Belnap's Epistemic States and Negation-as-Failure*

Gerd Wagner
Institut für Informatik, Universität Leipzig †

Abstract

Generalizing Belnap's system of *epistemic states* [Bel77] we obtain the system of *disjunctive factbases* which is the paradigm for all other kinds of disjunctive knowledge bases. Disjunctive factbases capture the nonmonotonic reasoning based on paraminimal models. In the schema of a disjunctive factbase, certain predicates of the resp. domain are declared to be *exact*, i.e. two-valued, and in turn some of these exact predicates are declared to be subject to the Closed-World Assumption (CWA). Thus, we distinguish between three kinds of predicates: inexact predicates, exact predicates subject to the CWA, and exact predicates not subject to the CWA.

Contents

*To appear in H. Wansing (Ed.), *Negation – A Notion in Focus*, de Gruyter, 1996.
†E-Mail: `gw@inf.fu-berlin.de`

1 Introduction

In this paper, we shall address issues of disjunctive information processing, resp. disjunctive knowledge representation and reasoning, such as the interplay between disjunctive and negative information (concerning explicit negative information, negation-as-failure and the Closed-World Assumption), and the notion of inclusive versus exclusive disjunctive information. These issues have first been discussed in the field of disjunctive logic programming (see, e.g. [Min82, RT88, Sak89, GL91, LMR92]). However, there is no generally acknowledged semantics for disjunctive logic programs, and many proposals are not based on clear logical principles but seem to be rather ad-hoc. It appears that the non-rule-related issues of disjunctive information processing are complicated by the presence of deduction rules, and a semantical account for the interplay between disjunctive, negative and deductive knowledge is hard to find. Therefore, we believe it is methodologically preferable to settle these issues in a simpler framework without rules, and this is the approach taken here.

The concept of a knowledge representation and reasoning system, or shorter: *knowledge system (KS)*, consists essentially of two main components: an inference and an update operation manipulating knowledge bases as abstract objects,[1] together with a set of formal properties these operations may have. In general, there are no specific restrictions on the internal structure of a knowledge base. It appears, however, that a computational design can be achieved by 'compiling' incoming information into some normal form rather than leaving it in the form of arbitrarily complex formulas. This is the case, for instance, in Belnap's KS which can be considered as a paradigm for knowledge systems.

The concept of a KS constitutes a useful framework for the classification and comparison of various computational systems and formalisms like, e.g., relational and deductive databases, logic programs and other rule-based systems. It is more general than that of a logic (i.e. a consequence relation). A standard logic can be viewed as a special kind of KS. On the other hand, by defining the inference and update operations procedurally, KSs can serve as the basis for the operational definition of logics. For instance, by appropriate settings of certain 'parameters' in the system of disjunctive factbases one can obtain three-valued and classical logic, in addition to Belnap's four-valued logic.

In knowledge representation, two different notions of falsity arise in a natural way. Certain facts are *implicitly false by default* by being not verified in any intended model of the knowledge base. Others are *explicitly false* by virtue of a direct proof of their falsity, corresponding to their falsification in all intended models. These two kinds of falsity in knowledge representation are captured by the two negations, called *weak* and *strong*, of partial logic. In the monotonic

[1] This distinction was already proposed in [Lev84] where the resp. operations were called *ASK* and *TELL*.

base system of partial logic, weak negation corresponds to classical negation by virtue of a straightforward translation of partial logic into classical logic which is discussed in [HJW96]. In the nonmonotonic refinements of partial logic based on (para-)minimal and stable reasoning, weak negation corresponds to negation-as-failure, and hence can be used to express local Closed-World Assumptions, default rules, and the like.

Both relational and deductive database systems can be considered as computational paradigms of real world knowledge systems. They implement a form of nonmonotonic reasoning caused by the use of negation-as-failure refering to default-implicit negative information. On the other hand, relational and deductive databases, as well as normal logic programs, are not capable of representing and processing explicit negative information. This shortcoming has led to the extension of logic programming by adding a second negation (in addition to negation-as-failure) as proposed independently in [GL90, GL91] and in [PW90, Wag91]. We call the general concept of an operator expressing default-implicit negative information in the style of negation-as-failure *weak negation*, and denote it by '$-$', while the concept of an operator expressing explicit negative information will be called *strong negation*, denoted by '$\sim$'. Our concept of a *vivid knowledge system (VKS)* is a two-fold generalization:

1. it extends already known logics, such as Belnap's 4-valued or Nelson's constructive logic,[2] by adding *weak* negation, and

2. it extends already known knowledge systems, such as relational or deductive database systems, by adding *strong* negation.

In the framework of a VKS, a specific meaning is assigned to the *Closed-World Assumption*: if the Closed-World Assumption holds for a predicate, its weak negation implies its strong negation, in other words, an atomic sentence formed with such a predicate is already false if it is false by default.

In real world knowledge bases like, for instance, relational or deductive databases, it is essential to be able to infer negative information by means of *minimal* (resp. *stable*) reasoning, i.e. drawing inferences on the basis of minimal (resp. stable) models. Relational databases, being finite sets of tables the rows of which represent atomic sentences, have traditionally been viewed as finite models. On this account, answering a query F is rather based on the model relation, $\mathcal{M}_\Delta \models F$, where $\mathcal{M}_\Delta$ is the finite interpretation corresponding to the database Δ, and not on an inference relation. However, especially with respect to the generalization of relational databases (e.g. in order to allow for incomplete information), it seems to be more adequate to regard a relational database as a set of atomic sentences A_Δ, and to infer a query F whenever it holds in the unique minimal model of A_Δ, i.e.

$$A_\Delta \vdash F :\Leftrightarrow \text{Min}(\text{Mod}(A_\Delta)) \subseteq \text{Mod}(F) \Leftrightarrow \mathcal{M}_\Delta \models F$$

While minimal models are adequate for definite extensional knowledge bases (such as factbases), a refinement of the notion of minimality, called *paraminimality*, is needed to capture the inclusiveness of disjunctive knowledge.

[2] See [Bel77], resp. [AN84].

1.1 Preliminaries

A *signature* $\sigma = \langle Rel, ExRel, Const\rangle$ consists of a set of relation symbols Rel, a set $ExRel \subseteq Rel$ of exact relation symbols, and a set of constant symbols $Const$.[3]

We consider the following logical functors: conjunction ($\wedge$), disjunction ($\vee$), strong negation ($\sim$), weak negation (alias negation-as-failure, denoted by $-$), exclusive disjunction ($|$), and the truth constant 1; relation symbols are denoted by $p, q, r, \ldots$; constant symbols by $c, d, \ldots$; and variables by $x, y, \ldots$. Quantifiers, $\exists$ and $\forall$, are only incidentally considered. If $\mathcal{F}$ is a set of logical functors, $L(\sigma; \mathcal{F})$ denotes the corresponding set of wellformed formulas. $L(\sigma) = L(\sigma; -, \sim, \wedge, |, \vee)$ is the smallest set containing the atomic formulas of σ, and being closed with respect to the following condition: if $F, G \in L(\sigma)$, then $\{-F, \sim F, F \wedge G, F \vee G, F|G\} \subseteq L(\sigma)$.

With respect to a signature σ we define the following sublanguages: $\mathrm{At}(\sigma) = L(\sigma; \emptyset)$, the set of all atomic sentences (also called *atoms*); $\mathrm{Lit}(\sigma) = L(\sigma; \sim)$, the set of all *literals*; and $\mathrm{XLit}(\sigma) = \mathrm{Lit}(\sigma) \cup \{-l : l \in \mathrm{Lit}(\sigma)\}$, the set of all *extended literals*. We shall frequently omit the reference to a specific signature, and simply write L instead of $L(\sigma)$. We introduce the following convention: when L is a set of sentences, L^x denotes the corresponding set of formulas.

An *atom* $a \in \mathrm{At}$ is called *proper*, if $a \neq 1$. We use $a, b, \ldots, l, k, \ldots, e, f, \ldots,$ and $F, G, H, \ldots$ as metavariables for atoms, literals, extended literals and well-formed formulas, respectively.

With each negation a complement operation for the resp. type of literal is associated: $\tilde{a} = \sim a$ and $\widetilde{\sim a} = a$, $\bar{l} = -l$ and $\overline{-l} = l$. These complements are also defined for sets of resp. literals $L \subseteq \mathrm{Lit}$, and $E \subseteq \mathrm{XLit}$: $\widetilde{L} = \{\tilde{l} : l \in L\}$, resp. $\overline{E} = \{\bar{e} : e \in E\}$. We distinguish between the positive and negative elements of $E \subseteq \mathrm{XLit}$ by writing $E^+ := E \cap \mathrm{Lit}$ and $E^- := \{l : -l \in E\}$.

If X is a set of sets, then $\mathrm{Fin}(X)$ denotes its restriction to finite elements. If Y is an ordered set, then $\mathrm{Min}(Y)$ denotes the set of all minimal elements of Y, i.e. $\mathrm{Min}(Y) = \{X \in Y \mid \neg\exists X' \in Y : X' < X\}$

Let $L \subseteq L(\sigma)$ be a nonempty language. An operation $C : 2^L \to 2^L$ is called an *inference operation*. The corresponding *inference relation* $\vdash$ is defined by $X \vdash F$ iff $F \in C(X)$. An inference operation (relation) is called a *consequence* operation (relation) if it satisfies Inclusion (Reflexivity), Idempotence (Transitivity), and Monotony.

A *model-theoretic system* $\langle L, \boldsymbol{I}, \models\rangle$ is determined by a language L, a set $\boldsymbol{I}$ whose elements are called *interpretations* and a *model relation* $\models \subseteq \boldsymbol{I} \times L$ between interpretations and formulas. With every model-theoretic system $\langle L, \boldsymbol{I}, \models\rangle$, we can associate a model operator $\mathrm{Mod}_{\mathbf{I}}$, a consequence operation $C_{\mathbf{I}}$, and a consequence relation $\models_{\mathbf{I}}$ in the following way. Let $X \subseteq L$, then the associated model operator is defined as $\mathrm{Mod}_{\mathbf{I}}(X) = \{\mathcal{I} \in \boldsymbol{I} : \mathcal{I} \models X\}$, where $\mathcal{I} \models X$ iff for every $F \in X : \mathcal{I} \models F$. The associated consequence operation is defined by $C_{\mathbf{I}}(X) = \{F \in L : \mathrm{Mod}_I(X) \subseteq \mathrm{Mod}_I(F)\}$, and finally $X \models_{\mathbf{I}} F$ iff $F \in C_{\mathbf{I}}(X)$.

An inference operation C is called *correct*, resp. *complete*, with respect to the model-theoretic system $\langle L, \boldsymbol{I}, \models\rangle$ iff $C(X) \subseteq C_{\mathbf{I}}(X)$, resp. $C(X) = C_{\mathbf{I}}(X)$.

1.2 Partial Logics with Two Kinds of Negation

Definition 1 (Interpretation) *Let $\sigma = \langle Rel, ExRel, Const\rangle$ be a signature. A partial Herbrand σ-interpretation $\mathcal{I}$ consists of:*

[3] For the sake of simplicity we shall not consider functional terms but only variables and constants; therefore, we define signatures without function symbols leading to a finite Herbrand universe.

1. *A set U, the* universe *or* domain *of $\mathcal{I}$ which is equal to the set of constant symbols, $U = Const$;*

2. *an assignment $c^{\mathcal{I}} = c$ to every constant symbol $c \in Const$;*

3. *an assignment of a pair $\langle r^{\mathcal{I}}, \tilde{r}^{\mathcal{I}} \rangle$ to every relation symbol $r \in Rel$ such that*

$$r^{\mathcal{I}} \cup \tilde{r}^{\mathcal{I}} \subseteq U^{a(r)},$$

and in the special case of an exact relation symbol $r \in ExRel$,

$$r^{\mathcal{I}} \cup \tilde{r}^{\mathcal{I}} = U^{a(r)},$$

where $a(r)$ denotes the arity of r.

In the sequel we shall also simply say 'interpretation' instead of 'partial Herbrand interpretation'.

