A Framework for Knowledge-Based Systems

Alexei Yu. Muravitsky
Department of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012
Internet: `mrvtskya@acf4.nyu.edu`

Abstract

The paper continues the theme of [KM 93]. We differentiate between our approach to knowledge representation and that of others by expressing the following Working Hypothesis: *Knowledge is a data type, and knowledge revision is accomplished by continuous operations on it, which are coordinated with its effective basis.* Staying in the limits of Belnap's paradigm of the admittance of contradictory information into the computer's memory, our purpose in this paper is to reduce as much as possible all the computable processes needed for modifing the current state of the computer's knowledge and describe conditions for possible maneuvering. In particular, we solve some problems of decidability concerning operations on the minimal states, which are regarded as natural knowledge transformers. We show, also, how to express those operations in lattice theory terms that leads to the simplification of their computation on the lattice of minimal states. The problem of backtracking in the presented context is considered as well.

1 Introduction

The choice of one or another Knowledge Representation (KR) strategy is determined by the choice of some underlying working hypothesis. In the entire KR field, without being very specific, one assumes that KR systems process information, qualifying as knowledge or beliefs, using "mechanical operations over symbolic representation"; and *de facto* "an AI system can be described as knowledge-based not because knowledge is required to build it, nor just that it behaves as if it had knowledge, but rather that its architecture includes explicit KBs that are more or less direct symbolic encoding of the knowledge of the system" (cf. [Lev 86]). Moreover, as to the knowledge represented in the system, according to the Knowledge Representation Hypothesis: "Any mechanically embodied intelligent process will comprise structural ingredients that a) we as external observers naturally take to represent a propositional account of the knowledge that the overall process exibits, and b) independent of such external semantical attribution, play a formal but causal and essential role in engendering the behavior that manifests that knowledge" (cf. [Smi 82]; see comments in [Isr 93, Lev 86]). This view appears to indicate, first, that the knowledge represented in the intelligent system must be thought as propositions in a formal language and, second, the process of running the knowledge embodied is independent of that declarative form of knowledge.

It is clear that the former just means that we limit ourselves to the linguistic interpretation of the knowledge. Such a limitation is for example justified by the fact that we would like to consider the knowledge with which we deal in the light of some theory of the truth [1]. What does, however, stand behind the latter? How have, in particular, to understand that independence above? It seems that both the knowledge represented in the intelligent system and the knowledge revision operations and related strategies should be considered as a realization of more general conceptions that would express the conceptual nature of the computer-oriented understanding of the knowledge. Let us briefly look at the previous work in the KR field.

As to the problem of the knowledge representation itself, a main "goal is

---

[1] G. Frege was probably the first to connect the notion of knowledge with the theory of truth via the notion of the proposition. (Cf. G. Frege, Der Gedanke: eine logische Untersuchung, *Logische Untersuchungen*, Vandenhoeck & Ruprecht in Göttingen, 1966, ss. 30–53.)

to design formal languages for expressing facts, say, of history, engineering, science, commonsense experience, so that computers can operate with these facts" [Lif 93]. As to operations applied to the knowledge represented in the system, sometimes interpreted as belief revision, it seems there is no general framework at all, except for mechanical manipulations of the symbolic representations mentioned above. The logic-based approach in the line proposed by John McCarthy in the 'Advice Taker' [McC 58] was the begining of the investigation and use of the situational logics that are "occasionally referenced as an implementational technique" (cf. [DL 80]). In the 'Advice Taker', for example, process operations only involve "printing sentences, moving sentences on lists, and reinitiating the basic deduction process on these lists" (cf. [McC 58]). Also, in numerous implementation-oriented approaches we do not find any foundation in the form of mathematical concepts for representing and revising knowledge implemented in the computer system (cf. [DL 80]). Of course we are by no means denigrating the important and valuable developments in the field; we are simply drawing attention to what we perceive to be the lack of meaningful mathematical foundations.

However, if we were to ask what knowledge is from the computer-oriented point of view, we might perhaps receive the answer: It is made up of certain kinds of data types. An axiomatic approach to the notion of data type was undertaken by Dana Scott in [Sco 71]. According to his approach, a data type is a lattice $\mathcal{E}$ of values of data of that type acceptable to the computer. Furthermore, the ordering relation of the lattice $\mathcal{E}$ is, roughly speaking, recursive, and $\mathcal{E}$ itself is a sublattice of a complete (and even continuous in the sense of [GHKLMS 80]) lattice $\mathcal{L}$. Moreover, $\mathcal{E}$ is an effective basis of $\mathcal{L}$, that is, every element in $\mathcal{L}$ is a limit of a sequence of elements in $\mathcal{E}$ with respect to the Scott topology on $\mathcal{L}$ (cf. [GHKLMS 80]).

So far, we have said nothing specific about knowledge representation and nothing at all about knowledge revision. As to the latter, we will only admit continuous operations on data types, which are coordinated with its basis in the sense that the result of the operation from points of the basis belongs to the basis (*Scott Principle*; cf. [KM 93]). Now, let us accept the following
Working Hypothesis [2]: *Knowledge is a data type, and knowledge revision*

---

[2] Accepting this Working Hypothesis we would like to remind the reader that the history of science teaches us to prefer the abstract to the empirical approach. As to the physical science, today we know that it is the abstract notion of geometrizable space, rather than rational representations imbued with Aristotle's empiricism (though more suitable for the

is accomplished by continuous operations on it, which are coordinated with its effective basis.

One well-known way of treatment with entities "computable in advance" is to make them as being points in a specified continuum [3]. Such a continuum for the computer-represented knowledge, which was able to tolerate any information (not excluding contradictions) as the computer's input, was introduced in [KM 93] as the lattice AGE. In representing knowledge we proceeded from interpretations, i.e. assignments, or *setups*, of our logical language rather than from the symbolic expressions themselves. And, as Belnap clearly showed, it is in general not enough to maintain one setup for processing information knowledge (cf. [Bel 75, Bel 76]). However, Belnap's basic definition of epistemic state meant as some nonempty set of setups played only an auxiliary role in our discussion in [KM 93]. Instead as our main notions we used those of minimal and generalized (epistemic) states. To include all computer-tractable elements of the knowledge represented we formed the lattice AFE of all the minimal states. Then, we sought to find a topological space, where the needed continuous operations would work. So, we arrived at the lattice AGE of all the generalized states. In some sense AFE is prior to AGE being a field of feasable knowledge of the computer; and we needed to draw something like AGE in order to define continuous operations. It remains to add that AFE is an effective basis of AGE. Thus, though elements in AGE are in general transfinite, we are able to approach any of them by elements in AFE with any sharpness allowed in AGE as a topological space.

We consider below some problems which the designer, who would attempt to underly the presented approach, can be faced with. Our purpose is to reduce as much as possible all the computable processes needed for modifying the current state of the computer's knowledge and describe conditions for possible maneuvering. We will briefly concern the backtracking problem as

---

immediate observation of facts) that Galilei's genius put into the foundations of the new physics "and indeed, already as pure theory, which sets forth nature as representing itself via relationship between forces computable in advance" (Heidegger). (Cf. A. Koyré, Galilée et Platon, in: A. Koyré, *Les études d'histoire de la pensée scientifique*, Paris, Presses Universitaires de France, 1966, pp. 147–175.)

[3] There is a definite correlation between Scott's approach and the underlying idea in the construction of the number-theoretic continuum. Comp. the Definition 2.3 in [Sco 72] and §7 in [R. Dedekind, Continuity and irrational numbers, in: R. Dedekind, *Essays on the theory of numbers*, New York, Dover Publications, 1963]. Also, see the Exercise III-4.19 in [GHKLMS 80].

well.

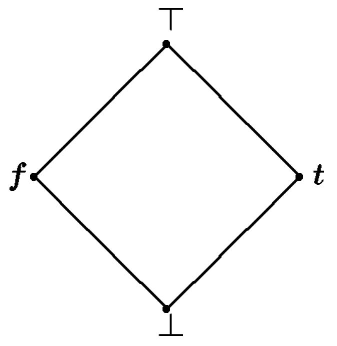
2 Basic Definitions

It is a matter of indifference to the computer which kind of truth theory we have in mind, nor does it distinguish knowledge from belief when working with the information flow, that is, in a changeable information environment. When dealing with information knowledge we do not do this either; moreover, the computer should be prepared to tolerate inconsistency. Although *ontological* sensing of contradictions (in the terms of possible-world semantics) only was developed in [RB 79], Nuel Belnap had previously proposed a computer-oriented ontology for maintaining contradictory knowledge in two papers [Bel 75, Bel 76].

