A. Avron
J. Ben-Naim
B. Konikowska

Processing Information from a Set of Sources

Abstract. We introduce a general framework for solving the problem of a computer collecting and combining information from various sources. Unlike previous approaches to this problem, in our framework the sources are allowed to provide information about complex formulae too. This is enabled by the use of a new tool — non-deterministic logical matrices. We also consider several alternative plausible assumptions concerning the framework. These assumptions lead to various logics. We provide strongly sound and complete proof systems for all the basic logics induced in this way.

Keywords: Information processing, multiple sources, non-deterministic matrices, non-classical logics, paraconsistency

1. Introduction

The idea considered in this paper has originated from Belnap, whose famous four-valued logic [7, 6] [1] stemmed from considering the problem of a computer collecting and combining information from various sources. Later Belnap's approach was extended by Carnielli and Lima-Marques in their *society semantics* [8] to consider various information collecting and processing strategies applied by the computer (or some other agent). However, both works considered just the simple case of sources providing information only about *atomic* formulas of some logical language (which corresponds to the case of simple relational databases). Unfortunately, this does not capture all the situations encountered in practice, for e.g. knowledge bases and disjunctive databases can also provide information about complex formulas. Accordingly, in this paper we extend the previous approaches in an essential way by allowing the sources to provide information about complex formulae too. This is enabled by the use of a new tool — non-deterministic logical matrices (*Nmatrices* — see [3, 4, 2]), which is necessary in view of the fact that ordinary logical matrices are unable to capture the above general case.

The structure of the paper is as follows. In Section 2 we describe our general framework for processing information coming from different sources, as well as several various plausible assumptions concerning it, leading to important special cases. In Section 3 we investigate the four basic logics

[1] Actually, this logic should be called Dunn-Belnap logic, since it was originally introduced by Dunn [15].

Studia Logica **0**: 1–**??**, 2007.

obtained by adopting the simplest such assumption, according to which a processor accepts any proposition declared true by one of its sources (even if this leads to contradictions). Two of these logics (Dunn-Belnap logic and the basic paraconsistent 3-valued logic) are well-known. The two others are new. Section 4 shortly investigates an alternative strategy, in which a processor initially accepts a proposition only if *all* its sources declare it to be true. This strategy also leads to a famous logic: Kleene's 3-valued logic. In Section 5 we introduce calculi of sequents for all these logics, and prove their strong soundness and completeness, as well as a strong version of the admissibility of the cut rule in them. Finally, in Section 6 we outline directions for future research. [2]

2. The framework

2.1. Informal description

Assume we have a framework for information collecting and processing, which consists of a set of information sources S and a processor P. The sources provide information about formulas of the classical propositional logic L_C (which we take here to be based on the connectives $\{\neg, \vee, \wedge\}$). We assume that for each such formula φ, a source $s \in S$ can say that φ is true, that φ is false, or that it has no knowledge about φ. Thus, every source defines some (possibly partial) valuation (using the two classical truth-values). In turn, the processor collects information from the sources, combines it according to some strategy, processes the result and finally defines its resulting combined valuation (denoted in the sequel by d) of formulas in L_C.

Clearly, for any formula $\varphi \in L_C$, the processor can encounter at the collecting stage four possible situations concerning the information it gets from the sources:

- It has information that φ is true but no information that φ is false
- It has information that φ is false but no information that φ is true
- It has both information that φ is true and information that φ is false
- It has no information on φ at all

[2] The first short description of our framework was given in [5]. In that paper, the basic proof system used here was derived using some general method, yielding a roundabout proof of a rather weak form of the soundness and completeness theorem for that system.

In view of the above, a natural logical domain for the considered framework features four logical values corresponding to the four cases above, which are usually denoted[3] by

$$\mathbf{t} = \{1\}, \ \mathbf{f} = \{0\}, \ \top = \{0, 1\}, \ \bot = \emptyset,$$

Here 1 and 0 represent the classical logical values of *true* and *false* (respectively), and so $\top$ represents inconsistent information, while $\bot$ denotes absence of information. Among these four truth values, we take as designated $\mathbf{t}$ and $\top$ — the truth values whose assignment to a formula φ means that the processor has information that φ is true (even though it might also have information that φ is false). This represents the so-called *weak semantics.* Another possible option could have been to consider *strong semantics*, whereby the only designated value is $\mathbf{t}$, which means φ is deemed satisfied if the processor has information that φ is true, but has no information that φ is false. However, the consequence relation induced by the strong semantics can be simulated by the weaker one employed here (see Subsection 2.4).