The class of all partial Herbrand σ-interpretations is denoted by $\boldsymbol{I}_4(\sigma)$. We define the classes of *coherent*, of *total*, and of total coherent (or *2-valued*) interpretations by

$$\begin{aligned} \boldsymbol{I}_c(\sigma) &= \{\mathcal{I} \in \boldsymbol{I}_4(\sigma) : r^{\mathcal{I}} \cap \tilde{r}^{\mathcal{I}} = \emptyset \text{ for all } r \in Rel\} \\ \boldsymbol{I}_t(\sigma) &= \{\mathcal{I} \in \boldsymbol{I}_4(\sigma) : r^{\mathcal{I}} \cup \tilde{r}^{\mathcal{I}} = U^{a(r)} \text{ for all } r \in Rel\} \\ \boldsymbol{I}_2(\sigma) &= \boldsymbol{I}_c(\sigma) \cap \boldsymbol{I}_t(\sigma) \end{aligned}$$

The model relation $\models \subseteq \boldsymbol{I}_4(\sigma) \times L(\sigma;\, -, \sim, \wedge, \vee, |, \exists, \forall)$ between an interpretation and a sentence is defined inductively as follows.

Definition 2 (Model Relation) *1. $\mathcal{I} \models r(c_1, \ldots, c_m)$ iff $\langle c_1, \ldots, c_m \rangle \in r^{\mathcal{I}}$.*
$\mathcal{I} \models \sim r(c_1, \ldots, c_m)$ iff $\langle c_1, \ldots, c_m \rangle \in \tilde{r}^{\mathcal{I}}$.

2. *$\mathcal{I} \models F \wedge G$ iff $\mathcal{I} \models F$ and $\mathcal{I} \models G$.*

3. *$\mathcal{I} \models F \vee G$ iff $\mathcal{I} \models F$ or $\mathcal{I} \models G$.*

4. *$\mathcal{I} \models \exists x F(x)$ iff $\mathcal{I} \models F(c)$ for some $c \in Const$.*

5. *$\mathcal{I} \models \forall x F(x)$ iff $\mathcal{I} \models F(c)$ for all $c \in Const$.*

6. *$\mathcal{I} \models -F$ iff $\mathcal{I} \not\models F$.*

All other cases of compound formulas are handled by the following DeMorgan-style rewrite rules:

$$\begin{array}{rclrcl} \sim (F \wedge G) & \longrightarrow & \sim F \vee \sim G & \qquad \sim (F \vee G) & \longrightarrow & \sim F \wedge \sim G \\ \sim \exists x F(x) & \longrightarrow & \forall x \sim F(x) & \sim \forall x F(x) & \longrightarrow & \exists x \sim F(x) \\ \sim\sim F & \longrightarrow & F & \sim -F & \longrightarrow & F \end{array}$$

and the definition for exclusive disjunction:

$$F|G \longrightarrow (F \wedge -G) \vee (G \wedge -F)$$

in the sense that for every rewrite rule $LHS \longrightarrow RHS$, we define $\mathcal{I} \models LHS$ iff $\mathcal{I} \models RHS$.

Mod_* denotes the model operator associated with the system $\langle L(\sigma), \boldsymbol{I}_*, \models \rangle$, and $\models_*$ denotes the resp. consequence relation, for $* = 4, c, t, 2$, i.e.

$$X \models_* F \quad \text{iff} \quad \mathrm{Mod}_*(X) \subseteq \mathrm{Mod}_*(F)$$

Definition 3 (Diagram) *The* diagram *of* $\mathcal{I} \in \boldsymbol{I}_4(\sigma)$ *is defined as* $D_{\mathcal{I}} = \{l \in \mathrm{Lit}(\sigma) : \mathcal{I} \models l\}$.

Observation 1 *Partial Herbrand interpretations can be identified with their diagrams, i.e. there is a one-to-one correspondence between* $\boldsymbol{I}_4(\sigma)$ *and* $2^{\mathrm{Lit}(\sigma)}$.

Definition 4 (Extension) *Let* $\mathcal{I}$ *and* $\mathcal{I}'$ *be two interpretations. We say that* $\mathcal{I}'$ extends $\mathcal{I}$, *symbolically* $\mathcal{I} \leq \mathcal{I}'$, *if* $D_{\mathcal{I}} \subseteq D_{\mathcal{I}'}$.

Definition 5 (Minimal Models) *For* $F \in L(\sigma) \supseteq X$, *and* $* = 4, c$, *we define* $\mathrm{Mod}_*^m(X) = \mathrm{Min}(\mathrm{Mod}_*(X))$, *and* $X \models_*^m F$ *iff* $\mathrm{Mod}_*^m(X) \subseteq \mathrm{Mod}_*(F)$.

Observation 2 *An interpretation is the unique minimal model of its diagram: for* $* = 4, c$, *and for every* $\mathcal{I} \in \boldsymbol{I}_*(\sigma)$, $\mathrm{Mod}_*^m(D_{\mathcal{I}}) = \{\mathcal{I}\}$, *and consequently,* $\mathcal{I} \models F$ *iff* $D_{\mathcal{I}} \models_*^m F$.

Formulas of partial logic (with two kinds of negation) can be normalized in the same manner as those of classical logic. For this purpose, we introduce $\mathrm{DNS}(F)$, the *disjunctive normal set* of a formula F, which is defined as follows:

$$
\begin{array}{lcl}
\mathrm{DNS}(1) & = & \{\emptyset\} \\
\mathrm{DNS}(e) & = & \{\{e\}\} \\
\mathrm{DNS}(F \wedge G) & = & \{E \cup D \ : \ E \in \mathrm{DNS}(F),\ D \in \mathrm{DNS}(G)\} \\
\mathrm{DNS}(F \vee G) & = & \mathrm{DNS}(F) \cup \mathrm{DNS}(G)
\end{array}
$$

All other cases of compound formulas can be handled by the above and the following DeMorgan-style rewrite rules:

$$
\begin{array}{rcl@{\qquad}rcl}
-(F \vee G) & \longrightarrow & -F \wedge -G & -(F \wedge G) & \longrightarrow & -F \vee -G \\
-\!\sim\!(F \vee G) & \longrightarrow & -\!\sim\! F \vee -\!\sim\! G & -\!\sim\!(F \wedge G) & \longrightarrow & -\!\sim\! F \wedge -\!\sim\! G \\
-\!\sim\sim\! F & \longrightarrow & -F & --F & \longrightarrow & F \\
-\!\sim\! -F & \longrightarrow & -F & & &
\end{array}
$$

The disjunctive normal form of a formula $G \in L(-, \sim, \wedge, \vee)$ is obtained as

$$
\mathrm{DNF}(G) = \bigvee_{E \in \mathrm{DNS}(G)} \bigwedge E
$$

Observation 3 *Let* $F \in L(\sim, \wedge, \vee)$. *Then,*

1. $\mathrm{Mod}_4^m(F) \subseteq \mathrm{DNS}(F)$.
2. $\mathrm{DNS}(F) \subseteq \mathrm{Mod}_4(F)$.

2 Knowledge Systems

Before presenting the formal definitions, we start with a semi-formal discusssion of the basic concepts to be introduced, notably: knowledge base, query, inference, answer, information ordering, input and update.

In general, a knowledge base (KB) can consist of any kind of data structures capable of representing knowledge, e.g. a set, or multiset, or sequence, of (logical) expressions, or a directed graph, etc. For the sake of simplicity, we shall assume that a KB is a set of expressions from a representation language. Only certain formulas may make sense for representing knowledge, that is, there will be a specific representation language L_{Repr}, and a KB will be a (usually finite) collection of elements of L_{Repr}, possibly constrained in some way determined by the set L_{KB} of all admissible KBs: $\mathrm{KB} \in L_{\mathrm{KB}} \subseteq 2^{L_{\mathrm{Repr}}}$. Likewise, since not every formula may be appropriate as a sensible query, the set of admissible queries is specified by L_{Query}.

The basic scenario of a knowledge system (KS) consists of two operations: an inference operation processing queries posed to the KB, and an update operation processing inputs entered by users or by other (e.g. sensoric) information suppliers. A KS restricts the admissible inputs to elements of a specific input language L_{Input}, and an update is performed by processing the input formula in an appropriate way in order to assimilate its information content into the KB. Since it appears reasonable to require that any information entered to a KB can be queried afterwards, we shall assume that $L_{\mathrm{Input}} \subseteq L_{\mathrm{Query}}$.

Definition 6 (Knowledge System) *An abstract knowledge system* $\boldsymbol{K}$ *is a quintuple:*[4]

$$\boldsymbol{K} = \langle L_{\mathrm{KB}}, \vdash, L_{\mathrm{Query}}, \mathsf{Upd}, L_{\mathrm{Input}} \rangle$$

where the inference relation $\vdash \subseteq L_{\mathrm{KB}} \times L_{\mathrm{Query}}$*, together with the update operation* $\mathsf{Upd} : L_{\mathrm{KB}} \times L_{\mathrm{Input}} \to L_{\mathrm{KB}}$*, satisfy for any* $X \in L_{\mathrm{KB}}$*,*

(KS1) $X \vdash 1$, *and* $\mathsf{Upd}(X, 1) = X$.

(KS2) $L_{\mathrm{Input}} \subseteq L_{\mathrm{Query}}$.

(KS3) $\mathsf{Upd}(X, F) \vdash F$, *for any* $F \in L_{\mathrm{Input}}$ *which is consistent with* X.[5]

If elements of L_{KB} are finite sets (resp. structures), $\boldsymbol{K}$ is called *finitary.* In the sequel, we shall sometimes simply write 'KB' in formal expressions standing for an arbitrary knowledge base $X \in L_{\mathrm{KB}}$. An inference operation C is defined as usual:

$$C(\mathrm{KB}) = \{F \in L_{\mathrm{Query}} : \mathrm{KB} \vdash F\}$$

Also, an answer operation taking a knowledge base and an open query formula, and providing the corresponding set of valid answers, can be defined in terms of the inference relation.

Definition 7 (Answer Operation) *If KB is* definite,[6] *an answer to an open query formula* $F(x)$ *is a tuple, and the set of all answers is a relation:*

$$\mathsf{Ans}(\mathrm{KB}, F(x)) = \{t : \mathrm{KB} \vdash F(t)\}$$

[4]The formulation of a KS in terms of query and input processing was already implicitly present in Belnap's [1977] view of a KS. In [Lev84] it was proposed as a 'functional approach to knowledge representation'. In [Wag94] the concept of knowledge systems was further extended and used as an integrating framework for knowledge representation and logic programming.

[5]According to some notion of consistency associated with the abstract knowledge system. E.g., one might want to exclude contradictory pieces of information from this reflexivity principle: $\mathsf{Upd}(\{\sim p\}, p) \not\vdash p$. We shall not discuss this issue in the present paper, however.

[6]KB is called *definite* if $\mathrm{KB} \vdash F \vee G$ implies that either $\mathrm{KB} \vdash F$, or $\mathrm{KB} \vdash G$, where $\vdash$ is a constructive inference relation (see below).

where x is a variable (tuple) and t a constant (tuple). In general, however, an answer may be indefinite, i.e. a minimal set of possible answer substitutions corresponding to a minimal disjunctive consequence:

$$\mathsf{Ans}(\mathrm{KB}, F(x)) = \mathrm{Min}(\{T : \mathrm{KB} \vdash \bigvee\{F(t) : t \in T\}\})$$

Not all open query formulas can be answered sensibly. We therefore require that queries are *evaluable*.[7] Answers to evaluable queries on the basis of definite KBs may be computed by means of relational algebra operations, such as projection, selection, set difference, union, and join. For instance,

$$\mathsf{Ans}(\mathrm{KB}, F(x,y) \wedge G(y,z)) = \mathsf{Ans}(\mathrm{KB}, F(x,y)) \bowtie \mathsf{Ans}(\mathrm{KB}, G(y,z))$$

In many cases, it is useful to be able to update by a set of inputs and we 'overload' the symbol Upd to denote also this more general update operation

$$\mathsf{Upd} : L_{\mathrm{KB}} \times 2^{L_{\mathrm{Input}}} \to L_{\mathrm{KB}}$$

which has to be defined in such a way that for any finite $A \subseteq L_{\mathrm{Input}}$, $\mathsf{Upd}(\mathrm{KB}, A) = \mathsf{Upd}(\mathrm{KB}, \bigwedge A)$. We sometimes write $\mathrm{KB} + F$ as an abbreviation of $\mathsf{Upd}(\mathrm{KB}, F)$, resp. $\mathrm{KB} - F$ as an abbreviation of $\mathsf{Upd}(\mathrm{KB}, -F)$.