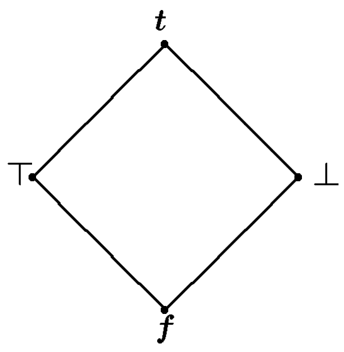
Regarding information knowledge propositionally and not distinguishing the status of valid (or known) and plausible (or believed) sentence, we accept (following [Bel 75, Bel 76]) (at least) a four-valued truth theory with the set $\Im$ of truth values: $\boldsymbol{f}$ (falsehood), $\boldsymbol{t}$ (truth), $\bot$ (unknown), $\top$ (both truth and falsehood). We limit ourselves here to a simple propositional language with the set Var $(= \{p_1, p_2, \ldots\})$ and connectives: $\wedge$, $\vee$, and $\neg$. We denote arbitrary formulas of the language via A, B, C... (with or without indices) and define (also following [Bel 75, Bel 76]) the truth value of a formula in terms of the values of its propositional variables according to the Belnap's "logical lattice" $\mathsf{L4}$ $(=< \Im, \wedge, \vee >)$ partially ordered by the relation $\leq$ (see Fig. 1). The set of the variables of A we denote via $V(A)$. For representing knowledge we need another Belnap's "approximate lattice" $\mathsf{A4}$ $(=< \Im, \sqcap, \sqcup >)$ partially ordered by the relation $\sqsubseteq$ (see Fig. 1).

Figure 1: Lattices $\mathsf{A4}$ and $\mathsf{L4}$.

A4



L4



A *setup* is a mapping $s : \mathsf{Var} \to \Im$ that is extended to all the formulas with respect to the following definition:

$$s(A \wedge B) = s(A) \wedge s(B),$$

$$s(A \vee B) = s(A) \vee s(B),$$

$$s(\neg A) = \neg s(A),$$

where the operation $\neg$ on $\Im$ is defined by means of the following conditions: $\neg \boldsymbol{t} = \boldsymbol{f}$, $\neg \boldsymbol{f} = \boldsymbol{t}$ and $\neg\tau = \tau$, if $\tau \in \{\bot, \top\}$.

All the setups form the lattice AS with the order:

$$s \leq s_1 \text{ if and only if } s(p) \sqsubseteq s_1(p) \text{ for every } p \in \mathsf{Var}.$$

A setup s is *finite* if the set $\{p \,|\, s(p) \neq \bot\}$ is finite. We denote the last set via $V(s)$. An *epistemic state* is a nonempty set of setups. For any formula A, for example, we will consider the epistemic states $Tset(A)$ and $Fset(A)$ defined as follows:

$$Tset(A) \overset{\text{def}}{=} \{s \,|\, s \in \mathsf{AS}, \boldsymbol{t} \sqsubseteq s(A), V(s) \subseteq V(A)\},$$

$$Fset(A) \overset{\text{def}}{=} \{s \,|\, s \in \mathsf{AS}, \boldsymbol{f} \sqsubseteq s(A), V(s) \subseteq V(A)\},$$

where $V(A)$ are all the variables of the formula A.

A finite set of finite setups is called a *finite (epistemic) state.* For example, $Tset(A)$ and $Fset(A)$ are finite states for any formula A. Because of Descending Chain Condition, the set $m(\varepsilon)$ of all the minimal elements in a finite stae ε is a nonempty set. Therefore, $m(\varepsilon)$ is a finite state whenever ε is; moreover, $m(m(e)) = m(\varepsilon)$. Each $m(\varepsilon)$, where the state ε is finite, we call *minimal (epistemic) state* — nonempty finite set of incomparable finite setups. All the minimal states form the lattice AFE with the order defined as follows:

$$\varepsilon' \leq \varepsilon \text{ if and only if for any } s \in \varepsilon \text{ there is } s' \in \varepsilon' \text{ such that } s' \leq s$$

(cf. [KM 93]). That relation makes sense for all the set of the epistemic states, however, as a relation of the quasi-order.

Following [Bel 75, Bel 76], we introduce an *assignment* of formula A in epistemic state ε as follows:

$$\varepsilon(A) \overset{\text{def}}{=} \sqcap\{s(A)\,|s \in \varepsilon\}.$$

Then, we define:

$$\varepsilon \equiv \varepsilon' \text{ if and only if } \varepsilon(A) = \varepsilon'(A) \text{ for any formula } A.$$

A *generalized (epistemic) state* $\overline{\varepsilon}$ is a set $\{\varepsilon'\,|\varepsilon \equiv \varepsilon'\}$. All the generalized states as a whole form the lattice AGE with the order:

$$\overline{\varepsilon} \leq \overline{\varepsilon'} \text{ if and only if } \varepsilon(A) \sqsubseteq \varepsilon'(A) \text{ for every formula } A.$$

The mapping $\varepsilon \rightarrow \overline{\varepsilon}$ is an embedding from AFE into AGE. (See [KM 93] for details.)

We will work sometimes with restrictions of setups and epistemic states. Let V be a set of variables. *V-down-restriction* of setup s is the setup $s^{V\perp}$ defined as follows:

$$s^{V\perp} \overset{\text{def}}{=} \begin{cases} s(p) & \text{for } p \in V \\ \perp & \text{for } p \notin V. \end{cases}$$

V-down-restriction of epistemic state ε is the state $\varepsilon^{V\perp}$ defined as follows:

$$\varepsilon^{V\perp} \overset{\text{def}}{=} \left\{s^{V\perp}\,|s \in \varepsilon\right\}.$$

Also, we define: $\overline{\varepsilon}^{V\perp} \overset{\text{def}}{=} \overline{\varepsilon^{V\perp}}$ (cf. [KM 93]).

3 Main Theorems

In [KM 93] we considered three kinds of operations as modifying knowledge in agreement with the above Working Hypothesis. They are A^+, A^- and $[A \rightarrow B]$ for every formula A and B. It was proved there that $A^-(\overline{\varepsilon}) = \neg A^+(\overline{\varepsilon})$ for every generalized state $\overline{\varepsilon}$. Thus, we only need consider operations of the form A^+ (denoting them as $[A]$) and those corresponding to *conditionals* (or *rules*) $A \rightarrow B$ (see [KM 93] for definitions).

The introduction of the operations $[A]$ and $[A \rightarrow B]$ is motivated as follows. $[A]$ corresponds to the situation when the computer receives a message thet the statement expressed by the formula A is true. In the case that the message states that A is false we use $[\neg A]$. The operation $[A \rightarrow B]$ in turn corresponds to the correction of the current information state with respect to the conditional **if** A **then** B. Now, let us note some properties in addition to those mentioned in [KM 93].

We think about $[A]$- and $[A \rightarrow B]$-operations as being *elementary actions* admitted in our intelligent system. Now the question arises: What kind of action is a result of performing a finite sequence of elementary actions? If we limit ourselves to $[A]$-operations then the Theorem 1 says that the result is an $[A]$-action.

Theorem 1 (on Actions Blocking) *For any formulas A and B and every generalized state $\overline{\varepsilon}$*

$$[A]([B](\overline{\varepsilon}))=[A \wedge B](\overline{\varepsilon}).$$

Proof (Deferred to the Appendix).

It is natural that different actions could affect the same whenever applied to any minimal (or even generalized) state. We solve the decidabilty problem for the quality of the $[A]$-actions. In what follows, E_{fde} means the Anderson-Belnap's calculus of first degree entailment from [AB 75], and $\vdash_{E_{fde}} A \leftrightarrow B$ means that both $\vdash_{E_{fde}} A \rightarrow B$ and $\vdash_{E_{fde}} B \rightarrow A$ hold.

Theorem 2 *For any formulas A and B, $[A](\overline{\varepsilon}) = [B](\overline{\varepsilon})$ for every $\overline{\varepsilon}$ if and only if $\vdash_{E_{fde}} A \leftrightarrow B$.*

Proof (Deferred to the Appendix).

Corollary 2.1 *Two elementary actions* $[A]$ *and* $[B]$ *are identical on* AGE *if and only if they are identical on* AFE.

Proof follows immediately from the Proposition 2 and Theorem 1 in [Mur 93] and the Theorem 2.

The local version of the problem solved in the Theorem 2 is the problem, whether the correlation $[A](\varepsilon) = [B](\varepsilon)$ is effectively decidable. We solve it in the

Theorem 3 *Given minimal state* ε *and formulas* A *and* B*, the correlation*

$$[A](\varepsilon) = [B](\varepsilon)$$

is recursively decidable.

Proof (Deferred to the Appendix).