2.2. Variants of the model

The general model introduced above has many variants, corresponding to various assumptions on the kind of information provided by the sources and the strategy used by the processor to combine it. Within this general framework, we can classify the resulting system under four kinds of criteria:

1. Behavior of each source.
2. Behavior of the whole set of information sources.
3. Procedure for collecting information from the sources.
4. Procedure for processing the collected information.

Exemplary basic assumptions concerning them are listed below.

2.2.1. Behavior of each source

i) Scope of information provided by a source:

   (a) It provides information about all propositions (*complete knowledge*), i.e., assigns either 0 or 1 to each formula.

[3] Especially in the literature on bilattices. See e.g. [11, 10].

(b) It provides information about some propositions only (*partial knowledge*), i.e., assigns either 0 or 1 to some formulas only (with no particular logical restrictions).

(c) It provides information only about (some/all) atomic propositions (*partial/complete atomic knowledge*).

ii) Logical characteristics of a source:
The assignment of values by a source can be restricted by certain logical constraints. For example, we could demand that:

(a) For any formulas A, B such that $A \sim B$ (where $\sim$ denotes classical equivalence), each source should assign the same value to A and B. Instead of classical equivalence, other types of logical equivalence, more plausible from the implementation viewpoint, can also be considered here.

(b) The sources should be *classically coherent*: i.e., the partial valuation provided by each of the sources should be extendable to a full classical valuation.

(c) The sources should be *classically closed*, meaning that if φ classically follows from Γ, then any source which assigns 1 to all formulas in Γ should assign 1 to φ too, and that 1 (0) is assigned to $\neg\varphi$ iff 0 (1) is assigned to φ.

2.2.2. Behavior of the whole set of information sources

We may assume that, e.g.:

1. For each atomic proposition F, there is at least one source which provides information about F.

2. For an arbitrary proposition F, there is at least one source which provides information about F.

2.2.3. Procedure for collecting information from the sources

The processor can use various strategies in combining information from the sources. Thus it can accept a formula ψ as *true* (*false*) whenever:

Existential strategy: At least one source assigns ψ the value 1 (0). Note that in this case there is a possibility of assigning both 1 and 0 to the same formula. In such a case, the processor uses the truth value $\top$, which has no counterpart among those used by the sources.

Universal strategy: All the sources assign ψ the value 1 (0). Note that in this case the processor might assign no value to a formula even if its sources are of the "know all" type; then the processor might use $\perp$ even if the sources do not (implicitly) use that value.

Unanimous voting strategy: Some sources assign ψ the value 1 (0), and no source assigns 0 (1). This amounts to the *universal* policy, but with "all sources" applies only to the sources which give a definite answer.

Preferred sources strategies: Each of the three preceding strategies can be applied using a preferred set of sources (determined by ψ) rather than the whole set of sources.

2.2.4. Procedure for information processing

After collecting the direct information from the sources, the processor processes that information to define its own valuation d of formulas in L_C. We assume that during that stage the processor derives from the above direct information at least the most basic implicit information dictated by the truth tables of classical logic applied (wherever possible) in both directions. By this we mean that if a truth-value a assigned by the processor to some formula φ is possible according to some truth table of classical logic only if another (*single*) formula ψ is assigned the value b, then the processor assigns b to ψ. For example: if φ is assigned 1 then $\varphi \vee \psi$ is also assigned 1, while if $\varphi \vee \psi$ is assigned 0 then both φ and ψ are assigned 0. This might again lead to 0 and 1 being both assigned to the same formula. A stronger possible assumption (not investigated here) on the processor's procedure for information processing is that it fully respects everything dictated by the truth tables of classical logic. For example: If ψ is assigned 0, and $\varphi \vee \psi$ is assigned 1, then φ is assigned 1.

2.3. Formal definitions

The source-processor framework is formalized as follows: [4]

DEFINITION 2.1. Let $\mathcal{A}$ and $\mathcal{F}$ be the set of all atomic formulas and the set of all formulas of the language L_C of propositional classical logic, respectively.

