic programming [6, 16, 1]. Consequently, we need a new notion of negation, which does not necessarily satisfy the law of the excluded middle.

In [15, 17], we showed that one form of such non-standard negation, called *strong negation*, can be easily added to the autoepistemic logic of beliefs, AEB, by:

- augmenting the original objective language $\mathcal{K}$ with new *objective* propositional symbols $\sim A$, called *strong negation atoms*, resulting in a new objective language $\mathcal{K}'$ and the new language of beliefs $\mathcal{K}'_{\mathcal{B}}$.
- ensuring that the intended meaning of $\sim A$ is "$\sim A$ *is the opposite of* A" by assuming the following *strong negation* axiom:

  (S) $\quad A \wedge \sim A \supset \bot \quad$ or, equivalently, $\quad \sim A \supset \neg A,$

  which says that A and its opposite $\sim A$ cannot be both true. Formally, the addition of the axiom schema (S) means that the set $Cn_{*}(T)$ of formulae derivable from a given belief theory T, used in the definition of the static expansion, is now replaced by the smallest set, $Cn^{s}_{*}(T)$, which contains the theory T and all the (substitution instances of) the axioms (K), (D) and (S) and is closed under the necessitation rule (N).

For example, a proposition A may describe the property of being *"good"* while the proposition $\sim A$ describes the property of being *"bad"*. The strong negation axiom states that things cannot be *both* good and bad. We do not assume, however, that things must always be either good or bad.

Example 2.3. Consider the belief theory T with strong negation:

$$\sim Football \qquad \mathcal{B}\neg Baseball \supset Football$$
$$\mathcal{B}\neg Football \supset Baseball$$

It is easy to verify that T has precisely one consistent static expansion:

$$T^{\diamond} = Cn_{*}(T \cup \{\mathcal{B}\neg Football,\ \mathcal{B}Baseball.\})$$

Indeed, axiom (S) implies that $T^0 \models \neg Football$ and thus $T^1 \models \mathcal{B}\neg Football$ and, consequently, $T^1 \models Baseball$ and $T^{\diamond} = T^2 \models \mathcal{B}Baseball.$

As the following result shows, we can use strong negation to translate extended logic programs with "classical negation", originally introduced in [6], into belief theories.

Theorem 2.3. [Embeddability of Extended Stationary and Stable Semantics] [15, 17] *There is a one-to-one correspondence between stationary (or partial stable) models $\mathcal{M}$ of an extended logic program P with "classical negation", as defined in [16], and consistent static autoepistemic expansions $T^\diamond$ of its translation $T_{\mathcal{B}\neg}(P)$ into belief theory, in which "classical negation" of an atom A is translated into $\sim A$.*

In particular, (total) stable models (or answer sets) $\mathcal{M}$ of P, as defined in [6], correspond to those consistent static autoepistemic expansions $T^\diamond$ of $T_{\mathcal{B}\neg}(P)$ that satisfy the condition that for all objective atoms A, either $\mathcal{B}A \in T^\diamond$ or $\mathcal{B}\neg A \in T^\diamond$.

Since the axiom (S) has no effect on those belief theories T that do not include strong negation atoms $\sim A$, in the sequel we will assume the axiom (S) without any further mention whenever strong negation is used. For a more detailed study of strong negation the reader is referred to [3].

2.5. Explicit Negation

In some commonsense reasoning domains, even the strong negation axiom (S) appears to be too strong [2, 3], and is thus replaced by the following *explicit negation inference rule*:

$$(ER) \qquad \frac{\sim A}{\mathcal{B}\neg A}$$

and the following *explicit negation axiom*:

$$\text{(EA)} \qquad \mathcal{B}A \wedge \mathcal{B}\sim A \supset \bot$$

for every atom A in $\mathcal{K}$. Both of these assumptions can be shown [3] to be weaker than the strong negation axiom (S).

Formally, the addition of the inference rule (ER) and the axiom (EA) (instead of the axiom (S)) means that the set $Cn_*(T)$ of formulae derivable from a given belief theory T, used in the definition of the static expansion, is now replaced by the smallest set, $Cn^e_*(T)$, which contains the theory T and all the (substitution instances of) the axioms (K), (EA) and (D), and is closed under both the necessitation rule (N) and the explicit negation rule (ER). Both strong and explicit negations can be easily generalized to arbitrary, non atomic, formulae [3].

In order to avoid confusion between strong and explicit negation, from now on we will denote the explicit negation of an atom A by $\overline{A}$ instead of $\sim A$. While the intended meaning of the strong negation $\sim A$ of A is *"the opposite of A"*, the intended meaning of explicit negation $\overline{A}$ is *"there is evidence against A"*. In particular, since the strong negation axiom (S) does not hold for explicit negation, it is possible to have both A (i.e., *"evidence for A"*) and $\overline{A}$ (i.e., *"evidence against A"*) in the same model of a belief theory. Having evidence both for and against a given proposition occurs frequently in common-sense reasoning.

Since the explicit negation inference rule (ER) and the axiom (EA) have no effect on those belief theories T that do not include explicitly negated atoms $\overline{A}$, in the sequel we will assume them both without any further mention whenever explicit negation is used. As the following result shows, we can use belief theories with explicit negation to obtain the well-founded semantics with explicit negation originally defined in [11]:

Theorem 2.4. [Embeddability of WFSX Semantics] [11] *There is a one-to-one correspondence between the partial stable models $\mathcal{M}$ of an extended logic program P with "explicit negation", as defined in [11], and the consistent static autoepistemic expansions $T^\diamond$ of its translation $T_{\mathcal{B}\neg}(P)$ into belief theory, where "explicit negation" of an atom A is translated into $\overline{A}$.*

For a more detailed study of explicit negation the reader is referred to [3].

3. Belief Revision

While static expansions seem to provide a natural and intuitive semantics for many (consistent) belief theories (in particular, for all affirmative belief theories) they often lead to inconsistent expansions for theories in which (subjective) beliefs clash with the observable (objective) facts or with some other beliefs. In particular, this applies to belief theories and logic programs with strong (or explicit) negation.