Though the definition of the operation of the sort of $[A \rightarrow B]$ presented in [KM 93] look too complicated to use for computing, we show below how to determine that operation in lattice terms in AFE; we recall that in regard of $[A]$-operations as well.

For any epistemic state ε and formula A let

$$\varepsilon_A^- \overset{\text{def}}{=} \{s \,|\, s \in \varepsilon, \boldsymbol{t} \not\sqsubseteq s(A)\} \text{ and } \varepsilon_{\overline{A}} \overset{\text{def}}{=} \begin{cases} \varepsilon_A^- & \text{if } \varepsilon_A^- \neq \emptyset \\ \text{the unit } \mathbf{1} \text{ in AGE} & \text{otherwise.} \end{cases}$$

Note one useful property:

$$\varepsilon_1 \leq \varepsilon_2 \Longrightarrow (\varepsilon_1)_{\overline{A}} \leq (\varepsilon_2)_{\overline{A}}. \qquad (1)$$

Theorem 4 (on Unification) *For every minimal state* ε *and any formulas* A *and* B*,*

$$[A](\varepsilon) = \varepsilon \sqcup m(Tset(A)); \qquad (2)$$

$$[A \rightarrow B](\varepsilon) = \varepsilon \sqcup (\varepsilon_{\overline{A}} \sqcap m(Tset(B)) = \varepsilon_{\overline{A}} \sqcap [B](\varepsilon).$$

Proof. The equation (2) is proved in [KM 93]. The rest is deferred to the Appendix.

We would want the order in which rules are applied not to matter as is the case for $[A]$-operations (cf. Theorem 1). Unfortunately, it does not hold for $[A \rightarrow B]$-operations in general as is shown in the following

Example 1 *Let* $s_1(p_1) = \boldsymbol{t}$ *and* $s_1(p) = \bot$ *for* $p \neq p_1$, $s_2(p_2) = \boldsymbol{t}$ *and* $s_2(p) = \bot$ *for* $p \neq p_2$, *and* $\varepsilon = \{s_1, s_2\}$. *Then*

$$[p_1 \rightarrow p_2]([p_2 \rightarrow p_3](\varepsilon)) \neq [p_2 \rightarrow p_3]([p_1 \rightarrow p_2](\varepsilon)).$$

Proof is easy to check, using the Theorem 4, and left to the reader.

However, provided some conditions, the commutativity of performences of $[A \rightarrow B]$-actions already holds. Note that these conditions are effectively decidable.

Theorem 5 *For any minimal state* ε *such that* $m(Tset\,(B)) \leq \varepsilon_{\overline{A_1}}$, $m(Tset\,(B_1)) \leq \varepsilon_{\overline{A}}$ *and* $\varepsilon^-_A = \emptyset$ *if and only if* $\varepsilon^-_{A_1} = \emptyset$, *the following equation*

$$[A_1 \rightarrow B_1]([A \rightarrow B](\varepsilon)) = [A \rightarrow B]([A_1 \rightarrow B_1](\varepsilon))$$

holds.

Proof (Deferred to the Appendix).

4 Backtracking Problem

The problem of backtracking in the context presented is as follows. Suppose the computer begins in a minimal state ε, and that a number of sequential operations move the computer to state ε_1. Notice that the composition of continuous operations is a continuous operation. Then we have $\varepsilon_1 = F(\varepsilon)$, where F is Scott-continuous. We may call F a *path.* It is possible that at this point the computer's user decides to defeat (or modify) the information about A, which the computer currently has, for example, because of its excessive uncertainty (or over-determination, as Dana Scott says). At this current instant of time the computer certainly knows ε_1. What else need it know in order to move to the state of corrected information about A? And how should it perform such a manoeuvre? Notice the computer must not lose any information, except about A(comp. [Gin 93] for a backtracking strategy in problem solving).

Let us limit ourselves to $[A]$-operations [4]. Then, let π_i mean $m(Tset(A_i))$. We call it an *elementary steps.* Now, every path can be described by some

---

[4] This limitation has been proved to seem as recent investigation has shown. In fact, we may admit all the class of ***CAC***-operations to be considered (cf. [Mur 94]).

list $< \pi_1, \ldots, \pi_n >$. It is clear that the result $< \sigma(\pi_1), \ldots, \sigma(\pi_n) >$ of the transposition σ of the list $< \pi_1, \ldots, \pi_n >$ leads to the equation:

$$< \pi_1, \ldots, \pi_n > (\varepsilon) =< \sigma(\pi_1), \ldots, \sigma(\pi_n) > (\varepsilon)$$

for any minimal state ε (cf. (2) in the Theorem 4). Thus, though any path from a state ε is covered by a finite sequence, say $< \pi_1, \ldots, \pi_n >$, of elementary steps, we can associate with the path the corresponding set $\{\pi_1 \ldots, \pi_n\}$ and write $\{\pi_1, \ldots, \pi_n\}(\varepsilon)$. For any nonempty finite set of elementary steps Π $(= \{\pi_1, \ldots, \pi_n\})$, we will say about the path $\{\pi_{i_1}, \ldots, \pi_{i_k}\}$ as *a path over* Π. The necessary strategy is extracted in an obvious way from the following

Theorem 6 (on Backtracking) *Let ε be an initial minimal state and $\Pi(= \{\pi_1, \ldots, \pi_n\})$ be a finite nonempty set of elementary steps. Further, let us consider an arrow diagram, all nodes of which are states of the form $\overline{\pi}(\varepsilon)$, where $\overline{\pi}$ is a path over Π; and for any paths $\overline{\alpha}$ and $\overline{\beta}$ over Π, $\overline{\alpha}(\varepsilon) \longrightarrow \overline{\beta}(\varepsilon)$ if there is a path $\overline{\gamma}$ over Π such that $\overline{\gamma}(\overline{\alpha}(\varepsilon)) = \overline{\beta}(\varepsilon)$. Then the diagram descibes all the paths over Π and all their intermediate minimal states between and including ε and $\{\pi_1, \ldots, \pi_n\}(\varepsilon)$.*

Proof follows immediately from the Theorem 4.

Our backtracking strategy limited to $[A]$-actions is as follows: If we need to defeat actions, or steps, $\pi_{i_1}, \ldots, \pi_{i_k}$ (all over Π) associated with the formulas $A_{i_1}, \ldots, A_{i_k}$, we should cut off those steps from Π and consider paths over $\Pi \setminus \{\pi_{i_1}, \ldots, \pi_{i_k}\}$. Moreover, according to the Theorem 1, every such a path, say $\{\pi_{j_1}, \ldots, \pi_{j_n}\}$, associated with the formulas $A_{j_1}, \ldots, A_{j_n}$ could be covered by alone $[A_{j1} \wedge \ldots \wedge A_{j_n}]$-action.

Thus, the computer must keep in its memory a trace of its path. There is no possibility to go back to the initial state if the computer only knows the final state ε_1 and the path F, except trivial cases.

5 More about the Lattice AGE

We maintain the belief that the proper understanding of the structure of the continuum can help in the solution of some problems concerning its tractable elements. Although we are not able now to point out any immediate application of the results of this section below, we hope that, in future, it could probably help in forming necessary strategies in knowledge processing.

At first, we establish an algebraic characterization of the lattice AGE.

Theorem 7 *The lattice* AGE *is a co-atomic continuous Heyting algebra,in which each nonunit element is an infimum of the greater co-atoms.*

Proof (Deferred to the Appendix).

From a computer-theoretical point of view, the lattice AGE is proved to be isomorphic to a domain determined by the information system $(\boldsymbol{D}, \star, \boldsymbol{Con}, \vdash)$ in the sense of [Sco 82], where $\boldsymbol{D}$ is the set of all the formulas in the language L^*, $\boldsymbol{Con}$ is all the finite subsets of $\boldsymbol{D}$ and, for every $u \in \boldsymbol{Con}$ and formula A, $u \vdash A$ means $\vdash_{E^*} \wedge u \to A$. We postpone providing a proof of this fact to [Mur 94].

6 Appendix

Assume $\mathcal{U} \subseteq \mathsf{AGE}$. We say on functions F and G of the type $\mathsf{AGE} \to \mathsf{AGE}$ that G *$\mathcal{U}$-majorizes* F, if $F(\overline{\varepsilon}) \leq G(\overline{\varepsilon})$ for every $\overline{\varepsilon} \in \mathcal{U}$.