- By a *source valuation* we mean a partial function $s : \mathcal{F} \to \{0, 1\}$.
- By a *processor valuation* we mean a function $v : \mathcal{F} \to \mathcal{P}(\{0, 1\})$.

[4] This section has benefited from discussions with David Makinson.

- By a *source-processor structure* we mean a tuple $\mathcal{S} = \langle S, g, d\rangle$, where S is a non-empty set of source valuations, g is an arbitrary processor valuation, and d is a processor valuation satisfying the following conditions:

  (d0) $g(\varphi) \subseteq d(\varphi)$ for every formula φ;

  (d1) $0 \in d(\neg\varphi)$ iff $1 \in d(\varphi)$;

  (d2) $1 \in d(\neg\varphi)$ iff $0 \in d(\varphi)$;

  (d3) $1 \in d(\varphi \vee \psi)$ if $1 \in d(\varphi)$ or $1 \in d(\psi)$;

  (d4) $0 \in d(\varphi \vee \psi)$ iff $0 \in d(\varphi)$ and $0 \in d(\psi)$;

  (d5) $1 \in d(\varphi \wedge \psi)$ iff $1 \in d(\varphi)$ and $1 \in d(\psi)$;

  (d6) $0 \in d(\varphi \wedge \psi)$ if $0 \in d(\varphi)$ or $0 \in d(\psi)$.

  (d1)–(d6) are called the *standard integrity conditions* for L_C.

NOTE 2.1. (d1)–(d6) are the rules which correspond to our above minimal assumption concerning the processor's procedure for information processing. Note that the converses of (d3) and (d6) do *not* hold, because if the sources might provide information about complex formulas, the processor might e.g. be informed that $\varphi \vee \psi$ is true without being told that either φ or ψ is true.

DEFINITION 2.2. A source-processor structure $\langle S, g, d\rangle$ for L_C is called *standard*, if d is the minimal processor valuation satisfying conditions (d0)–(d6).

Each source-processor structure $\mathcal{S} = \langle S, g, d\rangle$ can be seen as a representation of an instance I of the source-processor framework defined informally in Subsection 2.1, with the two being related as follows:

- S is the set of source valuations defined by the sources present in the instance I, where, for each $s \in S$, $s(\varphi) = 1$ iff φ is true according to source s, and $s(\varphi) = 0$ iff φ is false according to source s;

- g represents the global information collected by the processor directly from the sources, i.e., $1 \in g(\varphi)$ (resp. $0 \in g(\varphi)$) iff, after information collecting, φ is accepted by the processor as true (resp. false);

- d represents the information derived by the processor from g during the information processing stage, i.e., $1 \in d(\varphi)$ (resp. $0 \in d(\varphi)$) iff, after processing the global information in g, the processor concludes that φ is true (resp. false).

From the viewpoint of the information collecting strategy, in this paper we consider the following two basic types of source-processor structures:

Definition 2.3. A source-processor structure $\mathcal{S} = \langle S, g, d \rangle$ is called:

- *existential*, iff for any $\varphi \in \mathcal{F}$,

$$1 \in g(\varphi) \text{ iff } \exists\, s \in S. s(\varphi) = 1 \quad \text{and} \quad 0 \in g(\varphi) \text{ iff } \exists\, s \in S. s(\varphi) = 0$$

- *universal* iff for any $\varphi \in \mathcal{F}$,

$$1 \in g(\varphi) \text{ iff } \forall\, s \in S. s(\varphi) = 1 \quad \text{and} \quad 0 \in g(\varphi) \text{ iff } \forall\, s \in S. s(\varphi) = 0$$

Next we turn to the *logics* induced by source-processor structures. Each such structure $\mathcal{S} = \langle S, g, d \rangle$ naturally generates a satisfaction relation on the formulas in $\mathcal{F}$ (determined by the final processor valuation d):

Definition 2.4. Let $\mathcal{S} = \langle S, g, d \rangle$ be a source-processor structure. $\mathcal{S}$ *satisfies* (is a *model* of) a formula $\varphi \in \mathcal{F}$, in symbols $\models_{\mathcal{S}} \varphi$, iff $1 \in d(\varphi)$.