Example 3.1. Consider again the simple belief theory introduced in Example 2.1.. As we have seen, its static semantics implies that we believe that the car is not broken and thus runs fine. Suppose, however, that upon inspection we found out that the car actually *does not* run:

$$\neg Runs.$$

It is clear that the resulting new belief theory does not have any consistent static expansions. Indeed, since there is no evidence that the car is broken, $Broken$ is false in all minimal models and thus $\mathcal{B}\neg Broken$ is derivable. This implies $Runs$ and thus results in a contradiction. In other words, our belief that the car is not broken and thus runs, based on the fact that there is no evidence to the contrary, apparently contradicts the objective fact that the car does not run.

In view of the contradictory factual information that the car does not run, we could very well conclude that our initial belief (assumption) that the car is not broken must have been incorrect and thus has to be *revised* and *rejected*.

Example 3.2. Consider now the belief theory discussed in the introduction:

$$\mathcal{B}\neg FlatTire \wedge \mathcal{B}\neg BadBattery \supset \neg Broken$$
$$Broken,$$

which says that, in the absence of any indication that something is wrong with the tires or with the battery, we can safely conclude that the car is not broken, and yet the fact is that it is broken. This theory, again, does not have any consistent static expansions because both $\neg FlatTire$ and $\neg BadBattery$ are minimally entailed and thus the premise $\mathcal{B}\neg FlatTire \wedge \mathcal{B}\neg BadBattery$ is derivable. This implies $\neg Broken$ and results in a contradiction.

Again, a natural way to remedy this problem is to conclude that, in view of the contradictory objective information that the car is broken, at least one of our initial beliefs (assumptions) that the car does not have a flat tire and does not have a bad battery must have been incorrect and thus has to be *revised* and *rejected*.

3.1. Careful Autoepistemic Expansions

The approach illustrated in the previous two examples is based on the idea of *rejecting* or *revising* beliefs that contradict the existing factual information or are mutually contradictory. It leads to a simple modification of the definition of static expansions which results in a natural and potent framework for belief revision in AEL.

Definition 3.1. [Careful Autoepistemic Expansion] A belief theory $T^\diamond$ is called a *careful autoepistemic expansion* of a belief theory T if it satisfies the following fixed-point equation:

$$T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F \text{ and } T^\diamond \cup \{\mathcal{B}F\} \text{ is consistent}\}),$$

where F ranges over all formulae of $\mathcal{K}_\mathcal{B}$.

The only difference between the definition of static expansions and careful expansions is the requirement that only those belief atoms $\mathcal{B}F$ should be added to the expansion whose addition does not lead to a contradiction. Recall that, by definition, $T^\diamond \cup \{\mathcal{B}F\}$ is consistent if and only if $Cn_*(T^\diamond \cup \{\mathcal{B}F\})$ is consistent.

Example 3.3. It is easy to see that the theory considered in Example 3.1. has precisely one careful expansion, namely $T^\diamond = Cn_*(T \cup \{\mathcal{B}Car, \mathcal{B}\neg Runs\})$, which does not include any beliefs about the car being broken and corresponds therefore to the intuitive approach of rejecting beliefs that contradict existing factual information.

Example 3.4. On the other hand, the theory considered in Example 3.2. has precisely two careful expansions namely:

$$T_1^\diamond = Cn_*(T \cup \{\mathcal{B}Broken, \mathcal{B}\neg FlatTire\}),$$
$$T_2^\diamond = Cn_*(T \cup \{\mathcal{B}Broken, \mathcal{B}\neg BadBattery\}),$$

which reflect the fact that one of the assumptions about not having a bad battery or not having a flat tire has to be rejected while the other can be kept without causing any inconsistency. The resulting semantics implies therefore $\mathcal{B}\neg FlatTire \vee \mathcal{B}\neg BadBattery$ and thus suggests that most likely the car does not have both a bad battery and a flat tire. It represents the intuitively appealing approach of rejecting only those beliefs that contradict factual information, while keeping all the remaining ones intact.

Example 3.5. Suppose that our belief theory T simply says: $\mathcal{B}God$. Clearly, T does not have any consistent static expansions. Indeed, since it does not offer any evidence for the existence of *God*, it yields $\mathcal{B}\neg God$ resulting, by virtue of the Consistency Axiom (D), in a contradiction with $\mathcal{B}God$. However, T has precisely one consistent careful static expansion which coincides with the closure $Cn_*(T)$ of T in AEB. The belief $\mathcal{B}\neg God$ is not added because it leads to inconsistency.

As the previous examples demonstrate, careful autoepistemic expansions no longer lead to inconsistencies when we add to our knowledge facts that seem to contradict our (default) beliefs.

Proposition 3.1. All careful expansions of a consistent belief theory are consistent.

Proof:
Let T be a belief theory and let $T^\diamond$ be a careful expansion of T. By definition,

$$T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F \text{ and } T^\diamond \cup \{\mathcal{B}F\} \text{ is consistent}\}).$$

Consequently, if $T^\diamond$ were inconsistent then we would have $T^\diamond = Cn_*(T)$ and thus T would also have to be inconsistent.

It turns out that every consistent belief theory has a consistent careful autoepistemic expansion.

Theorem 3.1. [Fundamental Theorem of Belief Revision] *Every consistent belief theory has a consistent careful autoepistemic expansion.*

Proof:
Let $\prec$ be some well-ordering of the set of all formulae of the propositional modal language, $\mathcal{K}_\mathcal{B}$, and, let T be a consistent belief theory. By definition, the theory $T_0 = Cn_*(T)$ is consistent and closed under the axioms (D) and (K) as well as the necessitation rule (N). Suppose that β is an ordinal such that for all $\alpha < \beta$ consistent and non-decreasing theories T_α have been constructed which are closed under the axioms (D) and (K) and the necessitation rule (N).

If β is a limit ordinal then we define $T_\beta = \bigcup_{\alpha<\beta} T_\alpha$. Due to the compactness theorem, the theory T_β is also consistent and closed under the axioms (D) and (K) and the necessitation rule (N). If $\beta = \alpha + 1$ is a successor ordinal then we choose the $\prec$-least formula F with the property that:

- $T_\alpha \models_{\min} F$,
- $T_\alpha \cup \{\mathcal{B}F\}$ is consistent,
- $\mathcal{B}F \notin T_\alpha$,

assuming such a formula F exists, and we define $T_{\alpha+1} = Cn_*(T_\alpha \cup \{\mathcal{B}F\})$. Otherwise, we define $T_{\alpha+1} = T_\alpha$.