Example 1 (Factbases) *A KB consisting of ground literals (viewed as positive and negative facts) is called a* factbase. *For instance, the factbase*

$$X_1 = \{r(S), r(P), s(S), \sim s(L), \sim s(T),\ m(P,L),\ m(T,S)\}$$

may represent the information that Susan and Peter are residents, Susan is a smoker, Linda and Tom are nonsmokers, Peter is married with Linda, and Tom is married with Susan.

As a kind of natural deduction from positive and negative facts an inference relation $\vdash$ between a factbase $X \subseteq \mathrm{Lit}$ and a sentence is defined in the following way:[8]

$$\begin{array}{lrll}
(\vdash a) & X \vdash a & \textit{if} & a \in X \\
(\vdash \sim a) & X \vdash \sim a & \textit{if} & \sim a \in X \\
(\vdash -l) & X \vdash -l & \textit{if} & l \notin X \\
(\vdash \wedge) & X \vdash F \wedge G & \textit{if} & X \vdash F \ \ \&\ \ X \vdash G \\
(\vdash \vee) & X \vdash F \vee G & \textit{if} & X \vdash F \ \textit{or}\ X \vdash G \\
(\vdash |) & X \vdash F|G & \textit{if} & X \vdash F \wedge -G \ \textit{or}\ X \vdash G \wedge -F \\
(\vdash \exists) & X \vdash \exists x F(x) & \textit{if} & X \vdash F(c)\ \textit{for some constant}\ c
\end{array}$$

For instance, one might ask X_1 "who is married with a non-resident ?",

$$\mathsf{Ans}(X_1,\ \exists y[m(x,y) \wedge -r(y)]) = \{\langle P \rangle\}$$

answered by "Peter", or "who is married with a nonsmoker ?",

$$\mathsf{Ans}(X_1,\ \exists y[m(x,y) \wedge \sim s(y)]) = \{\langle P \rangle\}$$

[7] See [vGT91] for the notion of *evaluable*, resp. *domain-independent*, formulas.

[8] This inductive definition is completed by the DeMorgan-style rewrite rules listed in section 1.2.

also answered by "Peter". The only updates we consider are insertions, $\mathsf{Upd}(X,l) := X \cup \{l\}$, *of literals. For* $K \subseteq \mathrm{Lit}$, *we have* $\mathsf{Upd}(X,K) = X \cup K$. *The knowledge system of factbases is then defined as*

$$\boldsymbol{F} \;:=\; \langle\, 2^{\mathrm{Lit}}, \vdash, L(-,\sim,\wedge,\vee,|,\exists), \mathsf{Upd}, \mathrm{Lit}\,\rangle$$

The restricted system where weak negation and exclusive disjunction are excluded from the query language, $L_{\mathrm{Query}} = L(\sim,\wedge,\vee)$, *is denoted by* $\boldsymbol{F}^{+}$.

It is easy to check that KS1, KS2 and KS3 hold.

Observation 4 *The inference relation of* $\boldsymbol{F}$ *captures model-based reasoning, resp. minimal reasoning on the basis of definite knowledge, in partial logic. Every factbase corresponds to the diagram of a partial Herbrand model, and for every* $F \in L(-,\sim,\wedge,\vee,|,\exists)$ *and every* $\mathcal{I} \in \boldsymbol{I}_4$, *it holds that* $D_{\mathcal{I}} \vdash F$ *iff* $\mathcal{I} \models F$, *and consequently, for every* $X \subseteq \mathrm{Lit}$,

$$X \vdash F \quad \textit{iff} \quad X \models_4^m F$$

where $\models_4^m$ *denotes entailment based on 4-valued minimal models (see definition 5).*

Whenever we deal with both kinds of negation, the strong negation $\sim$ is the principal negation operator (expressing explicit falsity), and the weak negation $-$ is an auxiliary negation operator (used, e.g., to express the CWA, see below). Sometimes we shall also make use of the symbol $\neg$ standing either for classical negation (which differs from both $\sim$ and $-$), or for an arbitrary negation.

2.1 Regular Knowledge Systems

In order to compare knowledge bases in terms of their information content we assume that there is an *information*, or *knowledge ordering* $\leq$ between KBs such that

$$\mathrm{KB}_1 \leq \mathrm{KB}_2 \quad \textit{if } \mathrm{KB}_2 \textit{ contains at least as much information as } \mathrm{KB}_1.$$

The information ordering should be defined in terms of the structural components of knowledge bases and not in terms of higher-level notions (like derivability).[9] The informationally *empty KB* will be denoted by 0. By definition, $0 \leq X$ for all $X \in L_{\mathrm{KB}}$, i.e. 0 is the least element of $\langle L_{\mathrm{KB}}, \leq\rangle$.

In general, more information does not mean more consequences. In other words: answers are not necessarily preserved under growth of information. Queries, for which this is the case, are called *persistent*.

Definition 8 (Persistent Queries) *A closed, resp. open, query formula* F *is called* persistent *if* $\forall X_1, X_2 \in L_{\mathrm{KB}} : X_1 \vdash F \;\Rightarrow\; X_2 \vdash F$, *resp.* $\mathsf{Ans}(X_1,F) \subseteq \mathsf{Ans}(X_2,F)$, *whenever* $X_1 \leq X_2$. *If all* $F \in L_{\mathrm{Query}}$ *are persistent, the KS and its inference relation* $\vdash$ *are called persistent. The set of all persistent query formulas is denoted by* L_{PersQ}. *An operator of the query language is called persistent, if every query formed with it and with persistent subformulas is again persistent.*

[9] The usual way to compare the information content of two KBs in standard logic, namely by means of checking the inclusion of consequences: $\mathrm{KB}_1 \leq \mathrm{KB}_2$ if $C(\mathrm{KB}_1) \subseteq C(\mathrm{KB}_2)$, does not work in a nonmonotonic setting.

Definition 9 (Ampliative Inputs) *An input formula F is called (i)* ampliative[10] *if* $\mathrm{KB} \leq \mathsf{Upd}(\mathrm{KB}, F)$*, or (ii)* reductive *if* $\mathrm{KB} \geq \mathsf{Upd}(\mathrm{KB}, F)$*. A KS and its update operation* Upd *are called* ampliative*, if all inputs $F \in L_{\mathrm{Input}}$ are ampliative. The set of all ampliative input formulas is denoted by L_{AmpI}.*

A certain subset $L_{\mathrm{Unit}} \subseteq L_{\mathrm{Input}}$ designates those elementary expressions which will be called *information units*, e.g. atoms, literals, or weighted (resp. labelled, or annotated) atoms, and the like. An information unit represents an elementary piece of information with a positive information content. A knowledge base may contain contradictory pieces of information, and we assume that all inconsistent information units contained in $X \in L_{\mathrm{KB}}$ are collected by $\mathrm{Inc}(X) \subseteq L_{\mathrm{Unit}}$.

Definition 10 (Regular KS) *A knowledge system $\boldsymbol{K}$ is called* regular*, if there is a preorder $\langle L_{\mathrm{KB}}, \leq, 0\rangle$ with least element 0, a designated set $L_{\mathrm{Unit}} \subseteq L_{\mathrm{Input}}$, and an operation $\mathrm{Inc} : L_{\mathrm{KB}} \rightarrow 2^{L_{\mathrm{Unit}}}$, such that*

(KS4) Unit inputs increase the information content (at least if they are consistent): $X \leq X + u$, for any $X \in L_{\mathrm{KB}}$, and for any $u \in L_{\mathrm{Unit}}$, such that $u \notin \mathrm{Inc}(X)$, and $\mathrm{Inc}(X+u) \subseteq \mathrm{Inc}(X)$.

(KS5) The information ordering is compatible with ampliative update and persistent inference: for all $X_1, X_2 \in L_{\mathrm{KB}}$,

$$X_1 \leq X_2 \quad \textit{iff} \quad \forall F \in L_{\mathrm{AmpI}} \forall G \in L_{\mathrm{PersQ}} : X_1 + F \vdash G \;\Rightarrow\; X_2 + F \vdash G$$

(KS6) Ampliative inputs are persistent queries: $L_{\mathrm{AmpI}} = L_{\mathrm{PersQ}} \cap L_{\mathrm{Input}}$.

(KS7) Consistent Inference: for any $X \in L_{\mathrm{KB}}$, and any $F \in L_{\mathrm{Input}}$, $X \vdash F$ implies $\mathrm{Inc}(X+F) \subseteq \mathrm{Inc}(X)$.

A regular KS will be represented as a 9-tuple

$$\langle\, 0,\, \leq,\, L_{\mathrm{KB}},\, \vdash,\, L_{\mathrm{Query}},\, \mathsf{Upd},\, L_{\mathrm{Input}},\, \mathrm{Inc},\, L_{\mathrm{Unit}} \,\rangle$$

Definition 11 (Consistency)

1. KB *is called* consistent*, if* $\mathrm{Inc}(\mathrm{KB}) = \emptyset$.
2. $F \in L_{\mathrm{Input}}$ *is called* consistent*, if* $\mathrm{Inc}(0 + F) = \emptyset$.
3. $F \in L_{\mathrm{Input}}$ *is called* consistent within $X \in L_{\mathrm{KB}}$*, if*
   (a) *F is consistent, and*
   (b) $\mathrm{Inc}(X + F) \subseteq \mathrm{Inc}(X)$ *(requiring that F does not increase the inconsistency of X), and*
   (c) *for every $u \in \mathrm{Inc}(X)$, $0 + F - u \vdash F$ (requiring that the information of F is not already inconsistent in X).*

[10] The name is adopted from [Bel77].

Example 2 (Standard Logics) *A standard logic (such as classical, or intuitionistic, logic), given by a language L and a consequence relation $\vdash \subseteq 2^L \times L$, resp. by the corrresponding consequence operation C, can be viewed as an infinitary knowledge system*

$$\langle \emptyset, \subseteq, \{X \in 2^L : X = C(X)\}, \vdash, L, \mathsf{Upd}, L, \mathrm{Inc}, L \rangle$$

where 1) a KB is a deductively closed set of formulas, 2) the knowledge ordering is set inclusion, 3) update by F is the addition of F and subsequent closure, i.e. $\mathsf{Upd}(X, F) = C(X \cup \{F\})$,[11] *4) the query, unit and input languages are all equal to L, and 5) $\mathrm{Inc}(X) = \emptyset$ if $X \neq L$, and $\mathrm{Inc}(X) = X$ otherwise. KS1–KS7 hold, more or less, trivially. Notice, however, that it is not clear whether a standard logic corresponds to a sensible finitary knowledge system, because set inclusion is no longer an adequate knowledge ordering if KBs are not deductively closed (KS5 is violated).*

Example 3 (Factbases) *The knowledge system of factbases is also regular:*

$$\boldsymbol{F} := \langle \emptyset, \subseteq, 2^{\mathrm{Lit}}, \vdash, L(-, \sim, \wedge, \vee, |, \exists), \mathsf{Upd}, \mathrm{Lit}, \mathrm{Inc}, \mathrm{Lit} \rangle$$

where $\mathrm{Inc}(X) = X \cap \widetilde{X}$. We have to show that KS4–KS6 hold. Proof: it is obvious that KS4 holds. Since $L_{\mathrm{AmpI}} = \mathrm{Lit}$, and $L_{\mathrm{PersQ}} = L(\sim, \wedge, \vee)$, KS5 follows by straightforward induction on the complexity of queries (it corresponds to the *permanence principle* of partial logic). KS6 and KS7 are again obvious. □

2.2 Formal Properties of Knowledge Systems

2.2.1 Basic Properties of Inference and Update

Definition 12 (Constructive Inference) *Let $L \subseteq \mathrm{Lit}$ be an arbitrary set of literals, and $\lceil L \rceil := \mathsf{Upd}(0, L)$. Then $\vdash$ is called* constructive *if it satisfies (i) constructible truth, and (ii) constructible falsity, i.e. both of the following conditions: for any $F, G \in L_{\mathrm{Query}}$: (i) $\lceil L \rceil \vdash F \vee G$ implies $\lceil L \rceil \vdash F$ or $\lceil L \rceil \vdash G$; (ii) $\lceil L \rceil \vdash \sim (F \wedge G)$ implies $\lceil L \rceil \vdash \sim F$ or $\lceil L \rceil \vdash \sim G$.*

The property of constructive inference guarantees that, on the basis of definite knowledge, query formulas are decomposable. Obviously, the first condition (of constructible truth) excludes the possibility of certain disjunctive tautologies such as the classical *tertium non datur*, whereas its negative counterpart excludes, for instance, the dual *principle of contradiction.*

The next property (due to Urbas [Urb90]) excludes the possibility of trivial inferences, i.e. non-tautological inferences which are solely based on the form of a KB and a query and not on their content. For example, $\{s(L), \sim s(L)\} \vdash m(P, S)$ is such a trivial inference which is valid in classical logic, i.e. from contradictory information on Linda being a smoker, we may infer that Peter is married with Susan, and thus we would get (infinitely many) unsensible answers to any query. This is clearly undesirable in a knowledge system.