Proposition 1 *Let F and G be Scott-continuous operations on* AGE. *Then G* AGE*-majorizes F if and only if G* AFE*-majorizes F.*

Proof. Let $\overline{\varepsilon_0} \in \mathsf{AGE}$. According to the Theorem 6.4 and the embedding $\varepsilon \mapsto \overline{\varepsilon}$ from the proof of the Theorem 4.3, both in [KM 93], we know that

$$\overline{\varepsilon_0} = \sqcup\{\varepsilon \,|\varepsilon \in \mathsf{AFE}, \varepsilon \ll \overline{\varepsilon_0}\}.$$

Then by means of the Scott's limit theorem (the Proposition 2.5 in [Sco 72] or the Proposition II-2.1 in [GHKLMS 80]), the equation

$$H(\overline{\varepsilon_0}) = \sqcup\{H(\varepsilon) \,|\varepsilon \in \mathsf{AFE}, \varepsilon \ll \overline{\varepsilon_0}\}$$

holds for any Scott-continuous operation H. The conclusion follows from a well-known lattice argument.

Lemma 1 *If* $\vdash_{E_{fde}} A \rightarrow B$ *then* $[B](\overline{\varepsilon}) \leq [A](\overline{\varepsilon})$ *for every generalized state* $\overline{\varepsilon}$.

Proof. In view of the Proposition 1, it is enough to prove that $[A]$ AFE-majorizes the operatiob $[B]$. At first, we will prove that

$$m(Tset(B)) \leq m(Tset(A)), \tag{3}$$

provided that $\vdash_{E_{fde}} A \rightarrow B$.

Let $s \in m(Tset(A))$. In vitue of the Theorem 1 in [Mur 93], we have: $\boldsymbol{t} \sqsubseteq s(A) \sqsubseteq s(B)$. Thus, $s^{V(B)\perp}(B) = s(B)$ and $s^{V(B)\perp} \leq s$. Then, there is $s' \in m(Tset(B))$ such that $s' \leq s^{V(B)\perp} \leq s$. Hence, (3) is proved. Now, we complete using the equation (2).

Lemma 2 *For any formulas* A *and* B*, the equation*

$$m(Tset(A)) \sqcup m(Tset(B)) = m(Tset(A \wedge B))$$

holds.

Proof. Previously, notice that the conditionals $A \wedge B \rightarrow A$ and $A \wedge B \rightarrow B$ are derived in E_{fde}. Then in virtue of the Lemma 1, $[A](\overline{\varepsilon}) \leq [A \wedge B](\overline{\varepsilon})$ hold for every generalized state $\overline{\varepsilon}$. According to [KM 93], we have

$$\overline{\varepsilon} \sqcup m(Tset(A)) \sqcup m(Tset(B)) \leq \overline{\varepsilon} \sqcup m(Tset(A \wedge B)) \tag{4}$$

for any generalized state $\overline{\varepsilon}$. Subsituting the lowest element in AGE for $\overline{\varepsilon}$ in (4) we receive:

$$m(Tset(A)) \sqcup m(Tset(B)) \leq m(Tset(A \wedge B)).$$

To complete is enough to show that

$$Tset(A \wedge B) \leq \{s_0 \sqcup s_1 \,|\, s_0 \in m(Tset(A)), s_1 \in m(Tset(B))\}$$

(cf. [KM 93], the Theorem 3.2).

Let $\boldsymbol{t} \sqsubseteq s_0(A)$ and $\boldsymbol{t} \sqsubseteq s_1(B)$. In virtue of the Proposition 2 in [KM 93], we have $\boldsymbol{t} \sqsubseteq s_0 \sqcup s_1(A)$, $\boldsymbol{t} \sqsubseteq s_0 \sqcup s_1(B)$ and, hence, $\boldsymbol{t} \sqsubseteq s_0 \sqcup s_1(A \wedge B)$. It is clear that $(s_0 \sqcup s_1)^{V(A \wedge B)\perp} \in m(Tset(A \wedge B))$ and $(s_0 \sqcup s_1)^{V(A \wedge B)\perp} \leq s_0 \sqcup s_1$.

So, the inequality (4) is proved. That completes the proof of the Lemma 2.

Proof of the Theorem 1 . We use the Theorem 3.1, the Theorem 4.3, both in [KM 93], and the Lemma 2 to conclude:

$$[A]([B](\overline{\varepsilon})) = \overline{\varepsilon} \sqcup \overline{Tset\,(A)} \sqcup \overline{Tset\,(B)} =$$

$$\overline{\varepsilon} \sqcup \overline{m(Tset\,(A))} \sqcup \overline{m(tsb)} = \overline{\varepsilon} \sqcup \overline{m(Tset\,(A \wedge B))} =$$

$$\overline{\varepsilon} \sqcup \overline{Tset\,(A \wedge B)} = [A \wedge B](\overline{\varepsilon}).$$

Our purpose now is to present a useful representation of the finite epistemic states. Let s be a finite setup and p^* be a literal, that is a formula of the form p or $\neg p$, where $p \in \mathsf{Var}$. With every setup s, we associate a nonempty collection of sets of literals $\kappa(s)$ as follows:

$$p^* \in \kappa(s) \overset{\text{def}}{\Longleftrightarrow} \begin{cases} p^* = p \quad \text{and} \quad \boldsymbol{t} \sqsubseteq s(p) \\ \qquad\quad \text{or} \\ p^* = \neg p \quad \text{and} \quad \boldsymbol{f} \sqsubseteq s(p). \end{cases}$$

Then, let $\kappa(\varepsilon)$ mean $\{\kappa(s)\,|\,s \in \varepsilon\}$. For any nonempty collection of sets of literals Σ, denote via $M(\Sigma)$ the minimal sets in Σ with respect to $\subseteq$. It is clear that $M(\Sigma)$ is never empty when Σ is a collection of finite sets. Consequently, $M(\kappa(\varepsilon))$ is nonempty for any finite epistemic state ε and

$$M(\kappa(\varepsilon)) = \kappa(\varepsilon) \tag{5}$$

for any minimal state ε.

Lemma 3 *For any setups s_1 and s_2,*

$$\kappa(s_1 \sqcup s_2) = \kappa(s_1) \cup \kappa(s_2)$$

and

$$s_1 \leq s_2 \textit{ if and only if } \kappa(s_1) \subseteq \kappa(s_2).$$

Proof is obvious.

We put in order the nonempty collections of sets of literals as follows:

$$\Sigma_1 \leq \Sigma_2 \overset{\text{def}}{\Longleftrightarrow} \text{ for every } x \in \Sigma_2 \text{ there is } y \in \Sigma_1 \text{ such that } y \subseteq x.$$

It is easy to check that the relation $\leq$ is a partial order on the set

$$\{M(\Sigma)\,|\Sigma \text{ is a nonempty finite collection of finite sets of literals}\}.$$

We denote that partial ordered set via Δ.

Lemma 4 *For any epistemic states ε_1 and ε_2,*

$$\varepsilon_1 \leq \varepsilon_2 \textit{ if and only if } \kappa(\varepsilon_1) \leq \kappa(\varepsilon_2).$$

Proof. Assume $\varepsilon_1 \leq \varepsilon_2$ and $\kappa(s_2) \in \kappa(\varepsilon_2)$ for some $s_2 \in \varepsilon_2$. According to the Lemma 3, there is a setup $s_1 \in \varepsilon_1$ such that $\kappa(s_1) \subseteq \kappa(s_2)$.

Now, suppose $\kappa(\varepsilon_1) \leq \kappa(\varepsilon_2)$ and $s_2 \in \varepsilon_2$. Then $\kappa(s_2) \in \kappa(\varepsilon_2)$ and, applying the Lemma 3 again, we receive that there is a setup $s_1 \in \varepsilon_1$ such that $s_1 \leq s_2$.

We will need furthermore the following simple property.