The so constructed transfinite sequence of theories is non-decreasing and therefore there must exist a γ such that $T_{\gamma+1} = T_\gamma$ is a fixed point. Define $T^\diamond = T_\gamma$. We have to show that:

$$T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F \text{ and } T^\diamond \cup \{\mathcal{B}F\} \text{ is consistent}\}).$$

Clearly, $T^\diamond \supseteq Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F$ and $T^\diamond \cup \{\mathcal{B}F\}$ is consistent$\})$. It suffices therefore to show that $T^\diamond \subseteq Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F$ and $T^\diamond \cup \{\mathcal{B}F\}$ is consistent$\})$. For that purpose it is enough to prove that, for any ordinal α, if $T_\alpha \models_{\min} F$ then $T^\diamond \models_{\min} F$.

Suppose that $T_\alpha \models_{\min} F$ and let M be a minimal model of $T^\diamond$. If M were not a minimal model of T_α then there would exist a smaller model N of T_α which coincides with M on all belief atoms. However, since $T^\diamond$ differs from T_α only by the addition of some belief atoms, this means that N would also be a smaller model of $T^\diamond$, which is impossible. This shows that M is a minimal model of T_α and thus F must be true in M. Consequently, $T^\diamond \models_{\min} F$ which completes the proof.

This result demonstrates that we can always assign a reasonable set of revised beliefs to any belief theory and thus underscores the important role played by *belief revision* in commonsense reasoning. It is also of crucial importance in applications of belief revision, such as the application to *diagnosis* illustrated below, where it is imperative that we should be able to derive a reasonable set of conclusions (diagnoses) from any given knowledge base T even though the observable facts may appear to contradict beliefs resulting from default assumptions contained in T.

The class of careful expansions extends the class of consistent static expansions. Moreover, for affirmative belief theories, the notions of a consistent static expansion and a careful expansion coincide.

Theorem 3.2. *Every consistent static expansion of a belief theory T is also a careful expansion of T.*

Proof:
Let T be a belief theory and let $T^\diamond$ be a consistent static expansion of T. By definition,

$$T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F\}).$$

Since $T^\diamond$ is consistent, $T^\diamond \cup \{\mathcal{B}F\}$ is consistent, for every F such that $T^\diamond \models_{\min} F$. Consequently, $T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F$ and $T^\diamond \cup \{\mathcal{B}F\}$ is consistent$\})$, which shows that $T^\diamond$ is a careful expansion of T.

Theorem 3.3. *For affirmative belief theories, the notions of a consistent static expansion and a careful expansion coincide.*

Proof:
¿From Theorem 3.2. it follows that all consistent static expansions are also careful expansions. We need to show that every careful expansion of an affirmative theory T is a consistent static expansion of T.

Let T be an affirmative belief theory and let $T^\diamond$ be a careful expansion of T. Since affirmative belief theories are consistent [15, 17], it follows from Proposition 3.1. that $T^\diamond$ is consistent. By definition, $T^\diamond = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F$ and $T^\diamond \cup \{\mathcal{B}F\}$ is consistent$\})$. It suffices to show that $T^\diamond \cup \{\mathcal{B}F\}$ is consistent for every F such that $T^\diamond \models_{\min} F$. For that purpose it suffices to prove that $T^* = Cn_*(T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F\})$ is consistent. We

first show that $T_0^* = T \cup \{\mathcal{B}F : T^\diamond \models_{\min} F\} \cup (K)$ is consistent as a standard propositional theory, where (K) represents all instances of the normality axiom. Let M be an interpretation in which all objective atoms A are true and those and only those belief atoms $\mathcal{B}F$ are true for which $T^\diamond \models_{\min} F$. Since T is affirmative all the clauses of T are satisfied in M. Moreover, all instances of the normality axiom (K) are satisfied as well because if $\mathcal{B}(F \supset G)$ and $\mathcal{B}(F)$ are true in M then $T^\diamond \models_{\min} (F \supset G) \wedge F$ and thus $T^\diamond \models_{\min} G$ and consequently $\mathcal{B}(G)$ is also true in M. This shows that T_0^* is consistent as a standard propositional theory.

Suppose that a consistent theory T_n^* has been already defined and let

$$T_{n+1}^* = T_n^* \cup \{\mathcal{B}F : T_n^* \models F\}.$$

An analogous argument shows that T_{n+1}^* is consistent as a standard propositional theory. Clearly, T^* is the fixed point $T_n^* = T_{n+1}^*$ of this sequence of theories and therefore it is also consistent, which completes the proof.

3.2. Application to Diagnosis

Belief revision based on the notion of a careful autoepistemic expansion can be applied to various reasoning domains. Below we illustrate its application to the domain of diagnosis.

For any careful expansion $T^\diamond$ of a belief theory T the set $\mathcal{R}(T^\diamond) = \{F : T^\diamond \models_{min} F$ and yet $\mathcal{B}F \notin T^\diamond\}$, namely, the set of those formulae F which should be believed in (because F is minimally entailed) in the expansion $T^\diamond$, and yet are not believed in $T^\diamond$ (because of the resulting inconsistency), plays an important diagnostic role by constituting the set of *possibly false assumptions*.

Definition 3.2. [Revision Set of a Careful Expansion] The revision set $\mathcal{R}(T^\diamond)$ of the careful autoepistemic expansion $T^\diamond$ of a belief theory T is defined by:

$$\mathcal{R}(T^\diamond) = \{F : T^\diamond \models_{\min} F \quad \text{and} \quad \mathcal{B}F \notin T^\diamond\}.$$

Clearly, a careful expansion is a static expansion if and only if its revision set is empty.

Example 3.6. Consider the careful expansions of the theory discussed in Example 3.2.:

$$T_1^\diamond = Cn_*(T \cup \{\mathcal{B}Broken,\ \mathcal{B}\neg FlatTire\})$$
$$T_2^\diamond = Cn_*(T \cup \{\mathcal{B}Broken,\ \mathcal{B}\neg BadBattery\})$$

Their revision sets are:

$$\mathcal{R}(T_1^\diamond) = \{\neg BadBattery\}$$
$$\mathcal{R}(T_2^\diamond) = \{\neg FlatTire\}$$

i.e. in $T_1^\diamond$ we refrain from believing $\neg BadBattery$, while in $T_2^\diamond$ we refrain from believing $\neg FlatTire$. As a result, the first revision set suggests that our assumption that the car does not have a bad battery may have been wrong and the second revision set suggests that our assumption that the car does not have a flat tire may have been incorrect. Both of them together provide us with a useful diagnosis of possible reasons why the car does not work.