Definition 13 (Tautology) *$F \in L_{\mathrm{Query}}$ is called a* tautology *in a knowledge system, if $X \vdash F$ for all $X \in L_{\mathrm{KB}}$.*

[11] Notice that this corresponds to the AGM *expansion* of 'belief sets', see [Gär88].

Definition 14 (Non-Explosive Inference) *An inference relation* $\vdash$ *is called* non-explosive *if for every non-tautology* $F \in L_{\text{Query}}$*, and for every knowledge base* $X > 0$*, there is a variant* F' *of* F *(obtained by uniform substitution of propositional constituents) such that* $X \not\vdash F'$.

While most standard logics allow for trivial inferences, their positive fragments and certain paraconsistent logics, such as Belnap's [Bel77] four-valued, or Nelson's [AN84] paraconsistent constructive logic, are non-explosive.

The following important property guarantees the freedom of knowledge base evolution.

Definition 15 (Input Completeness) *A KS is called* input complete *if*

$$\forall X_1, X_2 \in L_{\text{KB}}\, \exists A \subseteq L_{\text{Input}} : X_2 = \mathsf{Upd}(X_1, A)$$

Observation 5 *A KS is input complete iff KBs are both input constructible and input destructible, i.e. both of the following conditions hold:*

$$\begin{array}{ll} (i) & \forall X \in L_{\text{KB}}\, \exists A \subseteq L_{\text{Input}} : X = \mathsf{Upd}(0, A) \\ (ii) & \forall X \in L_{\text{KB}}\, \exists A \subseteq L_{\text{Input}} : \mathsf{Upd}(X, A) = 0 \end{array}$$

Further formal properties of knowledge systems are discussed in [Wag94b].

2.2.2 Nonmonotonicity

The following definition captures the idea that a system is considered *monotonic* if all consequences of a KB are preserved after it is updated by some new piece of information.

Definition 16 (Monotonicity) *A KS is called* monotonic *if for all* $X \in L_{\text{KB}}$*, and all* $F \in L_{\text{Input}}$*, we have* $C(X) \subseteq C(\mathsf{Upd}(X, F))$.

Though fundamental in the theory of consequence operations due to Tarski, this is too strong a requirement for knowledge systems in general.

There are two 'parameters' on which Monotonicity depends: the update operation may be ampliative or not, and the inference relation may be persistent or not.

Observation 6 *A KS is monotonic if it is ampliative and persistent.*[12]

Proof: For any $X \in L_{\text{KB}}$, and any $F \in L_{\text{Input}}$, we get $X \leq \mathsf{Upd}(X, F)$ by Ampliative Update, and consequently $C(X) \subseteq C(\mathsf{Upd}(X, F))$, by Persistent Inference. □

Practical systems will be nonmonotonic since they will allow for non-persistent queries (by means of negation-as-failure) and for non-ampliative updates (by means of deletion, resp. contraction).

[12] Or, rather exotically, if all inputs are reductive and all queries are 'antipersistent', i.e. preserved under information decrease. It is still an open problem, whether – or under which conditions – the converse holds.

2.3 Vivid Knowledge Systems

The idea of *vividness* as a design principle for knowledge systems was first proposed by Levesque in [Lev86]. However, while for Levesque the main issue was to have *complete* information, we have generalized and redefined the notion of vividness based on two fundamental principles: cognitive adequacy and computational feasibility.[13]

Definition 17 (VKS) *A knowledge system* $\boldsymbol{K}$ *is called a* vivid knowledge system (VKS)*, if it is a conservative extension of* $\boldsymbol{A}$*, the system of relational databases, i.e. if there are mappings* $f : 2^{\mathrm{At}} \to L_{\mathrm{KB}}$*, and* $g : L(-,\wedge,\vee,\exists) \to L_{\mathrm{Query}}$*, such that for any relational database* $X \subseteq \mathrm{At}$*, any input* $F \in \mathrm{At} \cup \overline{\mathrm{At}}$*, and any query* $G \in L(-,\wedge,\vee,\exists)$*,*

$$\mathsf{Upd}(X,F) \vdash G \quad \textit{iff} \quad \mathsf{Upd}(\hat{X},\hat{F}) \vdash \hat{G}$$

where we abbreviate $\hat{X} = f(X)$*,* $\hat{F} = g(F)$*, and* $\hat{G} = g(G)$*.*

It is easy to check that $\boldsymbol{F}$, the system of factbases, is a VKS.

We distinguish between positive and general knowledge systems. In a positive KS, such as $\boldsymbol{A}$, only positive knowledge is represented. In a general vivid knowledge system we have two kinds of negation: in addition to the *weak* negation (being able to express default-implicit negative information in the style of negation-as-failure), there is a second negation (called *strong*) expressing explicit falsity.

It is desirable for a KS to be robust towards any possible update. This includes inputs which are inconsistent with the current knowledge base. Such inputs may be erroneous, but it might be as well the case that the new input is correct, and some old piece of information in the KB is erroneous or outdated. In any case, it seems important that the main functions of a KS are not corrupted by inconsistent inputs. Thus, a sophisticated inconsistency handling mechanism might prevent that both a query formula and its negation can ever be inferred from a KB even if the KB contains contradictory information.

Definition 18 (Inherently Consistent Inference) *An inference relation* $\vdash$ *is called* inherently consistent *with respect to a negation operator* $\neg$*, if for every* $X \in L_{\mathrm{KB}}$ *and every* $F \in L_{\mathrm{Query}}$*, it is never the case that* $X \vdash \neg F$ *and* $X \vdash F$*.*

Notice that this holds trivially if we make the restriction that L_{KB} admits only of consistent KBs. But such a restriction is not realistic. We shall assume, therefore, that L_{KB} also contains inconsistent KBs. In this case, inference in classical logic is not inherently consistent. In fact, Inherently Consistent Inference is violated by any negation satisfying the classical explosion principle *ex contradictione sequitur quodlibet*, $\{F, \neg F\} \vdash G$. But it is also violated by those paraconsistent logics where contradictions are derivable, $\{F, \neg F\} \vdash F \wedge \neg F$, such as in Belnap's 4-valued logic.

Observation 7 *Inference in* $\boldsymbol{F}$ *is inherently consistent wrt weak negation:*

$$\forall X \in 2^{\mathrm{Lit}} \forall F \in L(-,\sim,\wedge,\vee) : X \not\vdash F \wedge -F$$

[13] This is further discussed in [Wag94].

Although we do not make it a strict requirement, it seems desirable for a KS that the following *coherence*[14] property, relating weak with strong negation, holds.

Definition 19 (Negation Coherence) *A knowledge system with weak and strong negation satisfies* Negation Coherence, *if for any consistent* $X \in L_{\mathrm{KB}}$*, and any* $F \in L_{\mathrm{Query}}$, $X \vdash -F$ *whenever* $X \vdash \sim F$.

Definition 20 (Minimal Change) [15] *Let* $u \in L_{\mathrm{Unit}}$ *be any information unit. We say that* Upd *satisfies the principle of* Minimal Change *if it satisfies both Minimal Change for* Unit Expansion *and for* Unit Contraction. *For any* $X, X' \in L_{\mathrm{KB}}$*, and any* $u \in L_{\mathrm{Unit}}$*, such that* u *is consistent with* X*, we require the following:*

(Minimal Change for Unit Expansion) $X + u := \mathsf{Upd}(X, u)$ *is the least extension of* X *such that* u *can be inferred, expressed by the conditions* (+1) *and* (+2)*:*

$$\begin{array}{ll} (+1) & X + u \geq X \\ (+2) & X' \geq X \ \ \& \ \ X' \vdash u \ \Rightarrow \ X + u \leq X' \end{array}$$

(Minimal Change for Unit Contraction) $X - u := \mathsf{Upd}(X, -u)$ *is the greatest reduction of* X *such that* $-u$ *can be inferred, expressed by* (-1) *and* (-2)*:*

$$\begin{array}{ll} (-1) & X - u \leq X \\ (-2) & X' \leq X \ \ \& \ \ X' \vdash -u \ \Rightarrow \ X - u \geq X' \end{array}$$

2.4 A Further Example: Deductive Factbases

A *deductive factbase* is a pair $\langle X, R \rangle$ consisting of a factbase X and a set R of range-restricted rules, called *deduction rules.* A non-ground rule $r = l \leftarrow F$ may be formed with any conclusion formula $l \in \mathrm{Lit}^x$, and any premise formula $F \in L^x(-, \sim, \wedge, \vee, |, \exists)$, such that 1) $\mathrm{Free}(l) \subseteq \mathrm{Free}(F)$, and 2) F is evaluable. The set of such rules which are called *range-restricted* is denoted by $R(\mathrm{Lit} \leftarrow L(-, \sim, \wedge, \vee, |, \exists))$.

For a non-ground rule $r = l(x) \leftarrow F(x)$, its application to a fact base X is defined by

$$r(X) \ := \ \mathsf{Upd}(X, \ \{l(c) : c \in \mathit{Const} \ \ \& \ \ X \vdash F(c)\})$$

In the basic setting, deduction rules are not affected by updates, i.e. only 'extensional' predicates may be updated:

$$\mathsf{Upd}_d(\langle X, R \rangle, \ l) \ = \ \langle \mathsf{Upd}(X, l), \ R \rangle$$

But deduction rules help to answer queries:

$$\langle X, R \rangle \vdash_d F \quad \text{iff} \quad R(X) \vdash F$$

where we assume that $R(X) \subseteq \mathrm{Lit}$ is the unique intended deductive closure of X under R, according to the following definition.

[14]The name is adopted from [PA92].

[15]This principle was already proposed in [Bel77] under the name of *minimal mutilation*. It is also one of the fundamental principles of AGM-style *belief revision*, see e.g. [Gär88].

Definition 21 (Deductive Closure) *A factbase $Z \in 2^{\mathrm{Lit}}$ is called a* deductive closure *of a deductive factbase $\langle X, R\rangle$, if*

1. *Z is closed under R, i.e. $r(Z) = Z$ for all $r \in R$, and*
2. *Z is supported, i.e. $Z = r_n \circ \ldots \circ r_1(X)$, for some sequence of ground rules $(r_i)_{1 \le i \le n} \subseteq [R]$, and*
3. *$Z \supseteq X$, i.e. Z contains all 'facts' from X.*

where $[R]$ denotes the instantiation of R. If all rules in R have persistent premise formulas, there is a unique minimal closure of a factbase X under R which is naturally the intended one. It can be computed by successively detaching applicable rules until all rules are satisfied. If R contains rules with non-persistent premise formulas, there may be several minimal closures, and not all of them might be intended.[16]

Example 4 *In the factbase X_1 from example 1, we cannot infer that Peter is not married with Susan because the CWA does not hold for* married. *However, one could argue, that since both Peter and Susan are residents, it should suffice that there is no record of their marriage in the local KB in order to conclude that they really are not married. This can be expressed by the following deduction rule*

$$r_1 \quad = \quad \sim m(x,y) \leftarrow r(x) \wedge r(y) \wedge -m(x,y)$$

which yields the deductive factbase $\langle X_1, \{r_1\}\rangle$. *We obtain the inference*

$$\langle X_1, \{r_1\}\rangle \vdash \sim m(P,S)$$

since the unique intended closure of X_1 under $\{r_1\}$, $r_1(X_1)$, contains $\sim m(P,S)$:

$$r_1(X_1) = X_1 \cup \{\sim m(S,P),\ \sim m(P,S)\}$$

Formally, the system of deductive factbases is defined as

$$\boldsymbol{DF} := \langle\, \langle \emptyset, \emptyset\rangle,\ \le_d,\ 2^{\mathrm{Lit}} \times 2^{R(\mathrm{Lit} \leftarrow L(-,\sim,\wedge,\vee,|,\exists))},\ \vdash_d,\ L(-,\sim,\wedge,\vee,|,\exists),\ \mathsf{Upd}_d,\ \mathrm{Lit},\ \mathrm{Inc}_d,\ \mathrm{Lit}\,\rangle$$

The knowledge ordering of deductive factbases $\le_d$ is defined by

$$\langle X, R\rangle \le_d \langle X', R'\rangle \quad :\Longleftrightarrow \quad \forall l \in \mathrm{Lit} : R(\mathsf{Upd}(X,l)) \subseteq R'(\mathsf{Upd}(X',l))$$

The inconsistency measure Inc_d is defined by $\mathrm{Inc}_d(\langle X, R\rangle) = \mathrm{Inc}(R(X))$.