Accordingly, each source-processor structure or, more generally, a class of source-processor structures, induces the corresponding consequence relation:

Definition 2.5. Let $\mathcal{J}$ be a class of source-processor structures. The *consequence relation* induced by $\mathcal{J}$ is the relation $\vdash_{\mathcal{J}}$ on $\mathcal{P}(\mathcal{F}) \times \mathcal{F}$ such that $T \vdash_{\mathcal{J}} \varphi$ if every $\mathcal{S} \in \mathcal{J}$ which is a model of T is also a model of φ.

2.4. The need for using sequents

In the context of source-processor structures, the expressive power of *formulas* of L_C is too weak. Thus there is no way to express that a certain formula φ is *not true* (meaning that $1 \notin d(\varphi)$). In the classical framework this is expressed by $\neg\varphi$, but in the present context the truth of $\neg\varphi$ means only that $0 \in d(\varphi)$, and this neither implies nor is implied by $1 \notin d(\varphi)$. Similarly, there is no way to express *disjunctive knowledge* of the form "one of the sentences φ and ψ is known to be true" (meaning that either $1 \in d(\varphi)$ or $1 \in d(\psi)$), because it is possible that $1 \in d(\varphi \vee \psi)$ but neither $1 \in d(\varphi)$ nor $1 \in d(\psi)$.

These problems can be overcome by using Gentzen-type *sequents* both for expressing knowledge and for reasoning about it. The idea is that given a source-processor structure $\langle S, g, d \rangle$, a sequent $\varphi_1, \ldots, \varphi_n \Rightarrow \psi_1, \ldots, \psi_k$ expresses the information that either $1 \notin d(\varphi_1)$, or $1 \notin d(\varphi_2)$, or $\ldots$ or $1 \notin d(\varphi_n)$, or $1 \in d(\psi_1)$, or $\ldots$ or $1 \in d(\psi_k)$. The notions of model and satisfaction, and the corresponding consequence relations are then extended to the language of sequents in a straightforward way:

Definition 2.6.

- A *sequent* is a structure of the form $\Gamma \Rightarrow \Delta$, where Γ and Δ are finite *sets* of formulas. We denote by Seq the set of all sequents in the language L_C.
- Let $\mathcal{S} = \langle S, g, d \rangle$ be a source-processor structure. $\mathcal{S}$ *satisfies* (is a *model* of) a sequent $\Sigma = \Gamma \Rightarrow \Delta$, in symbols $\models_{\mathcal{S}} \Sigma$, iff either $\mathcal{S}$ is a model of some formula in Δ, or it is not a model of some formula in Γ.
- Let $\mathcal{J}$ be a class of source-processor structures. The *sequent consequence relation* induced by $\mathcal{J}$ is the relation $\vdash_{\mathcal{J}}$ on $\mathcal{P}(Seq) \times Seq$ s.t. $Q \vdash_{\mathcal{J}} \Sigma$ if every $\mathcal{S} \in \mathcal{J}$ which is a model of Q is also a model of Σ.

Note 2.2. It can easily be seen that if Γ is a finite subset of $\mathcal{F}$, and φ is a formula in $\mathcal{F}$, then $\Gamma \vdash_{\mathcal{J}} \varphi$ iff $\vdash_{\mathcal{J}} \Gamma \Rightarrow \varphi$ iff $\{\Rightarrow \psi \mid \psi \in \Gamma\} \vdash_{\mathcal{J}} \Rightarrow \varphi$. Hence the sequent consequence relation $\vdash_{\mathcal{J}}$ can be seen as an extension of the formula consequence relation $\vdash_{\mathcal{J}}$ defined above (Definition 2.5). This justifies the use of the same symbol to denote both.