4. Belief Revision by Theory Change

In this section we study the issue of belief revision by theory revision, as opposed to belief revision by rejection of contradictory beliefs which was discussed in the previous section.

As remarked earlier, careful autoepistemic expansions represent a form of belief revision where the rational epistemic agent abstains from believing formulae which, if believed, would lead to contradiction. However, simply refraining from believing in certain formulae is often not enough, as it does not fully take into account all the consequences of withholding such

beliefs. In order to produce such consequences we must *revise* the theory by adding to it some statements that justify not holding the contradictory beliefs. In other words, we must compile into the theory *additional* knowledge that will prevent the detected belief inconsistency from occurring. This knowledge is gathered by analyzing the causes of inconsistencies.

Example 4.1. Suppose that to the theory of Example 3.1. we add:

$$Car \wedge Broken \supset FixIt$$

It is easy to check that the resulting theory T has a single careful expansion:

$$T^\diamond = Cn_* (T \cup \{\mathcal{B}Car, \mathcal{B}\neg Runs, \mathcal{B}\neg FixIt\})$$

Even though $\neg Broken$ is true in all minimal models of the expansion, $\mathcal{B}\neg Broken$ is not added since it leads to inconsistency. Since $Broken$ is no longer believed to be false, one would intuitively expect $\neg FixIt$ not to be believed either. However, this is not the case in the careful autoepistemic expansion above. Indeed, the expansion reflects only the fact that the agent must refrain from believing formulae that lead to contradiction. It does not invalidate the reasons that have led to such beliefs. In our example, we believed in the car not being broken because of the lack of evidence showing otherwise. This lack of evidence must therefore be invalidated by admitting the possibility that the car might in fact be broken.

This is the stance taken by most belief revision systems, where the outcome of revision is a modified theory, in which contradiction is avoided by eliminating the reasons for contradictory beliefs. It is clear that the only way of inhibiting $\neg Broken$ from being believed in static expansions is by introducing some evidence for $Broken$ to be true. This evidence could, for example, be stated in the form that $Broken$ is in fact true. However, this appears too strong: absence of belief in $\neg Broken$ does not warrant jumping to such a conclusion. We need only the weaker statement that $Broken$ is *possible*, i.e. there is at least one minimal model with $Broken$, so that, given our notion of evidence, there is no longer absence of evidence for the car not being broken. In such case, we would no longer believe $\neg Broken$. Moreover, $\neg FixIt$ would no longer be minimally entailed, and thus would no longer be believed.

Careful expansions already identify and inhibit the addition of beliefs that lead to contradiction. It is thus an easy matter to determine which sets of formulae do lead to contradiction: they are the revision sets $\mathcal{R}(T^\diamond)$ of careful expansions $T^\diamond$.

4.1. Revised Autoepistemic Expansions

Given a careful expansion $T^\diamond$ of a belief theory T one can revise T by adding to it the "possibility of F being false" for every F in the revision set $\mathcal{R}(T^\diamond)$. How can this be done?

Most belief revision systems take the position that if the belief in a given formula F leads to contradiction then its complement $\neg F$ should be assumed to be true. In our opinion this is, in general, unwarranted. First of all, it is not necessary to do so in order to inhibit the belief. Moreover, it is unwarranted to jump to a conclusion that some formula is true simply because belief in its falsity would lead to contradiction. That would be tantamount to imposing the law of the excluded middle on our beliefs, i.e. assuming the axiom $\mathcal{B}F \vee \mathcal{B}\neg F$.

In Example 4.1., we simply would like to prevent $\neg Broken$ from being believed. Given the meaning of beliefs, this can be arranged by changing the theory just enough so that "$Broken$ is no longer false in all minimal models" or, equivalently, by "guaranteeing the existence of a minimal model in which $Broken$ is true". Technically, this is achievable by adding to the theory the clause $Broken \vee Maybe_Not(Broken)$, where $Maybe_Not(Broken)$ is an atom not occurring elsewhere in the theory, and thus not constrained in value. This clause can be read as *"Broken is possible"*. Intuitively, this constitutes the "minimal" change

of the theory ensuring that contradiction is removed. Indeed, believing $\neg Broken$ leads to contradiction and therefore $Broken$ should be possible, which effectively and declaratively prevents believing in $\neg Broken$.

For the sake of modularity, instead of adding the clauses of the form $F \vee Maybe_Not(F)$, we prefer the addition of $Possible(F)$, where $Possible(F)$ is defined by:

$$Possible(F) \equiv F \vee Maybe_Not(F) \tag{7}$$

Definition 4.1. [Revision of a Belief Theory] A belief theory T_r is a *revision of a consistent belief theory* T if and only if

$$T_r = T \cup \{Possible(\neg F) : F \in \mathcal{R}(T^\diamond)\}$$

for some careful autoepistemic expansion $T^\diamond$ of T.

Theorem 4.1. [Revised Autoepistemic Expansion] *Let T_r be a revision of a consistent belief theory T. Then T_r is consistent and has a consistent least static autoepistemic expansion. The least static autoepistemic expansion of T_r is a* revised autoepistemic expansion *of T.*

Proof:
Let $T^\diamond$ be a careful autoepistemic expansion of the consistent theory T, and let $T_r = T \cup \{Possible(\neg F) : F \in \mathcal{R}(T^\diamond)\}$.

Since T is consistent, and T_r only differs from T on clauses of the form $F \vee Maybe_Not(F)$, where $Maybe_Not(F)$ is an atom not occurring elsewhere in T, it is easy to see that T_r is also consistent.

To prove that the least static expansion of T_r is also consistent, we begin by proving the following lemma:

Lemma 4.1. Let T' be a theory obtained by augmenting T_r with a set of belief formulae $\mathcal{B}G \in T^\diamond$. For every formula $F \in \mathcal{R}(T^\diamond)$, there is a minimal model M of T' such that $\neg F \in M$.