[16] In general, the intended closures are selected from the set of all closures by means of an appropriate preference criterion, such as the *stability* of rule application as in the *stable closure semantics* of [Wag94b].

3 Belnap's Epistemic States

A KS, according to Belnap, answers queries by invoking some inference mechanism, and it accepts input from a variety of sources by using an appropriate update mechanism. In such circumstances inconsistency threatens. Mr. X tells the KS that A while Mrs. Y tells it that $\sim A$. Or, in a different environment, an automatic measurement yields that $m > 0.3$ while the subsequent confirmation attempt yields that $m < 0.3$.

What is the KS to do ?

Possibility 1: Refuse to accept inconsistent information. However: this is unfair either to Mr. X or to Mrs. Y. Also, contradictions may not lie on the surface.

Possibility 2: Revise current beliefs in the presence of contradictions. However: it seems to be difficult to determine the proper revision policy doing justice to both Mr. X and Mrs. Y, and it seems to be even more difficult to mechanize it in a satisfactory way.

Possibility 3: Just accept contradictions and report them exactly as they were told, so the user can make up her mind.

Belnap advocates possibility 3, but emphasizes that even if the ultimate goal is possibility 2, i.e. revision, possibility 3 is a good first step towards that goal.

In order to be able to process arbitrary input formulas $F \in L(\sim, \wedge, \vee)$, an epistemic state has to consist of a set of possible situation descriptions (corresponding to partial interpretations, called 'set-ups' by Belnap). Thus, an epistemic state Y is a subset of 2^{Lit}, and the elements of Y represent different epistemic alternatives. For instance, if we know that Susan is a nonsmoker and that Peter is married either with Linda or with Susan, we get the following epistemic state:

$$Y_1 = \{\{\sim s(S),\, m(P,L)\},\, \{\sim s(S), m(P,S)\}\}$$

3.1 Formalization

Following Belnap, yet formulated in a different way, inputs to an epistemic state are processed as follows:

$$\begin{aligned}
\mathsf{Upd}_B(Y,l) &= \{X \cup \{l\} : X \in Y\} \\
\mathsf{Upd}_B(Y,F \wedge G) &= \mathsf{Upd}_B(\mathsf{Upd}_B(Y,F),G) \\
\mathsf{Upd}_B(Y,F \vee G) &= \mathsf{Upd}_B(Y,F) \cup \mathsf{Upd}_B(Y,G)
\end{aligned}$$

All other cases of compound formulas are treated by DeMorgan-style rewriting (see the rewrite rules above). Notice that Upd_B may create inconsistent situation descriptions even if there are consistent alternatives. It seems, however, more appropriate to discard inconsistent situation descriptions if they are not *minimally inconsistent*,[17] i.e. if there are alternatives with 'less inconsistency'. For this purpose we define

$$\begin{aligned}
\mathrm{Cons}(Y) &:= \{X \in Y \mid X \cap \widetilde{X} = \emptyset\} \\
\mathrm{MInc}(Y) &:= \{X \in Y \mid \neg\exists Z \in Y : (Z \cap \widetilde{Z}) \subset (X \cap \widetilde{X})\}
\end{aligned}$$

[17]The principle of *minimal inconsistency* was proposed in [Pri89].

The first operator, Cons, accepts only consistent situation descriptions, while the second one, MInc, accepts all situation descriptions which are minimally inconsistent. If an epistemic state Y is consistent, i.e. if it contains at least one consistent epistemic alternative, then $\text{Cons}(Y) = \text{MInc}(Y)$. We can now define

$$\begin{aligned} \mathsf{Upd}_{ex}(Y,F) &:= \text{Cons}(\mathsf{Upd}_B(Y,F)) \\ \mathsf{Upd}_{mi}(Y,F) &:= \text{MInc}(\mathsf{Upd}_B(Y,F)) \end{aligned}$$

Upd_{ex} does not accept inconsistent inputs at all. It implements the *ex contradictione sequitur quodlibet (ECSQ)* principle of classical logic by discarding all inconsistent epistemic alternatives. A good compromise between the hypersensitive inconsistency handling mechanism of Upd_{ex} and the too liberal Upd_B is the principle of *minimal inconsistency* proposed in [Pri89], and implemented by Upd_{mi}.

A query formula $F \in L(\sim, \wedge, \vee)$ can be inferred from an epstemic state Y if it can be inferred from every possible situation description $X \in Y$:

$$Y \vdash F \quad :\Longleftrightarrow \quad \text{for all } X \in Y : X \vdash F,$$

where $X \vdash F$ is the inference relation of $\boldsymbol{F}^+$.

The process of information growth can be captured by the following notion of informational extension. An epistemic state Y' is called an *(informational) extension* of Y, symbolically $Y \leq Y'$, if every epistemic alternative in Y' extends one in Y:

Definition 22 (Information Ordering) $\quad Y \leq Y' :\Longleftrightarrow \forall X' \in Y' \, \exists X \in Y : X \subseteq X'$

This ordering, used in [Bel77], was also suggested in domain theory for the semantics of parallel processes (see [Plo76, Smy78]), and is sometimes called 'Smyth ordering'.

Observation 8 $\quad \langle 2^{2^{\text{Lit}}}, \leq \rangle$ *is a preorder with least element* $0 := \{\emptyset\}$.

Notice that while $\{\text{Lit}\}$ is informationally larger than any non-empty epistemic state $Y \subseteq 2^{\text{Lit}}$, the empty epistemic state is still larger:

$$Y \leq \{\text{Lit}\} < \{\}$$

The elementary pieces of information in epistemic states are disjunctions of literals $l_1 \vee \ldots \vee l_m$. The set of all such disjunctions is denoted by $\text{Lit}^{\vee}$. The inconsistency operation Inc_B now collects all definite and indefinite contradictions:

$$\text{Inc}_B(Y) := \{\bigvee L : L \in \text{Min}(\{K \subseteq \text{Lit} \mid \forall X \in Y \exists l \in K : l \in X \cap \widetilde{X}\})\}$$

Definition 23 (Belnap's KS)

$$\boldsymbol{B} := \langle \{\emptyset\}, \leq, 2^{2^{\text{Lit}}}, \vdash, L(\sim, \wedge, \vee), \mathsf{Upd}_B, L(\sim, \wedge, \vee), \text{Inc}_B, \text{Lit}^{\vee} \rangle$$

is called Belnap's KS. *Besides* $\boldsymbol{B}$*, we also define*

$\boldsymbol{B}_{mi}$ *with* $L_{\text{KB}} = \{Y \in 2^{2^{\text{Lit}}} : Y = \text{MInc}(Y)\}$*, and* Upd_{mi}*, and*

$\boldsymbol{B}_{ex}$ *with* $L_{\mathrm{KB}} = \{Y \in 2^{2^{\mathrm{Lit}}} : Y = \mathrm{Cons}(Y)\}$ *and* Upd_{ex}.

We have to show that KS1–KS6 are satisfied.

KS1 and KS2 are obvious.

KS3: Reflexivity is proved by induction on input formulas. Clearly, $\mathsf{Upd}(Y, l) \vdash l$, since l is added to every element of Y. For conjunctive inputs we obtain $\mathsf{Upd}(Y, F \wedge G) = (Y + F) + G \vdash G$, by the induction hypothesis, and since $\boldsymbol{B}$ is persistent and ampliative (see the resp. observations below), it follows from $Y + F \vdash F$ that $(Y + F) + G \vdash F$, and hence $(Y + F) + G \vdash F \wedge G$. For disjunctive inputs, $\mathsf{Upd}(Y, F \vee G) = (Y + F) \cup (Y + G) \vdash F \vee G$, since for every $X \in (Y + F)$, it holds by the induction hypothesis that $X \vdash F$, hence $X \vdash F \vee G$, and similarly for every $X \in (Y + G)$.

KS4: easy to check.

KS5: It suffices to show that $Y_1 \leq Y_2$ iff $C(Y_1) \subseteq C(Y_2)$. The ($\Rightarrow$)-part, i.e. Persistent Inference, is proved below. We show the contraposition of the ($\Leftarrow$)-part. Assume $Y_1 \not\leq Y_2$, i.e. $\exists X_2 \in Y_2 \forall X_1 \in Y_1 \exists l \in X_1 - X_2$. Let X_2' be such an element of Y_2. Then we can construct a formula $F = \bigvee\{l \in \mathrm{Lit} \mid \exists X_1 \in Y_1 : l \in X_1 - X_2'\}$, for which $Y_1 \vdash F$, but $Y_2 \not\vdash F$.

KS6: Follows immediately from the fact that $\boldsymbol{B}$ is persistent and ampliative (see the resp. observations below). □

Observation 9 *Only the minimal elements of an epistemic state count:* $C(Y) = C(\mathrm{Min}(Y))$.

Proof: In $\boldsymbol{F}^+$, for $X, X' \subseteq \mathrm{Lit}$ it holds that $C(X) \subseteq C(X')$ whenever $X \subseteq X'$. Consequently,

$$C(Y) = \bigcap\{C(X) : X \in Y\} = \bigcap\{C(X) : X \in \mathrm{Min}(Y)\} \qquad \Box$$

Observation 10 (Persistent Inference) *In* $\boldsymbol{B}$, $\vdash$ *is persistent, i.e. if* $Y_1 \leq Y_2$ *then* $Y_2 \vdash F$ *whenever* $Y_1 \vdash F$ *for all* $F \in L(\sim, \wedge, \vee)$.

Proof: Let $Y_1 \leq Y_2$. Then,

$$\begin{aligned} C(Y_1) &= \bigcap\{C(X_1) \mid X_1 \in Y_1\} \\ &\subseteq \bigcap\{C(X_1) \mid X_1 \in Y_1 \ \& \ \exists X_2 \in Y_2 : X_1 \subseteq X_2\} \\ &\subseteq \bigcap\{C(X_2) \mid X_2 \in Y_2 \ \& \ \exists X_1 \in Y_1 : X_1 \subseteq X_2\} \\ &= \bigcap\{C(X_2) \mid X_2 \in Y_2\} \\ &= C(Y_2) \quad \Box \end{aligned}$$

Observation 11 (Ampliative Update) $Y \leq \mathsf{Upd}_B(Y, F) \leq \mathsf{Upd}_{mi}(Y, F) \leq \mathsf{Upd}_{ex}(Y, F)$

Proof: It suffices to show that $Y \leq \mathsf{Upd}_B(Y, F)$ by straightforward induction on the complexity of F. □

Claim 1 $\boldsymbol{B}$, $\boldsymbol{B}_{mi}$ *and* $\boldsymbol{B}_{ex}$ *satisfy Monotonicity, i.e.* $C(Y) \subseteq C(\mathsf{Upd}(Y, F))$.

Proof: The assertion follows as a corollary from the two previous observations and observation 6. □

3.2 Relation to Standard Logics

Recall that DNS(F) denotes the *disjunctive normal set* corresponding to the disjunctive normal form of F.

Observation 12

1. $\mathsf{Upd}_B(Y, F) = \{X \cup K : X \in Y, K \in \mathrm{DNS}(F)\}$.
2. $\mathsf{Upd}_B(0, F) = \mathrm{DNS}(F)$.

Proof: See [Wag94]. This observation implies that Contraction and Permutation hold in $\boldsymbol{B}$.