Note 2.3. Given a source-processor structure $\langle S, g, d \rangle$ and a formula φ, every known basic fact about $d(\varphi)$ can be expressed by sequents as follows:

- $1 \in d(\varphi)$ iff $\models_{\mathcal{S}} \Rightarrow \varphi$
- $1 \notin d(\varphi)$ iff $\models_{\mathcal{S}} \varphi \Rightarrow$
- $0 \in d(\varphi)$ iff $\models_{\mathcal{S}} \Rightarrow \neg\varphi$
- $0 \notin d(\varphi)$ iff $\models_{\mathcal{S}} \neg\varphi \Rightarrow$

One corrolary of this fact is that, given a class of source-processor structures $\mathcal{J}$, a formula φ follows from a set T of formulas according to the strong semantics (see Subsection 2.1) iff both $G(T) \vdash_{\mathcal{J}} \Rightarrow \varphi$ and $G(T) \vdash_{\mathcal{J}} \neg\varphi \Rightarrow$, where $G(T) = \{\Rightarrow \psi \mid \psi \in T\} \cup \{\neg\psi \Rightarrow \mid \psi \in T\}$. Hence the consequence relation induced by the strong semantics can be simulated by the weak one investigated in this paper.

3. Existential strategy for standard structures

In this section we assume that the existential strategy is adopted, and investigate under this assumption certain basic variants of standard source-processor structures for L_C (shortly referred to as "standard structures"). We shall consider the following four basic scenarios — the first two corresponding to well-known logics, and the other two new.

I **Dunn-Belnap's logic**: the sources provide information about atomic formulas only, but not necessarily about all of them;

II **D'Ottaviano and da Costa's basic paraconsistent logic:** Like the preceding case, but the sources taken together are required to provide some information about all atomic formulas.

III **The most general source-processor logic:** The sources provide information about arbitrary formulas, both atomic and composed ones, but not necessarily about all of them.

IV **General source-processor logic with complete information:** As the preceding case, but the sources taken together are required to provide some information about all atomic formulas.

To handle the new logics arising out of the last two cases we need the notion of non-deterministic matrices (shortly: Nmatrices) introduced in [3, 4]:

Definition 3.1.

1. A *non-deterministic matrix* (*Nmatrix* for short) for a propositional language $\mathcal{L}$ is a tuple $\mathcal{M} = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where:

   (a) $\mathcal{V}$ is a non-empty set of *truth values*.

   (b) $\mathcal{D}$ is a non-empty proper subset of $\mathcal{V}$.

   (c) For every n-ary connective $\diamond$ of $\mathcal{L}$, $\mathcal{O}$ includes a corresponding n-ary function $\widetilde{\diamond}$ from $\mathcal{V}^n$ to $2^{\mathcal{V}} - \{\emptyset\}$.

2. Let $\mathcal{W}$ be the set of formulas of $\mathcal{L}$. A *(legal) valuation* in an Nmatrix $\mathcal{M}$ is a function $v : \mathcal{W} \to \mathcal{V}$ that satisfies the following condition for every n-ary connective $\diamond$ of $\mathcal{L}$ and $\psi_1, \ldots, \psi_n \in \mathcal{L}$:

$$v(\diamond(\psi_1, \ldots, \psi_n)) \in \widetilde{\diamond}(v(\psi_1), \ldots, v(\psi_n))$$

3. A valuation v in an Nmatrix $\mathcal{M}$ is:

   - a *model* of (*satisfies*) a formula ψ in $\mathcal{M}$ ($v \models^{\mathcal{M}} \psi$) if $v(\psi) \in \mathcal{D}$.
   - a *model* of a set $T \subseteq \mathcal{F}$ in $\mathcal{M}$ ($v \models^{\mathcal{M}} T$) if $v \models^{\mathcal{M}} \psi$ for all $\psi \in T$.
   - *a model* of a sequent $\Sigma = \Gamma \Rightarrow \Delta$ ($v \models^{\mathcal{M}} \Sigma$) iff either $v \models^{\mathcal{M}} \psi$ for some $\psi \in \Delta$, or $v \not\models^{\mathcal{M}} \psi$ for some $\psi \in \Gamma$.

4. The *formula consequence relation* induced by the Nmatrix $\mathcal{M}$ (denoted by $\vdash_{\mathcal{M}}$) is defined by: $T \vdash_{\mathcal{M}} \varphi$ if every model of T in $\mathcal{M}$ is also a model of φ. The corresponding *sequent consequence relation* induced by $\mathcal{M}$ (also denoted by $\vdash_{\mathcal{M}}$) is defined similarly (compare Definition 2.6).