Proof:
Let $T' = T \cup \{Possible(\neg F) : F \in \mathcal{R}(T^\diamond)\} \cup B$, where B is a set of belief formulae contained in $T^\diamond$.

We start by proving that $T' \cup \{\neg F, \neg Maybe_Not(\neg F)\}$ is consistent, i.e. there exists a model of T' with $\neg F$ and $\neg Maybe_Not(\neg F)$. First note that $T^\diamond \cup \{\neg F\}$ is consistent: otherwise all models of $T^\diamond$ would contain F and, since $T^\diamond$ is closed under (N), $\mathcal{B}F$ would belong to $T^\diamond$, which is impossible because $F \in \mathcal{R}(T^\diamond)$. Thus $T \cup B \cup \{\neg F\}$ is consistent. Because in the remainder of T', $\neg F$ only occurs in $\neg F \vee Maybe_Not(\neg F)$, and $Maybe_Not(\neg F)$ does not occur elsewhere in T', $T' \cup \{\neg F, \neg Maybe_Not(\neg F)\}$ is also consistent.

Let N be a model of T' containing $\{\neg F, \neg Maybe_Not(\neg F)\}$. If N is minimal then, since $\neg F \in N$, the lemma is verified. Since $Maybe_Not(\neg F)$ is an atom not occurring elsewhere in T', if N is not minimal, there must exist a minimal model M of T' coinciding with N on belief atoms and containing $\neg Maybe_Not(\neg F)$. Since M must satisfy $\neg F \vee Maybe_Not(\neg F)$ it must also contain $\neg F$.

By Theorem 2.1., it is guaranteed that the sequence of the $\{T_r^\alpha\}$, constructed by successive applications of the belief closure operator Ψ, is monotonically increasing and has a unique fixed point. Thus, it is enough to prove that the obtained fixed point is consistent. Moreover, since $T^\diamond$ is consistent, it suffices to show that, for every T_r^α in the sequence, and any formula F not containing any occurrence of atoms of the form $Maybe_Not(G)$:

$$\mathcal{B}F \in T_r^\alpha \Rightarrow \mathcal{B}F \in T^\diamond$$

Suppose that β is an ordinal such that for all $\alpha < \beta$, if $\mathcal{B}F \in T_r^\alpha$ then $\mathcal{B}F \in T^\diamond$.

If β is a limit ordinal, since by definition of the sequence $T_r^\beta = \bigcup_{\alpha<\beta} T_r^\alpha$, then, by the compactness theorem for all $\mathcal{B}F \in T_r^\beta$, also $\mathcal{B}F \in T^\diamond$.

If $\beta = \alpha + 1$ is a successor ordinal, then $\mathcal{B}F \in T_r^\beta$ iff $T_r^\alpha \models_{\min} F$ or $\mathcal{B}F \in T_r^\alpha$. If $\mathcal{B}F \in T_r^\alpha$ then, by hypothesis, $\mathcal{B}F \in T^\diamond$. Otherwise, the proof proceeds by contradiction assuming that $\mathcal{B}F \notin T^\diamond$. In that case, since $T^\diamond$ is a careful expansion, either there is a minimal model of $T^\diamond$ with $\neg F$, or all of its minimal models have F but $T^\diamond \cup \{\mathcal{B}F\}$ is inconsistent. If the latter holds $F \in \mathcal{R}(T^\diamond)$, and so, by lemma 4.1., there is a minimal model of T_r^α with $\neg F$. Thus $T_r^\alpha \not\models_{\min} F$ – contradiction.

If there is a minimal model of $T^\diamond$ with $\neg F$ then, since $T^\diamond$ differs from T_r^α only by the addition of some belief atoms and clauses $Possible(G)$, it is clear that there must also exist a minimal model of T_r^α with $\neg F$ – contradiction.

¿From the above proof, the relation of revised expansions to careful autoepistemic expansions follows easily:

Corollary 4.1. [Relation to Careful Expansions] Let $T_r^\diamond$ be a revised autoepistemic expansion of a consistent belief theory T. There exists a careful expansion $T^\diamond$ of T such that, for any formula F not containing any occurrence of atoms of the form $Maybe_Not(G)$, $\mathcal{B}F \in T_r^\diamond \Rightarrow \mathcal{B}F \in T^\diamond$.

Intuitively, this means that revised expansions are more skeptical than careful expansions, in that the latter add more belief formulae than the former.

Example 4.2. The only revision of the theory T from Example 4.1. is given by $T_r = T \cup \{Possible(Broken)\}$. Accordingly, the only revised autoepistemic expansion of T is:

$$T_r^\diamond = Cn_*\left(T \cup \{Possible(Broken)\} \cup \{\mathcal{B}Car, \mathcal{B}\neg Runs\}\right)$$

It is easy to see that there are minimal models of the theory in which $Broken$ is true, and therefore, since Car is true in all models, those models include $FixIt$ too. Thus, neither $\mathcal{B}\neg Broken$ nor $\mathcal{B}\neg FixIt$ are added to the expansion.

Example 4.3. The revisions of theory T from Example 3.2. are

$$T_{r_1} = T \cup \{Possible(BadBattery)\} \text{ and } T_{r_2} = T \cup \{Possible(FlatTire)\}$$

Thus, the revised autoepistemic expansions are:

$$T_{r_1}^\diamond = Cn_*(T \cup \{Possible(BadBattery), \mathcal{B}Broken, \mathcal{B}\neg FlatTire\})$$
$$T_{r_2}^\diamond = Cn_*(T \cup \{Possible(FlatTire), \mathcal{B}Broken, \mathcal{B}\neg BadBattery\})$$

Each of them constitutes a diagnosis of a possible problem with the car.

4.2. Controlling the Level of Diagnosis

The belief in a formula may be conditional upon the belief in another formula. This is particularly true when diagnosing faults in a device: causally deeper component faults are sometimes preferred over less deep faults, that are simply consequences of the former. In such cases, one would like to control the level over which diagnosis is performed, by preventing diagnoses which do not focus on the causally deeper faults. We now show that revised autoepistemic expansion have sufficient expressive power to control the level of diagnosis.