Lemma 5 *For any finite state ε, $M(\kappa(\varepsilon)) = \kappa(m(\varepsilon))$.*

Proof. We use the Lemma 3 to conclude:

$$\begin{array}{rcl}
x \in M(\kappa(\varepsilon)) & \Longleftrightarrow & x = \kappa(s_1) \text{ for some } s_1 \in \varepsilon \text{ and } x \subseteq y \text{ for every } y \in \varepsilon \\
& \Longleftrightarrow & x \subseteq \kappa(s) \text{ for every } s \in \varepsilon \text{ and } x = \kappa(s_1) \text{ for some } s_1 \in \varepsilon \\
& \Longleftrightarrow & x = \kappa(s_1) \text{ for some } s_1 \in \varepsilon \text{ and } s_1 \leq s \text{ for every } s \in \varepsilon \\
& \Longleftrightarrow & x = \kappa(s_1) \text{ and } s_1 \in m(\varepsilon) \\
& \Longleftrightarrow & x \in \kappa(m(\varepsilon)).
\end{array}$$

We complete presenting the representation by the

Lemma 6 *The mapping $\varepsilon \mapsto \kappa(\varepsilon)$ is a lattice isomorphism from* AFE *onto Δ and the latter is a lattice with operations:*

$$\kappa(\varepsilon_1) \sqcap \kappa(\varepsilon_2) = M(\kappa(\varepsilon_1) \cup \kappa(\varepsilon_2)),$$

$$\kappa(\varepsilon_1) \sqcup \kappa(\varepsilon_2) = M(\{\kappa(s_1) \cup \kappa(s_2)\,|s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\}).$$

Proof. The first part follows from the Lemma 4 and the following notice. Let $\Sigma \in \Delta$ and $x \in \Sigma$. We define a setup s^x as follows:

$$s^x(p) \stackrel{\text{def}}{=} \begin{cases} \boldsymbol{t} & \text{if } p \in x \text{ and } \neg p \notin x \\ \boldsymbol{f} & p \notin x \text{ and } \neg p \in x \\ \top & p \in x \text{ and } \neg p \in x \\ \bot & \text{otherwise,} \end{cases}$$

and $\varepsilon \stackrel{\text{def}}{=} \{s^x \,|\, x \in \Sigma\}$. Then $\varepsilon \in \mathsf{AFE}$, because $s^x \leq s^y$ if and only if $x \subseteq y$, and $\kappa(\varepsilon) = \Sigma$, because $\kappa(s^x) = x$ for every $x \in \Sigma$.

Now, we prove the rest. We use the Lemma 5, the Theorem 3.2 in [KM 93] and the first part proved to conclude:

$$\kappa(\varepsilon_1) \sqcap \kappa(\varepsilon_2) = \kappa(\varepsilon_1 \sqcap \varepsilon_2) = \kappa(m(\varepsilon_1 \cup \varepsilon_2)) =$$

$$M(\kappa(\varepsilon_1 \cup \varepsilon_2)) = M(\kappa(\varepsilon_1) \cup \kappa(\varepsilon_2)),$$

and additionally with help of the Lemma 3:

$$\kappa(\varepsilon_1) \sqcup \kappa(\varepsilon_2) = \kappa(\varepsilon_1 \sqcup \varepsilon_2) =$$

$$\kappa(m(\{s_1 \sqcup s_2 \,|\, s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\})) =$$
$$M(\kappa(\{s_1 \sqcup s_2 \,|\, s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\})) =$$
$$M(\{\kappa(s_1) \cup \kappa(s_2) \,|\, s_1 \in \varepsilon_1 . s_2 \in \varepsilon_2\}).$$

Aiming to prove the Theorem 2, we will temporarily extend our propositional language L adding one new symbol $\star$ as meaning a new atomic formula. We denote via E^* the calculus E_{fde} of first degree entailment in the new language L* with one additional axiom scheme $A \to \star$.

Lemma 7 *The calculus E^* is a conservative extension of E_{fde}, that is, $\vdash_{E^*} A \to B$ implies $\vdash_{E_{fde}} A \to B$, provided that A and B do not contain $\star$.*

Proof. Extend the definition of the setup to the language L* as follows: $s(\star) = \boldsymbol{t}$. Then $\vdash_{E^*} A \to B$ implies $s(A) \leq s(B)$ for every new setup s, where $\leq$ is the order on $\mathsf{L4}$. Now, the conclusion follows from [AB 75].

With every nonempty finite collection of finite sets of literals Σ, we associate the formula A_Σ in the extended language, which is defined as follows:

$$A_\Sigma \stackrel{\text{def}}{=} \begin{cases} \bigvee\{\bigwedge x \,|\, x \in \Sigma\} & \text{if } \emptyset \not\in \Sigma \\ \star & \text{otherwise.} \end{cases}$$

Lemma 8 *For any such Σ, $\vdash_{E^*} A_\Sigma \leftrightarrow A_{M(\Sigma)}$.*

Proof. We need just to perform replacements with respect to some equivalences of the form $(A \wedge B) \vee B \leftrightarrow B$ that are proved in E_{fde} (cf. [AB 75]).

Lemma 9 *For any minimal states* ε_1 *and* ε_2*,*

$$\vdash_{E^*} A_{\kappa(\varepsilon_1)} \wedge A_{\kappa(\varepsilon_2)} \leftrightarrow A_{\kappa(\varepsilon_1) \sqcup \kappa(\varepsilon_2)}$$

and

$$\vdash_{E^*} A_{\kappa(\varepsilon_1)} \vee A_{\kappa(\varepsilon_2)} \leftrightarrow A_{\kappa(\varepsilon_1) \sqcap \kappa(\varepsilon_2)}.$$

Proof. With help of the Lemmas 6 and 8, we receive:

$$\begin{aligned}
\vdash_{E^*} A_{\kappa(\varepsilon_1)} \wedge A_{\kappa(\varepsilon_2)} & \leftrightarrow \bigvee \{\bigwedge x_1 \wedge \bigwedge x_2 \mid x_1 \in \kappa(\varepsilon_1), x_2 \in \kappa(\varepsilon_2)\} \\
& \leftrightarrow \bigvee \{\bigwedge (x_1 \cup x_2) \mid x_1 \in \kappa(\varepsilon_1), x_2 \in \kappa(\varepsilon_2)\} \\
& \leftrightarrow A_{\{\kappa(s_1) \cup \kappa(s_2) \mid s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\}} \\
& \leftrightarrow A_{M(\{\kappa(s_1) \cup \kappa(s_2) \mid s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\})} \\
& \leftrightarrow A_{\kappa(\varepsilon_1) \sqcup \kappa(e_2)}, \\
\vdash_{E^*} A_{\kappa(\varepsilon_1)} \vee A_{\kappa(\varepsilon_2)} & \leftrightarrow \bigvee \{\bigwedge x \mid x \in \kappa(\varepsilon_1) \cup \kappa(\varepsilon_2)\} \\
& \leftrightarrow A_{\kappa(\varepsilon_1) \cup \kappa(\varepsilon_2)} \\
& \leftrightarrow A_{M(\kappa(\varepsilon_1) \cup \kappa(\varepsilon_2))} \\
& \leftrightarrow A_{\kappa(\varepsilon_1) \sqcap \kappa(\varepsilon_2)},
\end{aligned}$$

providing $\emptyset \notin \kappa(\varepsilon_1) \cup \kappa(\varepsilon_2)$. If $\emptyset \in \kappa(e_1) \cup \kappa(\varepsilon_2)$, for instance, $\emptyset \in \kappa(\varepsilon_1)$, then, first, $\emptyset \in \kappa(\varepsilon_1) \sqcap \kappa(\varepsilon_2)$ and, hence,

$$\begin{aligned}
\vdash_{E^*} A_{\kappa(\varepsilon_1)} \vee A_{\kappa(\varepsilon_2)} & \leftrightarrow \star \vee A_{\kappa(\varepsilon_2)} \\
& \leftrightarrow \star \\
& \leftrightarrow A_{\kappa(\varepsilon_1) \sqcap \kappa(\varepsilon_2)}.
\end{aligned}$$

Second, ε_1 is the zero in AFE and, hence, $\kappa(\varepsilon_1) \sqcup \kappa(\varepsilon_2) = \kappa(\varepsilon_2)$. Therefore,

$$\begin{aligned}
\vdash_{E^*} A_{\kappa(\varepsilon_1)} \wedge A_{\kappa(\varepsilon_2)} & \leftrightarrow \star \wedge A_{\kappa(\varepsilon_2)} \\
& \leftrightarrow A_{\kappa(\varepsilon_2)} \\
& \leftrightarrow A_{\kappa(\varepsilon_1) \sqcup \kappa(\varepsilon_2)}.
\end{aligned}$$

Lemma 10 *Let A be a formula of the language* L. *Denote* $\varepsilon \overset{\text{def}}{=} m(Tset(A))$ *and* $\varepsilon_1 \overset{\text{def}}{=} m(Fset(A))$. *Then* $\vdash_{E^*} A \leftrightarrow A_{\kappa(\varepsilon)}$ *and* $\vdash_{E^*} \neg A \leftrightarrow A_{\kappa(\varepsilon_1)}$.

Proof (by induction on the length of A). Case: $A = p_i$. Then $\varepsilon = \{s\}$ and $\varepsilon_1 = \{s_1\}$, where

$$s(p) = \begin{cases} \boldsymbol{t} & \text{if } p = p_i \\ \bot & \text{otherwise} \end{cases} \text{ and } s_1(p) = \begin{cases} \boldsymbol{f} & \text{if } p = p_i \\ \bot & \text{otherwise.} \end{cases}$$

Therefore, $\kappa(\varepsilon) = \{\{p_i\}\}$ and $\kappa(\varepsilon_1) = \{\{\neg p_i\}\}$. Consequently, $A_{\kappa(\varepsilon)} = p_i$ and $A_{\kappa(\varepsilon_1)} = \neg p_i$.