NOTE 3.4. Again, we have (see Note 2.2) that for every Nmatrix $\mathcal{M}$, every finite subset $\Gamma \subseteq \mathcal{W}$, and every formula $\varphi \in \mathcal{W}$, $\Gamma \vdash_{\mathcal{M}} \varphi$ iff $\vdash_{\mathcal{M}} \Gamma \Rightarrow \varphi$ iff $\{\Rightarrow \psi \mid \psi \in \Gamma\} \vdash_{\mathcal{M}} \Rightarrow \varphi$.

NOTE 3.5. An ordinary (deterministic) multiple-valued matrix can be seen as a special case of an Nmatrix, in which the interpretations of the connectives always return singletons.

3.1. Dunn-Belnap's logic

The first case we examine is when the sources provide (possibly incomplete) information about atomic formulas only, and the processor uses the existential strategy to combine the direct information from the sources and obtain the global information g. We shall show that the logic induced by this class of structures coincides with Dunn-Belnap's four-valued logic ([15, 7, 6]). This logic is induced by the following four-valued matrix $\mathcal{M}_B^4 = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where $\mathcal{V} = \{\mathbf{f}, \bot, \top, \mathbf{t}\}, \mathcal{D} = \{\top, \mathbf{t}\}, \mathcal{O} = \{\tilde{\neg}, \tilde{\vee}, \tilde{\wedge}\}$, and the interpretations of the connectives are given by the following tables:

$\tilde{\vee}$	$\mathbf{f}$	$\bot$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\bot$	$\top$	$\mathbf{t}$
$\bot$	$\bot$	$\bot$	$\mathbf{t}$	$\mathbf{t}$
$\top$	$\top$	$\mathbf{t}$	$\top$	$\mathbf{t}$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$

$\tilde{\wedge}$	$\mathbf{f}$	$\bot$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
$\bot$	$\mathbf{f}$	$\bot$	$\mathbf{f}$	$\bot$
$\top$	$\mathbf{f}$	$\mathbf{f}$	$\top$	$\top$
$\mathbf{t}$	$\mathbf{f}$	$\bot$	$\top$	$\mathbf{t}$

$\tilde{\neg}$	$\mathbf{f}$	$\bot$	$\top$	$\mathbf{t}$
	$\mathbf{t}$	$\bot$	$\top$	$\mathbf{f}$

Recalling what $\mathbf{f}, \bot, \top, \mathbf{t}$ stand for, these tables are best understood using the following well-known equivalent representation of $\mathcal{M}_B^4$:

DEFINITION 3.2. Let $v_0 : \mathcal{A} \to \mathcal{P}(\{0,1\})$.

- The Belnap extension of v_0 is the function $v : \mathcal{F} \to \mathcal{P}(\{0,1\})$ defined inductively as follows:

  (b0) $v(p) = v_0(p)$ for $p \in \mathcal{A}$;

  (b1) If $1 \in v(\varphi)$, then $0 \in v(\neg\varphi)$;

(b2) If $0 \in v(\varphi)$, then $1 \in v(\neg\varphi)$;

(b3) If $1 \in v(\varphi)$ or $1 \in v(\psi)$, then $1 \in v(\varphi \vee \psi)$;

(b4) If $0 \in v(\varphi)$ and $0 \in v(\psi)$, then $0 \in v(\varphi \vee \psi)$;

(b5) If $1 \in v(\varphi)$ and $1 \in v(\psi)$, then $1 \in v(\varphi \wedge \psi)$;

(b6) If $0 \in v(\varphi)$ or $0 \in v(\psi)$, then $0 \in v(\varphi \wedge \psi)$.

- A Belnap valuation is a function $v : \mathcal{F} \to \mathcal{P}(\{0,1\})$ being a Belnap extension of some valuation $v_0 : \mathcal{A} \to \mathcal{P}(\{0,1\})$. The set of all Belnap valuations will be denoted by $\mathcal{V}(\mathcal{M}_B^4)$.

- A *Belnap model* of $\Gamma \subseteq \mathcal{F}$ is any $v \in \mathcal{V}(\mathcal{M}_B^4)$ such that $\forall\varphi \in \Gamma.1 \in v(\varphi)$ (note this is equivalent to taking $\mathbf{t}$ and $\top$ as the designated values).