Example 4.4. The theory T:

$$\neg Runs \qquad\qquad FlatTire \supset Broken$$
$$\mathcal{B}\neg Broken \supset Runs \qquad\qquad BadBattery \supset Broken$$

has a single revision: $T \cup \{Possible(Broken)\}$. The revised autoepistemic expansion contains both $\mathcal{B}\neg FlatTire$ and $\mathcal{B}\neg BadBattery$. This revision can be seen as a diagnosis of the car that just states the car might be broken.

However, in this case, one would like the diagnosis to delve deeper into the car problems, and obtain one diagnosis suggesting a possible problem with a flat tire and another suggesting a possible problem with a bad battery. This is justified by the fact that our belief in the car being broken seems to depend entirely on our belief that it either has a flat tire or a bad battery.

To obtain this more desirable result one has to somehow ensure that instead of just withholding our belief in the car not being broken we in fact also withhold our belief that the car neither has a flat tire nor a bad battery. In other words, a revision of this theory should not be initiated by revising $Broken$ but instead it should be initiated by revising $FlatTire$ or $BadBattery$ by adding either $Possible(FlatTire)$ or $Possible(BadBattery)$.

Note that, by the rule (N) and the axiom (K), the closure of T already contains:

$$\mathcal{B}FlatTire \vee \mathcal{B}BadBattery \supset \mathcal{B}Broken.$$

Thus, belief in the truth of $Broken$ is already determined by the belief in $FlatTire$ or in $BadBattery$. But we intend to express the stronger fact that belief in the falsity of $Broken$ must also be determined by the beliefs held about the latter literals. This is ensured by stating that if both $FlatTire$ and $BadBattery$ are believed false then $Broken$ must be also believed false:

$$\mathcal{B}\neg FlatTire \wedge \mathcal{B}\neg BadBattery \supset \mathcal{B}\neg Broken \tag{8}$$

Example 4.5. The theory T from Example 4.4., augmented with clause (8) now has two revised expansions:

$$T^{\diamond}_{r_1} = Cn_*(T \cup \{Possible(BadBattery), \mathcal{B}\neg Runs, \mathcal{B}\neg FlatTire\})$$
$$T^{\diamond}_{r_2} = Cn_*(T \cup \{Possible(FlatTire), \mathcal{B}\neg Runs, \mathcal{B}\neg BadBattery\})$$

each corresponding to one of the desired deeper diagnoses.
On the other hand, $T \cup \{Possible(Broken)\}$ is no longer a revision because it still derives $\mathcal{B}\neg Broken$, via clause (8), and thus is inconsistent.

Note the similarities between clause (8) and Clark's completion [4] of $Broken$. Clark's completion states that if both $FlatTire$ and $BadBattery$ are false then $Broken$ is false, whilst (8) refers instead to the corresponding beliefs. For this reason we call (8) the *belief completion clause* for $Broken$. More generally:

Definition 4.2. [Belief Completion Clauses] Let T be an AEB theory, and let:

$$B_{1,1} \wedge \ldots \wedge B_{1,m} \wedge \mathcal{B}\neg B_{1,m+1} \wedge \ldots \wedge \mathcal{B}\neg B_{1,n} \supset A$$
$$\ldots$$
$$B_{k,1} \wedge \ldots \wedge B_{k,m} \wedge \mathcal{B}\neg B_{k,m+1} \wedge \ldots \wedge \mathcal{B}\neg B_{k,n} \supset A$$

be all the clauses[3] for A in T, where A is an atom, each Bi,j is a literal, and $k > 0$. The *belief completion clauses for A in T*, $BelComp(A)$, are:

$$(\mathcal{B}\neg B_{1,1} \vee \ldots \vee \mathcal{B}\neg B_{1,m} \vee \mathcal{B}B_{1,m+1} \vee \ldots \vee \mathcal{B}B_{1,n})$$
$$\wedge \ldots \wedge (\mathcal{B}\neg B_{k,1} \vee \ldots \vee \mathcal{B}\neg B_{k,m} \vee \mathcal{B}B_{k,m+1} \vee \ldots \vee \mathcal{B}B_{k,n}) \supset \mathcal{B}\neg A$$

If there are no clauses for A in T then its belief completion is $\mathcal{B}\neg A$.

By adding the completion rules for an atom A, we can therefore prevent revision to be initiated in $\mathcal{B}\neg A$, i.e., in order to revise the belief in $\neg A$, beliefs in other literals on which A depends must also be revised. In diagnosis, the hierarchical component structure of artifacts naturally induces dependency levels into theories modeling them. In other words, we can impose, via belief completion clauses, the desired levels of diagnosis in artifacts.

[3] By a clause for an atom A we mean one in which A occurs positively.

4.3. Application to the debugging of logic programs

Here we illustrate the application of belief revision with completion clauses to declarative error diagnosis (or declarative debugging) of terminating normal logic programs, by first translating the programs to AEB theories. The restriction to terminating programs aims to simplify the exposition: all the major logic programming semantics coincide for such programs and there are no errors due to loops

Debugging of a logic program is required whenever the consequences of the program clash with the intended model of the user, and its goal is to detect the errors in the program. A debugger is declarative whenever the user needs only to know the intended model of an incorrect program to detect bugs. In particular, with a declarative debugger, the user does not need to know or be aware of the underlying operational behaviour of the program.

In terminating logic programs, errors manifest themselves only through two kinds of symptoms, or bug manifestations [19]:

- *wrong solution*, when some ground atom is an undesirable consequence of the program, i.e. a consequence which is not part of the intended user model.
- *missing solution*, when some ground atom belongs to the intended user model, but is not a consequence of the program.

Of course, whenever there is a missing or a wrong solution manifestation then the program is not correct with respect to its intended model, and so there must necessarily exist in it some bug requiring correction. In [19], two kinds of errors are identified: *uncovered atoms* and *incorrect clause instances*. An atom A is uncovered if it belongs to the intended model but there are no rules in the program with head A and true body. A clause instance is incorrect if the head of the clause instance does not belong to the intended model but its body is true in the intended model.

Example 4.6. Consider the logic program P:

$$a \leftarrow not\, b$$
$$b \leftarrow not\, c$$

whose consequences are $\{not\, a, b, not\, c\}$, and the indented user model $\{not\, a, not\, b, c\}$ that clashes with it.

In this case, the symptoms are that b is a wrong solution and c is a missing solution. The reader can easily check that the errors in this program that explain the clash are that c is uncovered, and that the first clause is incorrect.