Claim 2 (Four-Valued Logic) *Let* $X \in \mathrm{Fin}(2^{L(\sim,\wedge,\vee)})$*, and* $F \in L(\sim, \wedge, \vee)$*, then*

$$X \models_4 F \quad \textit{iff} \quad \mathsf{Upd}_B(0, X) \vdash F$$

Proof: From the previous observation it follows that $\mathrm{Mod}_4^m(X) \subseteq \mathsf{Upd}_B(0, X) \subseteq \mathrm{Mod}_4(X)$. This, together with the fact that $\models_4$ is determined by minimal models, i.e.

$$\mathrm{Mod}_4^m(F) \subseteq \mathrm{Mod}_4(G) \;\Rightarrow\; \mathrm{Mod}_4(F) \subseteq \mathrm{Mod}_4(G)$$

yields the assertion. □

Claim 3 (Three-Valued Logic) *Recall that standard (= Kleene's strong) 3-valued propositional logic corresponds to* $\models_c$.

$$X \models_c F \quad \textit{iff} \quad \mathsf{Upd}_{ex}(0, X) \vdash F$$

and moreover, if X *is consistent, then*

$$X \models_c F \quad \textit{iff} \quad \mathsf{Upd}_{mi}(0, X) \vdash F$$

Proof: If X is not consistent, i.e. it does not have a coherent model, then $\mathsf{Upd}_{ex}(0, X) = \{\}$, and the assertion holds trivially. Otherwise every minimal coherent model of X corresponds to a consistent element of $\mathsf{Upd}_B(0, X)$, and

$$\mathrm{Mod}_c^m(X) \subseteq \mathsf{Upd}_{ex}(0, X) = \mathsf{Upd}_{mi}(0, X) \subseteq \mathrm{Mod}_c(X)$$

This, together with the fact that $\models_c$ restricted to $L(\sim, \wedge, \vee)$ is determined by minimal models, i.e.

$$\mathrm{Mod}_c^m(X) \subseteq \mathrm{Mod}_c(F) \;\Rightarrow\; \mathrm{Mod}_c(X) \subseteq \mathrm{Mod}_c(F)$$

yields the assertion. □

Claim 4 (Classical Logic) *Let* X, F *be as before, let* $\mathrm{At}(X, F)$ *denote the set of all atoms occuring in* X *and* F*, and* $\models_2$ *denote classical propositional logic. Then,*

$$X \models_2 F \quad \textit{iff} \quad \mathsf{Upd}_{ex}(\mathsf{Upd}_{ex}(0, X), \{a \vee {\sim} a : a \in \mathrm{At}(X, F)\}) \;\vdash\; F$$

3.3 Problems with Belnap's KS

We briefly discuss three problems with $\boldsymbol{B}$.

3.3.1 No Unique Representation

In $\boldsymbol{B}$, epistemic states are not unique representations: $C(Y) = C(Y')$ does not imply that $Y = Y'$. For instance,

$$C(\{\{p\}\}) = C(\{\{p\},\ \{p,q\}\})$$

A possible remedy consists of admitting only minimal situation descriptions as elements of an epistemic state. An epistemic state Y is called *canonical* if $Y = \mathrm{Min}(Y)$. We denote the set of all canonical elements of a set of epistemic states $\mathcal{Y} \subseteq 2^{2^{\mathrm{Lit}}}$ by $\mathrm{Can}(\mathcal{Y})$.

We obtain the following system:

$$\boldsymbol{B}^m \ := \ \langle\, \{\emptyset\},\ \leq,\ \mathrm{Can}(2^{2^{\mathrm{Lit}}}),\ \vdash,\ L(\sim,\wedge,\vee),\ \mathsf{Upd}^m_B,\ \mathrm{Lit},\ L(\sim,\wedge,\vee)\,\rangle$$

where $\mathsf{Upd}^m_B(Y,F) := \mathrm{Min}(\mathsf{Upd}_B(Y,F))$. The collection of all canonical epistemic states forms a lattice, as was shown in [KM93].

Observation 13 $\langle \mathrm{Can}(2^{2^{\mathrm{Lit}}}), \leq\rangle$ *is a lattice order. Meet and join can be defined by* $Y_1 \sqcap Y_2 = \mathrm{Min}(Y_1 \cup Y_2)$*, and* $Y_1 \sqcup Y_2 = \mathrm{Min}(\{X_1 \cup X_2 : X_1 \in Y_1,\ X_2 \in Y_2\})$*.*

3.3.2 No Disjunctive Syllogism

For instance,

$$\mathsf{Upd}_B(0, (p \vee q) \wedge {\sim} p) = \{\{p, {\sim} p\},\ \{q, {\sim} p\}\} \not\vdash q$$

Notice that the resulting epistemic state contains an inconsistent situation description although there is a consistent epistemic alternative.

The Disjunctive Syllogism holds in $\boldsymbol{B}_{ex}$, where all inconsistent epistemic alternatives are discarded. In this system, however, if $Y \vdash l$ then $\mathsf{Upd}(Y, \tilde{l}) = \{\}$ yields an 'exploded' KB in the sense that everything follows from $\{\}$, i.e. $\boldsymbol{B}_{ex}$ is explosive. A good compromise seems to be $\boldsymbol{B}_{mi}$ which is both non-explosive and satisfies the Disjunctive Syllogism.

Example 5 *If we have the above* Y_1*, and we then learn that Susan or Linda smokes, the following update is performed in* $\boldsymbol{B}_{mi}$*:*

$$\begin{array}{rcl} Y_2 & := & \mathsf{Upd}_{mi}(Y_1, s(S) \vee s(L)) \\ & = & \mathrm{MInc}(\{\{s(S), {\sim} s(S), m(P,L)\},\ \{s(S), {\sim} s(S), m(P,S)\}, \\ & & \qquad \{s(L), {\sim} s(S), m(P,L)\},\ \{s(L), {\sim} s(S), m(P,S)\}\}) \\ & = & \{\,\{s(L), m(P,L), {\sim} s(S)\},\ \{s(L), m(P,S), {\sim} s(S)\}\,\} \\ & \vdash & s(L) \end{array}$$

3.3.3 No Input Completeness

Finite epistemic states are input constructible in $\boldsymbol{B}$, i.e.

$$\forall Y \in \mathrm{Fin}(2^{\mathrm{Fin}(2^{\mathrm{Lit}})}) \exists F \in L_{\mathrm{Input}} : Y = \mathsf{Upd}_B(0, F)$$

but since they are not input destructible, $\boldsymbol{B}$ does not satisfy Input Completeness.

4 Disjunctive Factbases

In a similar way as we have extended the monotonic system $\boldsymbol{F}^+$ to obtain Belnap's monotonic disjunctive KS in section 3, we now want to extend the nonmonotonic system $\boldsymbol{F}$ to obtain a nonmonotonic disjunctive KS. However, such an extension is not straightforward for at least two reasons. First, if we would simply extend the query language of $\boldsymbol{B}$ by adding weak negation, lemmas would be no longer compatible. This is easy to see. Consider the following example:

$$\begin{array}{ll} & \{\{p\}\} \vdash p \vee q \\ \text{but} & \mathsf{Upd}_B(\{\{p\}\}, p \vee q) = \{\{p\},\{p,q\}\} \not\vdash -q \\ \text{while} & \{\{p\}\} \vdash -q \qquad \Box \end{array}$$

Consequently, we should rather choose $\boldsymbol{B}^m$ as the basis of our extension, since

$$\mathsf{Upd}^m_B(\{\{p\}\},\, p \vee q) = \{\{p\}\}$$

But then we get another problem: disjunctive information would be always exclusive in the sense that $\mathsf{Upd}^m_B(0, p \vee q) \vdash -p \vee -q$. And this is clearly undesirable. To remedy the problem, we have to use another notion of minimality called *paraminimality* in [HJW96]. We first define $\mathrm{Min}_X(Y) = \{X' \in \mathrm{Min}(Y) : X' \leq X\}$, and by means of it an operator collecting all *paraminimal* elements:

$$\mathrm{PMin}(Y) = \{X \in Y \mid \neg\exists X' \in Y : X' < X \;\;\&\;\; \mathrm{Min}_{X'}(Y) = \mathrm{Min}_X(Y)\}$$

An ordered set Y is called *paracanonical* if $Y = \mathrm{PMin}(Y)$. The set of all paracanonical elements of a set of ordered sets $\mathcal{Y}$ is denoted by $\mathrm{PCan}(\mathcal{Y})$.

Disjunctive factbases are paracanonical epistemic states. For $Y \in \mathrm{PCan}(2^{2^{\mathrm{Lit}}})$, and $F \in L(-, \sim, \wedge, \vee, |, \exists)$, inference is defined elementwise:

$$Y \vdash F :\Longleftrightarrow \text{for all } X \in Y : X \vdash F$$

where $X \vdash F$ is inference in $\boldsymbol{F}$. For example,

$$Y_1 \vdash \sim s(S) \wedge [m(P,L) \,|\, m(P,S)]$$

The information ordering between disjunctive factbases $Y_1, Y_2 \subseteq 2^{\mathrm{Lit}}$ is the same ordering as in $\boldsymbol{B}$. Likewise, information units and the inconsistency operation are the same. Inputs are processed as follows:

$$\begin{array}{llcl} (Ul) & \mathsf{Upd}_B(Y, l) & = & \{X \cup \{l\} : X \in Y\} \\ (U\wedge) & \mathsf{Upd}_B(Y, F \wedge G) & = & \mathsf{Upd}_B(\mathsf{Upd}_B(Y, F), G) \\ (U\vee) & \mathsf{Upd}_B(Y, F \vee G) & = & \mathrm{PMin}(\mathsf{Upd}_B(Y, F) \cup \mathsf{Upd}_B(Y, G) \cup \mathsf{Upd}_B(Y, F \wedge G)) \end{array}$$

We define the same additional update operations Upd_{mi}, and Upd_{ex}, as in $\boldsymbol{B}$. The only difference with respect to update in $\boldsymbol{B}$ is the definition $(U\vee)$ for disjunctive inputs which renders disjunctive information now explicitly *inclusive*.

Definition 24 (Disjunctive Factbases) *The system of disjunctive factbases is defined as*

$$V_B\boldsymbol{F} \;:=\; \langle\, \{\emptyset\},\, \leq,\, \mathrm{PCan}(2^{2^{\mathrm{Lit}}}),\, \vdash,\, L(-, \sim, \wedge, \vee, |),\, \mathsf{Upd}_B,\, L(\sim, \wedge, \vee),\, \mathrm{Inc}_B,\, \mathrm{Lit}^{\vee}\, \rangle$$

$V_{mi}\boldsymbol{F}$ and $V_{ex}\boldsymbol{F}$ are formed from $V_B\boldsymbol{F}$ in the same way as $\boldsymbol{B}_{mi}$ and $\boldsymbol{B}_{ex}$ are formed from $\boldsymbol{B}$.

We have to show that $V_B\boldsymbol{F}$, $V_{ex}\boldsymbol{F}$ and $V_{mi}\boldsymbol{F}$ satisfy the KS postulates. KS1, KS2 and KS4 are obvious. KS3 (Reflexivity), KS5 (Knowledge Ordering Adequacy), and KS6 (Ampliative=Persistent) are proved in the same way as for $\boldsymbol{B}$, now using

$$L_{\mathrm{PersQ}} = L_{\mathrm{AmplI}} = L_{\mathrm{Input}} = L(\sim, \wedge, \vee) \quad \Box$$

Notice that by the addition of weak negation and exclusive disjunction, inference in disjunctive factbases is no longer persistent.

Conjecture 1 *Disjunctive factbases capture paraminimal reasoning in partial logic. Let X be a finite subset of $L(\sim, \wedge, \vee)$, and $F \in L(-, \sim, \wedge, \vee, |)$. Then,*

$$X \models_4^{pm} F \quad \textit{iff} \quad \mathsf{Upd}_B(0, X) \vdash F$$

where $X \models_4^{pm} F$ is defined as $\mathrm{PMin}(\mathrm{Mod}_4(X)) \subseteq \mathrm{Mod}_4(F)$.

Observation 14 *$V_B\boldsymbol{F}$ violates the principle of Negation Coherence: if, e.g., $Y = \{\{\sim p, q\}, \{p, \sim p\}\}$, then $Y \vdash \sim p$, but $Y \not\vdash -p$. On the other hand, Coherence holds in $V_{mi}\boldsymbol{F}$ and in $V_{ex}\boldsymbol{F}$.*

Observation 15 *$V_{mi}\boldsymbol{F}$ is non-explosive and negation coherent. It has (a restricted form of) the Disjunctive Syllogism. E.g.,*

$$\begin{aligned} \mathsf{Upd}_{mi}(0, (p \vee q) \wedge \sim p) &= \{\{q, \sim p\}\} \vdash q, \quad \textit{while} \\ \mathsf{Upd}_{mi}(\{\{\sim q\}\}, (p \vee q) \wedge \sim p) &= \{\{p, \sim p, \sim q\}, \{\sim p, q, \sim q\}\} \not\vdash q \end{aligned}$$

This example also shows that neither Permutation nor Update Monotonicity hold in $V_{mi}\boldsymbol{F}$.