- The Belnap formula consequence relation and the Belnap sequent consequence relation (both denoted in the sequel by $\vdash_{\mathcal{M}_B^4}$) are defined from the notion of a Belnap model in the usual way (see definitions 2.5, 2.6, and the end of Definition 3.1. Since ordinary matrices are a special type of Nmatrices, $\vdash_{\mathcal{M}_B^4}$ is actually an instance of the latter).

LEMMA 3.6. Each Belnap valuation satisfies the converses of (b1)–(b6).

PROOF. In the inductive process of extending $v_0 : \mathcal{A} \to \mathcal{P}(\{0,1\})$ to a Belnap valuation v, the inclusion of 0 or 1 in the value of a composed formula can be due to exactly one of the rules $(b1)$–$(b6)$. Hence the result. ■

DEFINITION 3.3. Let $\mathcal{EA}$ denote the class of standard source-processor structures $\mathcal{S} = \langle S, g, d\rangle$, where each $s \in S$ is undefined outside $\mathcal{A}$ (i.e., the sources provide information about atomic formulas only), and the processor uses the existential strategy to obtain g out of the valuations in S.

LEMMA 3.7. We have the following correspondence between $\mathcal{M}_B^4$ and $\mathcal{EA}$:

(1) $\forall v \in \mathcal{V}(\mathcal{M}_B^4)\ \exists\mathcal{S} \in \mathcal{EA}\ \exists S\ \exists g.\mathcal{S} = \langle S, g, v\rangle$

(2) $\forall\mathcal{S} \in \mathcal{EA}\ \forall S\ \forall g\ \forall d.\mathcal{S} = \langle S, g, d\rangle \to d \in \mathcal{V}(\mathcal{M}_B^4)$

PROOF. Ad (1): Define $\mathcal{S} = \langle S_v, g_v, v\rangle$, where $S_v = \{s_v^0, s_v^1\}$, and for $\varphi \in \mathcal{F}$:

(i) $s_v^i(\varphi) = i$ if $\varphi \in \mathcal{A}$ and $i \in v(\varphi)$, undefined otherwise

(ii) $g_v(\varphi) = v(\varphi)$ if $\varphi \in \mathcal{A}$, $\emptyset$ otherwise.

Since $v \in \mathcal{V}(\mathcal{M}_B^4)$, by (ii), v is the minimal extension of g_v which satisfies (b1)-(b6). Lemma 3.6 implies that it is also the minimal extension of g_v which satisfies (d1)-(d6). Hence $\mathcal{S}$ is a standard structure. It is easy to see that $i \in g_v(\varphi) \Leftrightarrow \exists s \in S.i = s(\varphi)$ (for $\varphi \in \mathcal{A}$ and $i \in \{0,1\}$). Since $g_v(\varphi)$ is nonempty only for $\varphi \in \mathcal{A}$, this implies that $\mathcal{S}$ is existential. Hence $\mathcal{S} \in \mathcal{EA}$.

Ad (2): Assume $\mathcal{S} = \langle S, g, d\rangle \in \mathcal{EA}$. Since $s(\varphi)$ is undefined for $\varphi \notin \mathcal{A}$, also $g(\varphi) = \emptyset$ for $\varphi \notin \mathcal{A}$. Accordingly, g can be viewed as a function from $\mathcal{A}$ to $\mathcal{P}(\{0,1\})$. As $\mathcal{S}$ is a standard structure, d is the minimal processor valuation which satisfies conditions (d0)–(d6). Let v be the Belnap extension of g. Then $v(p) = g(p)$ for $p \in \mathcal{A}$, which in view of $g(\varphi) = \emptyset$ for $\varphi \notin \mathcal{A}$ implies that $g(\varphi) \subseteq v(\varphi)$ for every $\varphi \in \mathcal{F}$. This and Lemma 3.6 imply that v satisfies conditions (d0)–(d6). Hence by the minimality of d we must have $d \subseteq v$. However, since $d(p) = v(p)$ for atomic p, and d satisfies conditions (b1)–(b6), the converse implication must also hold by definition 3.2. Thus $d = v$, and d is a Belnap valuation. ∎

Proposition 3.8. $\vdash_{\mathcal{EA}} = \vdash_{\mathcal{M}_B^4}$.