In order to use belief revision in AEB to perform declarative debugging of normal logic programs, the first step to take is to translate the programs into belief theories via the translation $T_{\mathcal{B}\neg}(P)$ which defines their semantics in AEB. Moreover, in order to allow for the existence of incorrect clause instances, clause instances must be conditional upon the assumption of their correctness. To further allow for the possibility of uncovered atoms, for each atom a clause stating that it is true whenever the atom is uncovered, is required. This yields the following translation:

Definition 4.3. [Debugging Translation] Let P be a normal logic program consisting of rule instances:

$$A \leftarrow B_1 \wedge \ldots \wedge B_m \wedge not\, C_1 \wedge \ldots \wedge not\, C_n$$

The translation $T_{debug}(P)$ is given by the set of the corresponding clause instances:

$$\mathcal{B}\neg incorrect(r_n) \supset (B_1 \wedge \ldots \wedge B_m \wedge \mathcal{B}\neg C_1 \wedge \ldots \wedge \mathcal{B}\neg C_n \supset A)$$

or, equivalently:

$$\mathcal{B}\neg incorrect(r_n) \wedge B_1 \wedge \ldots \wedge B_m \wedge \mathcal{B}\neg C_1 \wedge \ldots \wedge \mathcal{B}\neg C_n \supset A$$

where r_n is a unique name assigned to each clause instance n, plus a clause

$$uncovered(A) \supset A$$

for each atom A in the language of P. All atoms of the form $incorrect(_)$ or $uncovered(_)$ are new, not occurring elsewhere in the theory.

The following (easy to check) proposition shows that the differences between $T_{\mathcal{B}\neg}(P)$ and $T_{debug}(P)$ do not affect the semantics of the resulting theory. In fact their static expansions coincide modulo the new predicates introduced, i.e. coincide on all formulae common to the languages of both theories.

Proposition 4.1. Let P be a normal logic program, $T = T_{\mathcal{B}\neg}(P)$ and $T_d = T_{debug}(P)$. For each static expansion $T_d^\diamond$ of T_d there exists one static expansion $T^\diamond$ of T such that, for every formula F with no occurrences of $incorrect(_)$ or $uncovered(_)$, $T_d^\diamond \models F$ if and only if $T^\diamond \models F$, and reciprocally.

In fact, since there are no positive occurrences of atoms $incorrect(r_n)$ in $T_{debug}(P)$, $\neg incorrect(r_n)$ is true in all minimal models, and so $\mathcal{B}\neg incorrect(r_n)$ must necessarily belong to all static expansions. Thus the addition of $\mathcal{B}\neg incorrect(r_n)$ to clauses does not affect the expansion. Similarly for the addition of the clauses with the body $uncovered(A)$.

Example 4.7. The translation $T_{debug}(P)$ of the program in Example 4.6. is:

$$\begin{array}{ll} \mathcal{B}\neg incorrect(r_1) \supset (\mathcal{B}\neg b \supset a) & uncovered(a) \supset a \\ \mathcal{B}\neg incorrect(r_2) \supset (\mathcal{B}\neg c \supset b) & uncovered(b) \supset b \\ & uncovered(c) \supset c \end{array}$$

Missing and wrong solution declarations can easily be expressed in AEB:

- Stating that A is a missing solution of a program P simply means that, although A is not a consequence of the program, the user believes in A. So, just add $\mathcal{B}A$ to $T_{debug}(P)$.
- Stating that A is a wrong solution of a program P, means that, although A is a consequence of the program, the user believes A is false. So, just add $\mathcal{B}\neg A$ to $T_{debug}(P)$.

Example 4.8. To state that c is a missing solution of the program in Example 4.6. simply add $\mathcal{B}c$ to $T_{debug}(P)$. Note that the resulting theory has no static expansions. Indeed, since it does not offer any evidence for c, it yields $\mathcal{B}\neg c$, resulting in a contradiction with $\mathcal{B}c$.

In order to obtain the errors of the program, belief revision over the resulting theory is required. The revisions of the theory identify the errors of the program. However, not all beliefs should be considered for revision: only those for $incorrect(_)$ or $uncovered(_)$. This effect can be achieved by adding, for all the other atoms, their corresponding belief completion clauses.

Example 4.9. The theory:

$$\begin{array}{r} \mathcal{B}\neg uncovered(a) \wedge \mathcal{B}incorrect(r_1) \supset \mathcal{B}\neg a \\ \mathcal{B}\neg uncovered(b) \wedge \mathcal{B}incorrect(r_2) \supset \mathcal{B}\neg b \\ \mathcal{B}\neg uncovered(c) \supset \mathcal{B}\neg c \end{array}$$

$$\begin{array}{r} \mathcal{B}\neg uncovered(a) \wedge \mathcal{B}b \supset \mathcal{B}\neg a \\ \mathcal{B}\neg uncovered(b) \wedge \mathcal{B}c \supset \mathcal{B}\neg b \end{array}$$

resulting from $T_{debug}(P)$ plus $\mathcal{B}c$ and the belief completion clauses, has a single revision, namely the one resulting from adding $Possible(uncovered(c))$ to the theory, stating that possibly atom c is uncovered. In fact this corresponds to the only error in the program.

Note that for debugging a logic program by using revision of AEB theories in this manner there is no need for a complete description of the intended model. A description of the detected symptoms is enough. This was the case in the example above, where nothing was stated about either a or b. If extra symptoms are detected, they can be incrementally added to the theory in order to find additional errors of the program:

Example 4.10. If wrong solutions for both a and b become manifest, then we add to the theory $\{\mathcal{B}\neg a, \mathcal{B}\neg b\}$. The revision of the resulting theory is obtained by the addition of $\{Possible(uncovered(a)), Possible(incorrect(r_1))\}$.

Note that $T \cup \{Possible(incorrect(r_1))\}$ is no longer a revision. Indeed, since there is no evidence for $uncovered(b)$, all static expansion must have $\mathcal{B}\neg uncovered(b)$. Since $\mathcal{B}c$ belongs to the theory, belief completion on b implies $\mathcal{B}\neg b$ (cf. the clauses shown in Example 4.9.). ¿From $\mathcal{B}\neg b$ and the fact that there is no evidence for $incorrect(r_1)$ (i.e. the theory yields $\mathcal{B}\neg incorrect(r_1)$) a follows, resulting, by virtue of the Consistency Axiom (D) and necessitation (N), in a contradiction with $\mathcal{B}\neg a$.