Observation 16 *Only $V_{mi}\boldsymbol{F}$ is a vivid knowledge system. $V_{ex}\boldsymbol{F}$ is not vivid since it is explosive, $V_B\boldsymbol{F}$ is not vivid since it violates Negation Coherence.*

Claim 5 (Collapse of weak and strong negation) *Let $Y \in \mathrm{Cons}(\mathrm{PCan}(2^{2^{\mathrm{Lit}}}))$, and $a \in \mathrm{At}$. Then, whenever $Y \vdash a \vee \sim a$, weak and strong negation coincide for a in $V_{mi}\boldsymbol{F}$ and $V_{ex}\boldsymbol{F}$:*

$$Y \vdash \sim a \quad \textit{iff} \quad Y \vdash -a$$

Proof: Since Y is consistent, all $X \in Y$ are consistent, and consequently, $a \vee \sim a$ is derivable from Y iff either $a \in X$ or $\sim a \in X$ for all $X \in Y$, subsuming three cases:

1. If all $X \in Y$ contain a, then neither $Y \vdash \sim a$ nor $Y \vdash -a$.
2. If all $X \in Y$ contain $\sim a$, then both $Y \vdash \sim a$ and $Y \vdash -a$.
3. Otherwise some $X \in Y$ contain a, and all others contain $\sim a$, hence $Y \not\vdash \sim a$, and $Y \not\vdash -a$. $\Box$

Notice that this does not hold in $V_B\boldsymbol{F}$

Claim 6 *A disjunctive factbase is a unique representation, i.e. in $V_*\boldsymbol{F}$ (for $* = B, mi, ex$), it holds that $C(Y) = C(Y')$ implies $Y = Y'$.*

Proof: In [Wag94], the Unique Representation property was shown for arbitrary $Y \subseteq 2^{\mathrm{Lit}}$ (the reason for it is the availability of weak negation in the query language). $\Box$

	$\boldsymbol{F}^{+}$	$\boldsymbol{F}$	$\boldsymbol{B}$	$\boldsymbol{B}_{ex}$	$\boldsymbol{B}_{mi}$	$V_B\boldsymbol{F}$	$V_{ex}\boldsymbol{F}$	$V_{mi}\boldsymbol{F}$
Contraction	√	√	√	√	√	√	√	√
Permutation	√	√	√	√		√	√	
Update Monotonicity	√	√	√	√		√	√	
Cumulativity	√	√	√	√	√	?	?	?
Monotonicity	√		√	√	√			
Unique Representation	√	√				√	√	√
non-explosive	√	√	√		√	√		√
Negation Coherence	–	√	–	–	–		√	√
Disjunctive Syllogism	–	–		√	√		√	√

Table 1: Formal properties of some basic knowledge systems (– denotes *not applicable*, ? denotes *open problem*).

5 Exact Predicates and the Closed-World Assumption

In knowledge systems, three kinds of predicates can be distinguished. The first distinction, proposed by Körner in [Kör66], reflects the fact that many predicates (especially in empirical domains) have truth value gaps: neither $p(c)$ nor $\sim p(c)$ has to be the case for specific instances of such *inexact* predicates, like, e.g., color attributes which can in some cases not be determined because of vagueness.

Other predicates, e.g. from legal or theoretical domains, are *exact*, and we then have, for instance, $m(S) \vee \sim m(S)$ and $prime(2^{77}-1) \vee \sim prime(2^{77}-1)$, stating that Sophia is either married or unmarried and that $2^{77}-1$ is either a prime or a non-prime number. Only exact predicates can be totally represented in a knowledge base. Therefore, only exact predicates can be subject to the Closed-World Asssumption.

As in relational database theory, we distinguish between the *schema* Σ and the *instance*, resp. *state*, $X \in L_{\mathrm{KB}}$ of a vivid knowledge base. The schema determines the language of a KB, i.e. the available predicates (relation symbols) together with their domains and their epistemic category (exact or inexact). Since, in general, not all predicates are subject to the Closed-World Assumption (CWA), the schema also stipulates those relation symbols for which the CWA will be assumed. Finally, the schema contains a set of *integrity constraints*, i.e. closed query formulas which have to be satisfied by any evolving state of an associated knowledge base.[18]

Definition 25 (VKB Schema) *A vivid knowledge base schema is a quadruple*

$$\langle\, Rel,\ ExRel,\ CWRel,\ IC\,\rangle$$

consisting of a set Rel *of relation schemas, a set* ExRel $\subseteq$ Rel *of exact relation symbols, a set* CWRel $\subseteq$ ExRel *of CWA relation symbols, and a set* $IC \subseteq L_{\mathrm{Query}}$ *of integrity constraints.*

Definition 26 *For a schema* $\Sigma = \langle\, Rel,\ ExRel,\ CWRel,\ IC\,\rangle$*, we say that* Y *is a knowledge base over* Σ*, denoted* $Y : \Sigma$*, if*

1. *Y contains only predicates from* Rel*, such that the resp. domain constraints are satisfied.*

[18] We shall not discuss integrity constraints in this paper.

2. For any $F \in IC$, $Y \vdash F$.

Definition 27 (Closed-World Assumption) *For vivid knowledge bases $Y : \Sigma$, we have the following additional inference rule for inferring negative conclusions,*[19]

$$Y \vdash_\Sigma {\sim} p(c) \quad \textit{if} \quad p \in \mathit{CWRel} \;\; \& \;\; Y \vdash_\Sigma -p(c)$$

together with the corresponding extension of update,

$$\mathsf{Upd}_\Sigma(Y, {\sim} p(c)) = \mathsf{Upd}_\Sigma(Y, -p(c)) \quad \textit{if} \quad p \in \mathit{CWRel}$$

Notice that this CWA inference rule refers to both the state and the schema, i.e. it leads to an extension of the basic inference relation $\vdash$ defined by the underlying KS. The connection between them can be expressed by means of the *CWA closure*:

$$\mathit{CWA}(Y) \;:=\; \mathsf{Upd}(Y, \{{\sim} p(c) : p \in \mathit{CWRel} \;\; \& \;\; c \in \mathit{Const} \;\; \& \;\; \mathrm{KB} \vdash_\Sigma -p(c)\})$$

Obviously, we have $Y \vdash_\Sigma F$ iff $\mathit{CWA}(Y) \vdash F$.

Example 6 (The CWA in Factbases) *While we cannot assume the CWA for empirical predicates like* smoker*, we should also not assume it for the relation* married *in some local KB since people may get married all over the world, and thus* married *will not be totaly represented in a local KB. We may assume the CWA, however, for a predicate like* resident*, simply because all residents of a city are registered in the local KB of that city. Thus, from the factbase*

$$X_3 = \{{\sim} s(S),\, m(P,L),\, r(P),\, r(S)\}$$

over a schema with CWRel $= \{r\}$*, we may infer that Peter is married with a non-resident,*

$$\mathsf{Ans}(\mathit{CWA}(X_3), m(x,y) \wedge {\sim} r(y)) = \{\langle P, L\rangle\}$$

but neither that he is married with a nonsmoker,

$$\mathsf{Ans}(\mathit{CWA}(X_3), m(x,y) \wedge {\sim} s(y)) = \emptyset$$

nor that he is not married with Susan, $\mathit{CWA}(X_3) \not\vdash {\sim} m(P,S)$.

Definition 28 (Schema-Based Inference) *A query F is inferable from a disjunctive factbase $Y : \Sigma$ if it can be derived from the closure of Y with respect to* ExRel *and* CWRel*. Formally,*

$$Y \vdash_\Sigma F :\Longleftrightarrow \mathit{Exact}(Y) \vdash F$$

where

$$\mathit{Exact}(Y) = \mathsf{Upd}(\mathit{CWA}(Y),\; \{p(c) \vee {\sim} p(c) : p \in \mathit{ExRel} - \mathit{CWRel} \;\; \& \;\; c \in \mathit{Const}\})$$

[19] The Closed-World Assumption, in a less general form, was originally proposed in [Rei78]. Notice that our form of the CWA relates explicit with default-implicit falsity, i.e. strong with weak negation: an atomic sentence formed with a totaly represented predicate is (explicitly) false if it is false by default, i.e. its strong negation holds if its weak negation does.

Notice that in KBs of definite knowledge systems, like factbases, it is not possible to declare exact predicates not subject to the CWA. Therefore, in definite knowledge systems, $ExRel = CWRel$.

Observation 17 *For a schema $\Sigma = \langle\, Rel,\ ExRel,\ CWRel,\ IC\,\rangle$, and a knowledge base Y over Σ, it holds that*

1. *for any exact predicate $p \in ExRel$, and any constant c from its domain, the resp. instance of the* tertium non datur *holds: $Y \vdash p(c) \vee {\sim}p(c)$;*

2. *if $q \in CWRel$, then Y does not contain any indefinite information on q, i.e. $Y \vdash q(c)$, or $Y \vdash {\sim}q(c)$.*

Inputs leading to the violation of integrity constraints have to be rejected. The update operation for disjunctive factbases has to be modified accordingly.

Definition 29 (Schema-Based Update)

$$\mathsf{Upd}_*^{\Sigma}(Y, F) = \{X \in \mathsf{Upd}_*(Y, F) : X \vdash IC\}$$

where $* = B, mi, ex$.

6 Reasoning with Three Kinds of Predicates

Only certain exact predicates can be totally represented in a KB. Totally represented exact predicates are subject to the CWA. For example, the local KB of some city knows all residents of the city, i.e. the CWA holds for *resident*, but it does not have complete information of every resident whether (s)he is married or not because (s)he might have married in another city and this information is not present. Consequently, the CWA does not apply to *married* in this KB.

The CWA helps to reduce disjunctive complexity which is exponential in the number of exact non-CWA predicates: if n is the number of unknown ground atoms which can be formed by means of predicates declared as exact but not subject to the CWA by a VKB, then the VKB contains 2^n possible situation descriptions.

We illustrate these distinctions with an example. Let m, r, s, l denote the predicates *married*, *resident*, *smoker* and *is_looking_at*, and let M, P, S stand for the individuals *Mary*, *Peter* and *Susan*. Let

$$Y = \{\{m(M),\ r(M),\ s(M),\ {\sim}m(S),\ {\sim}s(S),\ l(M,P),\ l(P,S)\}\}$$

be a disjunctive factbase over the schema $\Sigma = \langle\{m,r,s,l\},\ \{m,r\},\ \{r\},\ \emptyset\rangle$. The interesting queries we can ask Y and the resp. answers are:

1. Does a married person look at an unmarried one ? Yes, but Y does not know who, either Mary at Peter, or Peter at Susan:

$$Y \vdash_{\Sigma} \exists x, y : l(x,y) \wedge m(x) \wedge {\sim}m(y)$$
$$\mathsf{Ans}(Y,\ l(x,y) \wedge m(x) \wedge {\sim}m(y)) = \{\{\langle M,P\rangle,\ \langle P,S\rangle\}\}$$

2. Does a resident look at a non-resident ? Yes, Mary at Peter.

$$\mathsf{Ans}(Y,\, l(x,y) \wedge r(x) \wedge \sim r(y)) = \{\{\langle M, P\rangle\}\}$$

since $Y \vdash_\Sigma \sim r(P)$ if $Y \vdash -r(P)$.