Case: $A = \neg B$. Denote $\varepsilon_2 \overset{\text{def}}{=} m(Fset(B))$ and notice that $\varepsilon = \varepsilon_2$. According to the induction hypothesis, we have $\vdash_{E_*} \neg B \leftrightarrow A_{\kappa(\varepsilon_2)}$, that is, $\vdash_{E^*} A \leftrightarrow A_{\kappa(\varepsilon)}$. Furthermore, denote $\varepsilon_3 \overset{\text{def}}{=} m(Tset(B))$ and notice that $\varepsilon_3 = \varepsilon_1$. Again by the induction hypothesis, we receive $\vdash_{E^*} B \leftrightarrow A_{\kappa(\varepsilon_3)}$, that is, $\vdash_{E^*} \neg A \leftrightarrow A_{\kappa(\varepsilon_1)}$.

Denote for the next two cases:

$$\varepsilon_2 \overset{\text{def}}{=} m(Tset(B)),\ \varepsilon_3 \overset{\text{def}}{=} m(Tset(C)),\ \varepsilon_4 \overset{\text{def}}{=} m(Fset(B)),\\ \varepsilon_5 \overset{\text{def}}{=} m(Fset(C)).$$

According to the induction hypothesis, we have:

$$\vdash_{E^*} B \leftrightarrow A_{\kappa(\varepsilon_2)},\ \vdash_{E^*} C \leftrightarrow A_{\kappa(\varepsilon_3)},\ \vdash_{E^*} \neg B \leftrightarrow A_{\kappa(\varepsilon_4)},\ \vdash_{E^*} \neg C \leftrightarrow A_{\kappa(\varepsilon_5)}.$$

Case: $A = B \wedge C$. My means of the Lemmas 9, 6 and 2, we consequently receive:

$$\begin{aligned} \vdash_{E^*} B \wedge C &\leftrightarrow A_{\kappa(\varepsilon_2)} \wedge A_{\kappa(\varepsilon_3)} \\ &\leftrightarrow A_{\kappa(\varepsilon_2) \sqcup \kappa(\varepsilon_3)} \\ &\leftrightarrow A_{\kappa(\varepsilon_2 \sqcup \varepsilon_3)} \\ &\leftrightarrow A_{\kappa(\varepsilon)}. \end{aligned}$$

In the next step, we establish the following equation:

$$m(Fset(B) \cup Fset(C)) = m(Fset(B \wedge C)). \tag{6}$$

Notice that (6) follows from two inequalities:

$$Fset(B \wedge C) \leq Fset(B) \cup Fset(C)$$

and

$$Fset(B) \cup Fset(C) \leq m(Fset(B \wedge C)),$$

because then we receive:

$$m(Fset(B \wedge C)) \leq m(Fset(B) \cup Fset(C))$$

and

$$m(Fset(B) \cup Fset(C)) \leq m(m(Fset(B \wedge C))) = m(Fset(B \wedge C)).$$

So, let $s \in Fset(B) \cup Fset(C)$ and, for instance, $s \in Fset(B)$. It is easy to notice that then $s \in Fset(B \wedge C)$.

Assume $s \in m(Fset(B \wedge C))$. Then $V(s) \subseteq V(B \wedge C)$ and $\boldsymbol{f} \sqsubseteq s(B)$ or $\boldsymbol{f} \sqsubseteq s(C)$. Let, for instance, $\boldsymbol{f} \sqsubseteq s(B)$. Consequently, there is a setup s_1 such that $V(s_1) \subseteq V(B)$ and $\boldsymbol{f} \sqsubseteq s_1(B)$. However, $\boldsymbol{f} \sqsubseteq s_1(B \wedge C)$, too. But s is minimal in $Fset(B \wedge C)$. So, $s = s_1$ that implies $s \in Fset(B) \cup Fset(C)$.

Now, with help of the Lemmas 9 and 6, (6) and Theorem 3.2 in [KM 93], we receive:

$$\begin{aligned} \vdash_{E^*} \neg(B \wedge C) & \leftrightarrow \neg B \vee \neg C \\ & \leftrightarrow A_{\kappa(\varepsilon_4)} \vee A_{\kappa(\varepsilon_5)} \\ & \leftrightarrow A_{\kappa(\varepsilon_4) \sqcap \kappa(\varepsilon_5)} \\ & \leftrightarrow A_{\kappa(\varepsilon_4 \sqcap \varepsilon_5)} \\ & \leftrightarrow A_{\kappa(\varepsilon_1)}. \end{aligned}$$

Case: $A = B \vee C$. In the same manner as in the proof of (6), we establish the equation

$$m(Tset(B) \cup Tset(C)) = m(Tset(B \vee C)) \tag{7}$$

that follows from two inequalities:

$$Tset(B \vee C) \leq Tset(B) \cup Tset(C)$$

and

$$Tset(B) \cup Tset(C) \leq m(Tset(B \vee C)).$$

By means of the Lemmas 9 and 6, Theorem 3.2 in [KM 93] and (7), we receive:

$$\begin{aligned} \vdash_{E^*} B \vee C & \leftrightarrow A_{\mathcal{K}(\varepsilon_2)} \vee A_{\mathcal{K}(\varepsilon_3)} \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon_2) \sqcap \mathcal{K}(\varepsilon_3)} \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon_2 \sqcap \varepsilon_3)} \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon)}. \end{aligned}$$

To continue we previously prove the equation

$$m(Fset(B)) \sqcup m(Fset(C)) = m(Fset(B \vee C)) \tag{8}$$

Indeed, with help of the Lemma 2, we receive:

$$m(Fset(B)) \sqcup m(Fset(C)) = m(Tset(\neg B)) \sqcup m(Tset(\neg C)) =$$

$$m(Tset(\neg B \wedge \neg C)) = m(Tset(\neg(B \vee C))) = m(Fset(B \vee C)).$$

Finally, using the Lemmas 9 and 6 and equation (8), we conclude:

$$\begin{aligned} \vdash_{E^*} \neg(B \vee C) & \leftrightarrow \neg B \wedge \neg C \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon_4)} \wedge A_{\mathcal{K}(\varepsilon_5)} \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon_4) \sqcup \mathcal{K}(\varepsilon_5)} \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon_4 \sqcup \varepsilon_5)} \\ & \leftrightarrow A_{\mathcal{K}(\varepsilon_1)}. \end{aligned}$$

That completes the proof of the Lemma 10.

Proof of the Theorem 2. The part "only if" follows from the Lemma 1.

Note that both A and B are of the language L. Suppose $[A](\overline{\varepsilon}) = [B](\overline{\varepsilon})$ for every generalized state $\overline{\varepsilon}$. In particular, $[A](\varepsilon) = [B](\varepsilon)$ for every minimal state ε. It follows from (2) that $m(Tset(A)) = m(Tset(B))$. The Lemma 10 implies that

$$\begin{aligned} \vdash_{E^*} A & \leftrightarrow A_{\mathcal{K}(m(Tset(A)))} \\ & \leftrightarrow A_{\mathcal{K}(m(Tset(B)))} \\ & \leftrightarrow B. \end{aligned}$$

Applying the Lemma 7, we conclude that $\vdash_{E_{fde}} A \leftrightarrow B$.

Lemma 11 *For any formulas* $A_1, \ldots, An$, $\boldsymbol{t} \sqsubseteq s(A_1 \wedge \ldots \wedge A_n)$ *if and only if* $\boldsymbol{t} \sqsubseteq s(A_1), \ldots, \boldsymbol{t} \sqsubseteq s(A_n)$.

Proof is obvious.

Lemma 12 *For any minimal state* ε, *the equation* $\varepsilon = m\Big(Tset\left(A_{\kappa(\varepsilon)}\right)\Big)$ *holds.*

Proof. Assume $s \in \varepsilon$. Then $\kappa(s) \in \kappa(\varepsilon)$. For every $p^* \in \kappa(s)$, $\boldsymbol{t} \sqsubseteq s(p^*)$. According to the Lemma 11, $\boldsymbol{t} \sqsubseteq s(\wedge\kappa(s)$. Thus, we have $\boldsymbol{t} \sqsubseteq s(A_{\kappa(\varepsilon)})$ and $V(s) \subseteq V(\varepsilon) = V(A_{\kappa(\varepsilon)})$. Therefore, $s \in Tset\left(A_{\kappa(\varepsilon)}\right)$ and, hence, $\varepsilon \subseteq Tset\left(A_{\kappa(\varepsilon)}\right)$. The latter follows an inequality $m\Big(Tset\left(A_{\kappa(\varepsilon)}\right)\Big) \leq \varepsilon$.