In the debugging of logic programs it might be useful to dismiss certain errors from the start. This is an easy matter in AEB: it simply requires the addition of facts with the negation of the corresponding error atoms:

Example 4.11. Consider the buggy program containing the single rule:

$$a \leftarrow not\ b$$

where a is a wrong solution. Moreover, the user wants to dismiss from the start the possibility of b being uncovered.

Finding the errors of the program can be done by revising the theory resulting from $T_{debug}(P)$ plus the belief completion clauses for a and b, and the facts $\mathcal{B}\neg a$ and $\neg uncovered(b)$, stating respectively that a is a wrong solution and that b is not uncovered:

$$\begin{array}{rl} \mathcal{B}\neg incorrect(r_1) \wedge \mathcal{B}\neg b \supset a & \qquad \mathcal{B}\neg a \\ uncovered(a) \supset a & \qquad \neg uncovered(b) \\ uncovered(b) \supset b & \end{array}$$

$$\begin{array}{rl} \mathcal{B}\neg uncovered(a) \wedge \mathcal{B}incorrect(r_1) \supset \mathcal{B}\neg a & \qquad \mathcal{B}\neg uncovered(b) \supset \mathcal{B}\neg b \\ \mathcal{B}\neg uncovered(a) \wedge \mathcal{B}b \supset \mathcal{B}\neg a & \end{array}$$

The only revision of T is $T \cup \{Possible(incorrect(r_1))\}$.

Note that if $\neg uncovered(b)$ were not added, the theory would have two revisions, $T \cup \{Possible(uncovered(b))\}$ and $T \cup \{Possible(incorrect(r_1))\}$, meaning that either some clause for b is missing or that rule r_1 is incorrect.

5. Concluding Remarks

We have argued that common-sense reasoning requires that general non-monotonic reasoning formalisms pay due attention to the issue of revising sets of assumptions that lead to contradiction.

We then went on to show how controlled revision of assumed beliefs can be naturally formalized within the broad and flexible framework of the auto epistemic logic of beliefs AEB. This logic encompasses other major general formalisms for non monotonic reasoning, for which such belief revision mechanisms have not yet been defined.

Subsequently, we exemplified the usefulness of our belief revision approach by applying it to the practical domains of model based diagnosis and debugging of normal logic programs, showing how one can resolve, in a natural and declarative way and without using meta linguistic devices, the issue of selective revision of beliefs.

For future work, we leave the application of AEB to the debugging of more general logic programs, and the debugging of AEB theories themselves.

References

[1] J. J. Alferes and L. M. Pereira. On logic program semantics with two kinds of negation. In K. Apt, editor, *International Joint Conference and Symposium on Logic Programming*, pages 574–588. MIT Press, 1992.

[2] J. J. Alferes and L. M. Pereira. Belief, provability and logic programs. In D. Pearce and L. M. Pereira, editors, *International Workshop on Logics in Artificial Intelligence, JELIA'94*, volume 838 of *Lecture Notes in Artificial Intelligence*, pages 106–121. Springer–Verlag, 1994.

[3] J. J. Alferes, L. M. Pereira, and T. C. Przymusinski. "Classical" negation in non monotonic reasoning and logic programming. In H. Kautz and B. Selman, editors, *4th Int. Symposium on Artificial Intelligence and Mathematics*. Florida Atlantic University, 1996.

[4] K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, 1978.

[5] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[6] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In *Proceedings of the Seventh International Logic Programming Conference, Jerusalem, Israel*, pages 579–597, Cambridge, Mass., 1990. Association for Logic Programming, MIT Press.

[7] Michael Gelfond, Halina Przymusinski, and Teodor C. Przymusinski. On the relationship between circumscription and negation as failure. *Journal of Artificial Intelligence*, 38(1):75–94, February 1989.

[8] W. Marek and M. Truszczynski. *Non-Monotonic Logic*. Springer Verlag, 1994.

[9] J. McCarthy. Circumscription – a form of non-monotonic reasoning. *Journal of Artificial Intelligence*, 13:27–39, 1980.

[10] J. Minker. On indefinite data bases and the closed world assumption. In *Proc. 6-th Conference on Automated Deduction*, pages 292–308, New York, 1982. Springer Verlag.

[11] L. M. Pereira and J. J. Alferes. Well founded semantics for logic programs with explicit negation. In B. Neumann, editor, *European Conference on Artificial Intelligence*, pages 102–106. John Wiley & Sons, 1992.

[12] L. M. Pereira and J. J. Alferes. Contradiction: when avoidance equal removal. Part II. In R. Dyckhoff, editor, *Extensions of Logic Programming*, number 798 in LNAI, pages 268–281. Springer–Verlag, 1994.

[13] L. M. Pereira, J. J. Alferes, and J. N. Aparício. Contradiction Removal within Well Founded Semantics. In A. Nerode, W. Marek, and V. S. Subrahmanian, editors, *Logic Programming and NonMonotonic Reasoning*, pages 105–119. MIT Press, 1991.

[14] T. C. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–464, 1990.

[15] T. C. Przymusinski. A knowledge representation framework based on autoepistemic logic of minimal beliefs. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94, Seattle, Washington, August 1994*, pages 952–959, Los Altos, CA, 1994. American Association for Artificial Intelligence, Morgan Kaufmann.

[16] T. C. Przymusinski. Static semantics for normal and disjunctive logic programs. *Annals of Mathematics and Artificial Intelligence*, Special Issue on Disjunctive Programs, 1994.

[17] T. C. Przymusinski. Autoepistemic logic of knowledge and beliefs. (In preparation), University of California at Riverside, 1995. (Extended abstract appeared in 'A knowledge representation framework based on autoepistemic logic of minimal beliefs' In *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94, Seattle, Washington, August 1994*, pages 952-959, Los Altos, CA, 1994. American Association for Artificial Intelligence, Morgan Kaufmann.).

[18] R. Reiter. On closed-world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 55–76. Plenum Press, New York, 1978.

[19] E. Y. Shapiro. *Algorithmic Program Debugging*. MIT Press, 1983.