3. Does a smoker look at a nonsmoker ? No. Y is completely ignorant about Peter being a smoker or not: neither is he a smoker, nor is he a nonsmoker, nor is he a smoker or nonsmoker (he might be neither):

$$\mathsf{Ans}(Y,\, l(x,y) \wedge s(x) \wedge \sim s(y)) = \emptyset$$

Notice that the explicit tertium-non-datur completion of Y wrt *ExRel* would yield $2^3 = 8$ possible situation descriptions, which are reduced to $2^1 = 2$ by the CWA declaration of *resident*.[20]

7 A General Construction of Disjunctive Knowledge Systems

We now generalize the previous discussion of specific disjunctive knowledge systems. Let

$$\boldsymbol{K} \;=\; \langle\, 0,\, \leq,\, L_{\mathrm{KB}},\, \vdash,\, L_{\mathrm{Query}},\, \mathsf{Upd},\, L_{\mathrm{Input}},\, \mathrm{Inc},\, L_{\mathrm{Unit}} \,\rangle$$

We shall inductively define the extension of $\boldsymbol{K}$ to a disjunctive knowledge system $V\boldsymbol{K}$,[21] with

$$V\boldsymbol{K} \;=\; \langle\, 0_\vee,\, \leq_\vee,\, L^\vee_{\mathrm{KB}},\, \vdash_\vee,\, L^\vee_{\mathrm{Query}},\, \mathsf{Upd}^\vee_{mi},\, L^\vee_{\mathrm{Input}},\, \mathrm{Inc}_\vee,\, L^\vee_{\mathrm{Unit}} \,\rangle$$

A disjunctive knowledge base $Y \in L^\vee_{\mathrm{KB}}$ consists of a set of KBs, each one describing a possible situation, i.e. $L^\vee_{\mathrm{KB}} \subseteq 2^{L_{\mathrm{KB}}}$, and the elements of Y represent different epistemic alternatives. The knowledge ordering is defined as

$$Y \leq_\vee Y' :\Longleftrightarrow \forall X' \in Y'\, \exists X \in Y : X \leq X'$$

implying that $0_\vee := \{0\}$. Only paracanonical elements of $2^{L_{\mathrm{KB}}}$ will be accepted as KBs, i.e. $L^\vee_{\mathrm{KB}} = \mathrm{PCan}(2^{L_{\mathrm{KB}}})$.

A query formula $F \in L^\vee_{\mathrm{Query}}$ can be inferred from Y if it can be inferred from every possible situation description $X \in Y$:

$$Y \vdash_\vee F \quad :\Longleftrightarrow \quad \text{for all } X \in Y : X \vdash F,$$

i.e. $L^\vee_{\mathrm{Query}} = L_{\mathrm{Query}}$.

Inputs to a disjunctive knowledge base Y are processed in the following way:

$$\begin{aligned}
\mathsf{Upd}^\vee_B(Y,u) &= \{\mathsf{Upd}(X,u) : X \in Y\} \text{ for } u \in L_{\mathrm{Unit}} \\
\mathsf{Upd}^\vee_B(Y, F \wedge G) &= \mathsf{Upd}^\vee_B(\mathsf{Upd}^\vee_B(Y,F), G) \\
\mathsf{Upd}^\vee_B(Y, F \vee G) &= \mathrm{PMin}(\mathsf{Upd}^\vee_B(Y,F) \cup \mathsf{Upd}^\vee_B(Y,G) \cup \mathsf{Upd}^\vee_B(Y, F \wedge G))
\end{aligned}$$

[20] Without the CWA there are 3 tertium non datur disjunctions formed with $m(P)$, $r(S)$, and $r(P)$, while with the CWA there is only one such disjunction: $m(P) \vee \sim m(P)$.

[21] V stands for disjunction.

All other cases of compound formulas are treated by DeMorgan-style rewriting. We define two further update operations:

$$
\begin{aligned}
\mathrm{Cons}(Y) &:= \{X \in Y \mid \mathrm{Inc}(X) = \emptyset\} \\
\mathrm{MInc}(Y) &:= \{X \in Y \mid \neg\exists X' \in Y : \mathrm{Inc}(X') \subset \mathrm{Inc}(X)\} \\
\mathsf{Upd}^{\vee}_{ex}(Y, F) &:= \mathrm{Cons}(\mathsf{Upd}^{\vee}_{B}(Y, F)) \\
\mathsf{Upd}^{\vee}_{mi}(Y, F) &:= \mathrm{MInc}(\mathsf{Upd}^{\vee}_{B}(Y, F))
\end{aligned}
$$

The elementary pieces of information in disjunctive KBs are disjunctions of information units $u_1 \vee \ldots \vee u_m$. The set of all such disjunctions is denoted by $L^{\vee}_{\mathrm{Unit}}$. The inconsistency operation $\mathrm{Inc}_{\vee}$ collects all contradictory disjunctions of elements from L_{Unit}:

$$
\mathrm{Inc}_{\vee}(Y) := \{\bigvee U : U \in \mathrm{Min}(\{U' \subseteq L_{\mathrm{Unit}} \mid \forall X \in Y \exists u \in U' : u \in \mathrm{Inc}(X)\})\}
$$

We obtain the following systems:

$$
V_B\boldsymbol{K} \;=\; \langle \{0\},\, \leq_{\vee},\, \mathrm{PCan}(2^{L_{\mathrm{KB}}}),\, \vdash_{\vee},\, L_{\mathrm{Query}},\, \mathsf{Upd}^{\vee}_{B},\, L(\sim, \wedge, \vee),\, \mathrm{Inc}_{\vee},\, L^{\vee}_{\mathrm{Unit}} \rangle
$$

and $V_{ex}\boldsymbol{K}$ and $V_{mi}\boldsymbol{K}$ as before.

8 A Case Study: Disjunctive Deductive Factbases

Using the above construction we can form $V_*D\boldsymbol{F}$, $* = B, mi, ex$, the system of *disjunctive deductive factbases (DDFB)*. A DDFB is a paracanonical set of deductive factbases.

Example 7 *$Y_7 = \{\langle X_1, R_1\rangle, \ldots, \langle X_7, R_7\rangle\}$, where*

i	X_i	R_i
1	$\{p\}$	$\{r \leftarrow -(p \wedge q)\}$
2	$\{q\}$	$\{\sim p \leftarrow -(p \wedge q)\}$
3	$\{q\}$	$\{r \leftarrow -(p \wedge q)\}$
4	$\{q\}$	$\{\sim p \leftarrow -(p \wedge q),\, r \leftarrow -(p \wedge q)\}$
5	$\{p, q\}$	$\{\sim p \leftarrow -(p \wedge q)\}$
6	$\{p, q\}$	$\{r \leftarrow -(p \wedge q)\}$
7	$\{p, q\}$	$\{\sim p \leftarrow -(p \wedge q),\, r \leftarrow -(p \wedge q)\}$

is a DDFB. In $V_{ex}D\boldsymbol{F}$, we obtain e.g. $Y_7 \vdash -p \vee r \vee q$.

Observation 18 *An extended disjunctive logic program (EDLP) can be transformed into a DDFB.*

Proof: An EDLP consists of rules of the form $H \leftarrow B$, where $H \subseteq \mathrm{Lit}$, and $B \subseteq \mathrm{XLit}$. Starting from the empty DDFB, $Y_0 = \langle \emptyset, \emptyset \rangle$, we successively 'compile' all the EDLP rules $H_i \leftarrow B_i$, $i = 1, \ldots, m$, into the resp. DDFB;

$$
Y_i = \mathrm{PMin}(\{\Pi \cup \{\bigvee G \leftarrow B_i\} : \Pi \in Y_{i-1} \;\&\; G \subseteq H_i\}) \quad \Box
$$

This is similar to the 'split database' procedure of [Sak89]. Notice, however, that by our PMin-normalization, we do not get the problem of non-cumulativity whereas adding a disjunctive lemma may change the set of 'possible models'.

Example 7 (continued) *The DDFB from the previous example is the result of compiling the EDLP* $\Delta_7 = \{p \vee q,\ {\sim}p \vee r \leftarrow -(p \wedge q)\}$.

Conjecture 2 *The answer set semantics [GL91] of an extended disjunctive logic program* Δ *can be captured by discarding all nonminimal elements of the corresponding DDFB* Y_Δ*: every answer set of* Δ *corresponds to a stable closure of some element of* $\mathrm{Min}(Y_\Delta)$.

For instance, the stable closures of $\mathrm{Min}(Y_7)$ are $\{p, r\}$, $\{q, {\sim}p\}$, and $\{q, r\}$ which are exactly the answer sets of Δ_7.

9 Conclusion

The system of disjunctive factbases which captures the inference relation based on paraminimal models is the paradigm for disjunctive knowledge systems. We have shown that there is an underlying general construction of disjunctive knowledge systems which can be applied to all kinds of definite base systems.

References

[AN84] A. Almukdad and D. Nelson: Constructible Falsity and Inexact Predicates, *JSL* **49**:1 (1984), 231–233.

[Bel77] N.D. Belnap: A Useful Four-valued Logic, in G. Epstein and J.M. Dunn (Eds.), *Modern Uses of Many-valued Logic*, Reidel 1977, 8–37.

[Gab85] D. Gabbay: Theoretical Foundations for Nonmonotonic Reasoning in Expert Systems, in K.R. Apt (Ed.), *Proc. NATO Advanced Study Institute on Logics and Models of Concurrent Systems*, Springer Verlag, 1985, 439–457.

[Gär88] P. Gärdenfors: *Knowledge in Flux*, MIT Press, Cambridge, 1988.

[vGT91] A. van Gelder and R.W. Topor: Safety and Translation of Relational Calculus Queries, *ACM Transactions on Database Systems* **16**:2 (1991), 235–278.

[GL90] M. Gelfond and V. Lifschitz: Logic Programs with Classical Negation, *Proc. ICLP 1990*, MIT Press, 1990.

[GL91] M. Gelfond and V. Lifschitz: Classical Negation in Logic Programs and Disjunctive Databases, *J. New Generation Computing* **9** (1991), 365–385.

[HJW96] H. Herre, J. Jaspars and G. Wagner: Partial Logics with Two Kinds of Negation as a Foundation for Knowledge-Based Reasoning, in D. Gabbay and H. Wansing (Eds.), *What is Negation?*, Kluwer, 1996.

[KM93] Y. Kaluzkuy and A.Y. Muravetsky: A Knowledge Representation Based on the Belnap's Four-Valued Logic, *J. of Applied Non-Classical Logics* **3**:2 (1993), 189–203.

[Kör66] S. Körner: *Experience and Theory*, Kegan Paul, London, 1966.

[Lev84] H.J. Levesque: Foundations of a Functional Approach to Knowledge Representation, *AI* **23**:2, 1984, 155-212.

[Lev86] H.J. Levesque: Making Believers out of Computers, *AI* **30** (1986), 81-107.

[LMR92] J. Lobo, J. Minker and A. Rajasekar: *Foundations of Disjunctive Logic Programming*, MIT Press, 1992.

[Min82] J. Minker: On Indefinite Databases and the Closed-World Assumption, *Proc. of CADE-82*, 1982, 292–308.

[PW90] D. Pearce and G. Wagner: Reasoning with Negative Information I – Strong Negation in Logic Programs, LWI Technical Report, Freie Universität Berlin, 1989. Also in L. Haaparanta, M. Kusch and I. Niiniluoto (Eds.), *Language, Knowledge, and Intentionality*, Acta Philosophica Fennica **49**, 1990.

[PA92] L.M. Pereira and J.J. Alferes: Wellfounded Semantics for Logic Programs with Explicit Negation, *Proc. ECAI'92*, Wiley, 1992.

[Plo76] G.D. Plotkin: A Powerdomain Construction, *SIAM J. Comp.* **5**:3 (1976), 452–487.

[Pri89] G. Priest: Reasoning about Truth, *AI* **39** (1989), 231–244.

[Rei78] R. Reiter: On Closed-World Databases, in J. Minker and H. Gallaire (Eds.): *Logic and Databases*, Plenum Press, 1978.

[RT88] K.A. Ross and R.W. Topor: Inferring Negative Information from Disjunctive Databases, *J. Automated Reasoning* **4**:2 (1988), 397–424.

[Sak89] C. Sakama: Possible Model Semantics for Disjunctive Databases, *Proc. 1st Int. Conf. on Deductive and Object-Oriented Databases (DOOD'89)*, North-Holland, 1989.

[Smy78] M. Smyth: Powerdomains, *J. Comp. Syst. Sci.* **16**:1 (1978), 23–36.

[Urb90] I. Urbas: Paraconsistency, *Studies in Soviet Thought* **39** (1990), Kluwer, 343–354.

[Wag91] G. Wagner: Logic Programming with Strong Negation and Inexact Predicates, *J. Logic and Computation* **1**:6 (1991), 835–859.

[Wag94] G. Wagner: *Vivid Logic – Knowledged-Based Reasoning with Two Kinds of Negation*, Springer Lecture Notes in Artificial Intelligence **764**, 1994.

[Wag94b] G. Wagner: *Disjunctive, Deductive and Active Knowledge Bases*, LWI Report 22/1994, Freie Universität Berlin; available as `ddakbs.dvi.gz`, resp. `ddakbs.ps.gz`, via ftp from `ftp.inf.fu-berlin.de` in the directory `/pub/reports/wagner` (notice that the file has to be 'gunzip'ed).