Now assume $s \in m\Big(Tset\left(A_{\kappa(\varepsilon)}\right)\Big)$. We have in particular $s \in Tset\left(A_{\kappa(\varepsilon)}\right)$ and, hence, $\boldsymbol{t} \sqsubseteq s(A_{\kappa(\varepsilon)})$. Thus, there is $s_1 \in \varepsilon$ such that

$$\boldsymbol{t} \sqsubseteq s(\wedge\kappa(\varepsilon_1)) \tag{9}$$

Consider cases.

Case 1: $p \in V(s_1)$ and $s_1(p) = \boldsymbol{t}$. Then $p \in \kappa(s_1)$ and, according to the Lemma 11 and (9), $\boldsymbol{t} \sqsubseteq s(p)$.

Case 2: $p \in V(s_1)$ and $s_1(p) = \boldsymbol{f}$. Again, according to the Lemma 11 and (9), $\boldsymbol{t} \sqsubseteq s(\neg p)$, that is, $\boldsymbol{f} \sqsubseteq s(p)$.

Case 3: $p \in V(s_1)$ and $s_1(p) = \top$. Then $p, \neg p \in \kappa(s_1)$. Again, the Lemma 11 and (9) give $\boldsymbol{t} \sqsubseteq s(p)$ and $\boldsymbol{t} \sqsubseteq s(\neg p)$, that is, $s(p) = \top$.

Case 4: $p \notin V(s_1)$, that is, $s(p) = \bot$.

We have $s_1(p) \sqsubseteq s(p)$ in all cases. It means $s_1 \leq s$ that follows $\varepsilon \leq m\Big(Tset\left(A_{\kappa(\varepsilon)}\right)\Big)$.

Lemma 13 *For any minimal states* ε *and* ε_1, $\varepsilon \leq \varepsilon_1$ *if and only if* $\vdash_{E_{fde}} A_{\kappa(\varepsilon_1)} \rightarrow A_{\kappa(\varepsilon)}$.

Proof. Assume $\varepsilon \leq \varepsilon_1$, that is, $\varepsilon \sqcup \varepsilon_1 = \varepsilon_1$. In virtue of the Lemma 4, we have $\kappa(\varepsilon) \sqcup \kappa(\varepsilon_1) = \kappa(\varepsilon_1)$. Then, the Lemma 9 gives:

$$\begin{aligned} \vdash_{E_{fde}} A_{\kappa(\varepsilon_1)} &\leftrightarrow A_{\kappa(\varepsilon)} \wedge A_{\kappa(\varepsilon_1)} \\ &\rightarrow A_{\kappa(\varepsilon)}. \end{aligned}$$

Let now $\vdash_{E_{fde}} A_{\mathcal{K}(\varepsilon_1)} \rightarrow A_{\mathcal{K}(\varepsilon)}$. By means of the Lemma 1, we conclude that

$$m\Big(Tset\left(A_{\mathcal{K}(\varepsilon)}\right)\Big) \leq m\Big(Tset\left(A_{\mathcal{K}(\varepsilon_1)}\right)\Big),$$

that, with help of the Lemma 12, gives $\varepsilon \leq \varepsilon_1$.

Proof of the Theorem 3. Let us denote $[A](\varepsilon)$ and $[B](\varepsilon)$ via ε_1 and ε_2, respectively. Then, $\varepsilon_1 = \varepsilon_2$ if and only if $\varepsilon_1 \leq \varepsilon_2$ and $\varepsilon_2 \leq \varepsilon_1$, that is equivalent to $\vdash_{E_{fde}} A_{\mathcal{K}(\varepsilon_1)} \leftrightarrow A_{\mathcal{K}(\varepsilon_2)}$.

Lemma 14 *For any minimal states ε_1 and ε_2 and formula A,*

$$(m(\varepsilon_1 \cup \varepsilon_2))^-_A = m\big((\varepsilon_1)^-_A) \cup (\varepsilon_2)^-_A\big).$$

Proof. Assume $s \in (m(\varepsilon_1 \cup \varepsilon_2))^-_A$, that is, $s \in m(\varepsilon_1 \cup \varepsilon_2)$ and $\boldsymbol{t} \not\sqsubseteq s(A)$. Then $s \in (\varepsilon_1)^-_A \cup (\varepsilon_2)^-_A$. Suppose there exists $s' \in (\varepsilon_1)^-_A \cup (\varepsilon_2)^-_A$ such that $s' < s$. It implies that both $s' \in \varepsilon_1 \cup \varepsilon_2$ and $\boldsymbol{t} \not\sqsubseteq s'(A)$, that leads to a contradiction.

Now, let $s \in m\big((\varepsilon_1)^-_A \cup (\varepsilon_2)^-_A\big)$ that implies $s \in \varepsilon_1 \cup \varepsilon_2$ and $\boldsymbol{t} \not\sqsubseteq s\,(A)$. For any $s' \in \varepsilon_1 \cup \varepsilon_2$ such that $s' \leq s$, we conclude with help of the Proposition 2 in [KM 93] that $s' = s$. That implies that $s \in (m(\varepsilon_1 \cup \varepsilon_2))^-_A$.

Lemma 15 *For any minimal states ε_1 and ε_2 and formula A,*

$$(\varepsilon_1 \sqcap \varepsilon_2)_{\overline{A}} = (\varepsilon_1)_{\overline{A}} \sqcap (\varepsilon_2)_{\overline{A}}.$$

Proof. Consider the following cases.

Case: $(\varepsilon_1)_{\overline{A}} = \mathbf{1}$ and $(\varepsilon_2)_{\overline{A}} = \mathbf{1}$, that is, $(\varepsilon_1)^-_A = \emptyset$ and $(\varepsilon_2)^-_A = \emptyset$. In view of the Theorem 3.2 in [KM 93] and Lemma 14, it implies that

$$(\varepsilon_1 \sqcap \varepsilon_2)_{\overline{A}} = (m(\varepsilon_1 \cup \varepsilon_2))_{\overline{A}} = \mathbf{1}.$$

Case: $(\varepsilon_1)_{\overline{A}} = \mathbf{1}$ and $(\varepsilon_2)_{\overline{A}} \neq \mathbf{1}$. Again, in vitrue of the Lemma 14 and Theorem 3.2 in [KM 93], we receive:

$$(\varepsilon_1 \sqcap \varepsilon_2)^-_A = (m(\varepsilon_1 \cup \varepsilon_2))^-_A = m\big((\varepsilon_1)^-_A \cup (\varepsilon_2)^-_A\big) = m\big((\varepsilon_2)^-_A\big) = (\varepsilon_2)^-_A \neq \mathbf{1}.$$

Then, by definition, we have:

$$(\varepsilon_1 \sqcap \varepsilon_2)_{\overline{A}} = (\varepsilon_1 \sqcap \varepsilon_2)^-_A = (\varepsilon_2)_{\overline{A}} = \mathbf{1} \sqcap (\varepsilon_2)_{\overline{A}}.$$

Case: $(\varepsilon_1)_{\overline{A}} \neq \mathbf{1}$ and $(\varepsilon_2)_{\overline{A}} = \mathbf{1}$. It is considered analogiously.

Case: $(\varepsilon_1)_{\overline{A}} \neq \mathbf{1}$ and $(\varepsilon_2)_{\overline{A}} \neq \mathbf{1}$. Then, the same argumentation gives:

$$(\varepsilon_1 \sqcap \varepsilon_2)^-_A = (m(\varepsilon_1 \cup \varepsilon_2))^-_A = m\big((\varepsilon_1)^-_A \cup (\varepsilon_2)^-_A\big) \neq \emptyset.$$

It implies:

$$(\varepsilon_1 \sqcap \varepsilon_2)_{\overline{A}} = (m(\varepsilon_1 \cup \varepsilon_2))^-_A = (\varepsilon_1)_{\overline{A}} \sqcap (\varepsilon_2)_{\overline{A}}.$$

Lemma 16 *The lattice* AFE *is distributive.*

Proof. First of all, recall that the inequality

$$(\varepsilon_1 \sqcap \varepsilon_3) \sqcup (\varepsilon_2 \sqcap \varepsilon_3) \leq (\varepsilon_1 \sqcup \varepsilon_2) \sqcap \varepsilon_3$$

holds in every lattice. Thus, we need just to prove that

$$(\varepsilon_1 \sqcup \varepsilon_2) \sqcap \varepsilon_3 \leq (\varepsilon_1 \sqcap \varepsilon_3) \sqcup (\varepsilon_2 \sqcap \varepsilon_3)$$

hold in AFE. We will establish first the following inequality:

$$\{s_1 \sqcup s_2 \,|s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\} \cup \varepsilon_3 \leq \{s_1 \sqcup s_2 \,|s_1 \in m(\varepsilon_1 \cup \varepsilon_3), s_2 \in m(\varepsilon_2 \cup \varepsilon_3)\}.$$

Assume $s_1 \in m(\varepsilon_1 \cup \varepsilon_3)$ and $s_2 \in m(\varepsilon_2 \cup \varepsilon_3)$. Consider two cases.

Case: $s_1 \in \varepsilon_3$ or $s_2 \in \varepsilon_3$. Then, for some $s \in \varepsilon_3$, $s \leq s_1 \sqcup s_2$.

Case: $s_1 \in \varepsilon_1$ and $s_2 \in \varepsilon_2$. The conclusion is obvious.

Now, we receive:

$$\begin{aligned}
(\varepsilon_1 \sqcup \varepsilon_2) \sqcap \varepsilon_3 &= m((\varepsilon_1 \sqcup \varepsilon_2) \cup \varepsilon_3) \\
&\leq (\varepsilon_1 \sqcup \varepsilon_2) \cup \varepsilon_3 \\
&= m(\{s_1 \sqcup s_2 \,|s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\}) \cup \varepsilon_3 \\
&\leq \{s_1 \sqcup s_2 \,|s_1 \in \varepsilon_1, s_2 \in \varepsilon_2\} \cup \varepsilon_3 \\
&\leq \{s_1 \sqcup s_2 \,|s_1 \in m(\varepsilon_1 \cup \varepsilon_3), s_2 \in m(\varepsilon_2 \cup \varepsilon_3)\} \\
&= \{s_1 \sqcup s_2 \,|s_1 \in \varepsilon_1 \sqcap \varepsilon_3, s_2 \in \varepsilon_2 \sqcap \varepsilon_3\}.
\end{aligned}$$

However, $(\varepsilon_1 \sqcup \varepsilon_2) \sqcap \varepsilon_3$ is a minimal state. Hence,

$$\begin{aligned}
(\varepsilon_1 \sqcup \varepsilon_2) \sqcap \varepsilon_3 &\leq m(\{s_1 \sqcup s_2 \,|s_1 \in \varepsilon_1 \sqcap \varepsilon_3, s_2 \in \varepsilon_2 \sqcap \varepsilon_3\}) \\
&= (\varepsilon_1 \sqcap \varepsilon_3) \sqcup (\varepsilon_2 \sqcap \varepsilon_3).
\end{aligned}$$

We recall two definitions from [KM 93] amd two from [Mur 93], respectively:

$$[A \rightarrow B]'(s) \overset{\text{def}}{=} \begin{cases} \{s \sqcup s' \,|\, s' \in Tset(B)\} & \text{if } \boldsymbol{t} \sqsubseteq s(A) \\ \{s\} & \text{otherwise,} \end{cases}$$

$$[A \rightarrow B](\varepsilon) \overset{\text{def}}{=} \cup\Big\{[A \rightarrow B]'(s) \,|\, s \in \varepsilon\Big\},$$

$$\varepsilon^+_A \overset{\text{def}}{=} \{s \,|\, s \in \varepsilon, \boldsymbol{t} \sqsubseteq s(A)\} \text{ and } \varepsilon_A \overset{\text{def}}{=} \begin{cases} \varepsilon^+_A & \text{if } \varepsilon^+_A \neq \emptyset \\ \text{the unit } \mathbf{1} \text{ in } \mathsf{AGE} & \text{otherwise.} \end{cases}$$

Proof of the Theorem 4. First of all, notice that

$$[A \rightarrow B](\varepsilon) = m\Big(\cup\Big\{[A \rightarrow B]'(s) \,|\, s \in \varepsilon\Big\}\Big)$$

and

$$\varepsilon_{\overline{A}} \sqcap \varepsilon_A = \varepsilon \tag{10}$$

for any minimal state ε. The former follows from the Theorems 7.7, 4.3 and 3.1 in [KM 93], the latter is obvious. Using those equations consequently and the distributivity of AFE, we receive:

$$\begin{aligned} [A \rightarrow B](\varepsilon) &= m\Big(\varepsilon^-_A \cup \Big\{s \sqcup s' \,\Big|\, s \in \varepsilon^+_A, s' \in m(Tset(B))\Big\}\Big) \\ &= \varepsilon_{\overline{A}} \sqcap (\varepsilon_A \sqcup m(Tset(B))) \\ &= \varepsilon \sqcup (\varepsilon_{\overline{A}} \sqcap m(Tset(B))) \\ &= (\varepsilon \sqcup \varepsilon_{\overline{A}}) \sqcap (\varepsilon \sqcup m(Tset(B))) \\ &= \varepsilon_{\overline{A}} \sqcap (\varepsilon \sqcup m(Tset(B))) \\ &= \varepsilon_{\overline{A}} \sqcap [B](\varepsilon). \end{aligned}$$

Lemma 17 *For any formulas A, B and minimal state ε,*

$$m(Tset(B)) \leq \varepsilon_{\overline{A}} \textit{ implies } [B]\Big(\varepsilon_{\overline{A}}\Big) = ([B](\varepsilon))_{\overline{A}}.$$

Proof. First of all, we will show that, providing the premise, $\varepsilon^-_A = \emptyset$ if and only if $([B](\varepsilon))^-_A = \emptyset$.

The implication:

$$\varepsilon^-_A = \emptyset \Longrightarrow ([B](\varepsilon))^-_A = \emptyset$$

is always correct. It is so, since $\varepsilon \leq [B](\varepsilon)$.

Let, now, $\varepsilon^{-}_{A} \neq \emptyset$. In virtue of the premise and (10),

$$\varepsilon \sqcup m(Tset\,(B)) \leq \varepsilon \sqcup \varepsilon_{\overline{A}} = \varepsilon_{\overline{A}}.$$

Then, by means of (2) and (1), we receive:

$$([B](\varepsilon))_{\overline{A}} \leq (\varepsilon_{\overline{A}})_{\overline{A}} = \varepsilon_{\overline{A}}.$$

Now, we may suppose that both $(\varepsilon)^{-}_{A} \neq \emptyset$ and $([B](\varepsilon))^{-}_{A} \neq \emptyset$.
Assume $s \in \varepsilon_{\overline{A}} \sqcup m(Tset\,(B))$, that is (cf. [KM 93], Theorem 3.2),

$$s \in m\Big(\Big\{s_1 \sqcup s_2 \,\Big|\, s_1 \in \varepsilon_{\overline{A}}, s_2 \in m(Tset\,(B))\Big\}\Big).$$

In view of the premise, $\boldsymbol{t} \not\sqsubseteq s\,(A)$. Also, notice that

$$s \in \{s_1 \sqcup s_2 \,|\, s_1 \in \varepsilon, s_2 \in m(Tset\,(B))\}.$$

Let $s_3 \leq s$ and $s_3 \in \{s_1 \sqcup s_2 \,|\, s_1 \in \varepsilon, s_2 \in m(Tset\,(B))\}$. Notice that $\boldsymbol{t} \not\sqsubseteq s_3\,(A)$ (cf. [KM 93], Proposition 2). Thus,

$$s_3 \in \Big\{s_1 \sqcup s_2 \,\Big|\, s_1 \in \varepsilon_{\overline{A}}, s_2 \in m(Tset\,(B))\Big\}.$$

It implies the equation $s_3 = s$ that, in turn, implies that $s \in [B](\varepsilon)$ and $\boldsymbol{t} \not\sqsubseteq s\,(A)$.

Now, suppose $s \in ([B](\varepsilon))_{\overline{A}}$, that is, $s \in \varepsilon \sqcup m(Tset\,(B))$ and $\boldsymbol{t} \not\sqsubseteq s\,(A)$. In particular,

$$s \in m(\{s_1 \sqcup s_2 \,|\, s_1 \in \varepsilon, s_2 \in m(Tset\,(B))\}).$$

That implies that

$$s \in \Big\{s_1 \sqcup s_2 \,\Big|\, s_1 \in \varepsilon_{\overline{A}}, s_2 \in m(Tset\,(B))\Big\}.$$

Let $s_3 \leq s$ and $s_3 \in \Big\{s_1 \sqcup s_2 \,\Big|\, s_1 \varepsilon_{\overline{A}}, s_2 \in m(Tset\,(B))\Big\}$. Then,

$$s_3 \in \{s_1 \sqcup s_2 \,|\, s_1 \in \varepsilon, s_2 \in m(Tset\,(B))\}.$$

That implies the equation $s_3 = s$ that, in turn, implies that $s \in [B]\Big(\varepsilon_{\overline{A}}$