Proof. This is immediate from Lemma 3.7. ∎

3.2. D'Ottaviano and da Costa's basic paraconsistent logic

Case II refers to the situation when the sources provide complete information about atomic formulas, and no information about complex ones. Since we assume here the existential strategy, this implies that $g(p) \neq \perp$ for every atomic formula p. By induction on the complexity of formulas (using (d1)–(d6)), one can show that $d(\varphi) \neq \perp$ for every formula. Accordingly, the difference between this case and the previous one is that this time we have to do with just the three logical values $\mathbf{f}, \top, \mathbf{t}$. Therefore, this case is represented by the ordinary *three*-valued submatrix $\mathcal{M}_B^3 = \langle \mathcal{V}, \mathcal{D}, \mathcal{O}\rangle$ of $\mathcal{M}_B^4$, with $\mathcal{V} = \{\mathbf{f}, \top, \mathbf{t}\}, \mathcal{D} = \{\top, \mathbf{t}\}, \mathcal{O} = \{\widetilde{\neg}, \widetilde{\vee}, \widetilde{\wedge}\}$, and the deterministic interpretations of the connectives given by:

$\widetilde{\vee}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
$\top$	$\top$	$\top$	$\mathbf{t}$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$

$\widetilde{\wedge}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
$\top$	$\mathbf{f}$	$\top$	$\top$
$\mathbf{t}$	$\mathbf{f}$	$\top$	$\mathbf{t}$

$\widetilde{\neg}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
	$\mathbf{t}$	$\top$	$\mathbf{f}$

This matrix corresponds to the $\{\neg, \vee, \wedge\}$-fragment of D'Ottaviano and da Costa's logic J_3 ([12, 13, 1, 14]).

PROPOSITION 3.9. Let $\mathcal{ECA}$ denote the class of standard source-processor structures $\mathcal{S} = \langle S, g, d\rangle$, where each $s \in S$ is undefined outside $\mathcal{A}$, the processor uses the existential strategy to obtain g out of the valuations in S, and $g(\varphi) \neq \perp$ for every $\varphi \in \mathcal{A}$ (i.e., the sources taken together provide information about all atomic formulae). Then $\vdash_{\mathcal{ECA}} = \vdash_{\mathcal{M}^3_B}$.

The proof is similar to that of Proposition 3.8, so we omit it.

3.3. The most general source-processor logic

Now we shall discuss the most general case (III), when the sources can provide information about arbitrary formulas, including the complex ones, but that information may not cover all formulas, i.e. it may be incomplete. It is easy to see that in this case the conditions (d1)–(d6) from Subsection 2.3, obeyed by the processor in assigning values to formulas, imply that the presented setup can be described by the four-valued *Nmatrix* $\mathcal{M}^4_I = \langle \mathcal{V}, \mathcal{D}, \mathcal{O}\rangle$, where $\mathcal{V} = \{\mathbf{f}, \perp, \top, \mathbf{t}\}, \mathcal{D} = \{\top, \mathbf{t}\}, \mathcal{O} = \{\widetilde{\neg}, \widetilde{\vee}, \widetilde{\wedge}\}$, and the non-deterministic interpretations of the connectives are given by the following tables:

$\widetilde{\vee}$	$\mathbf{f}$	$\perp$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\{\mathbf{f}, \top\}$	$\{\mathbf{t}, \perp\}$	$\{\top\}$	$\{\mathbf{t}\}$
$\perp$	$\{\mathbf{t}, \perp\}$	$\{\mathbf{t}, \perp\}$	$\{\mathbf{t}\}$	$\{\mathbf{t}\}$
$\top$	$\{\top\}$	$\{\mathbf{t}\}$	$\{\top\}$	$\{\mathbf{t}\}$
$\mathbf{t}$	$\{\mathbf{t}\}$	$\{\mathbf{t}\}$	$\{\mathbf{t}\}$	$\{\mathbf{t}\}$

$\widetilde{\wedge}$	$\mathbf{f}$	$\perp$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\{\mathbf{f}\}$	$\{\mathbf{f}\}$	$\{\mathbf{f}\}$	$\{\mathbf{f}\}$
$\perp$	$\{\mathbf{f}\}$	$\{\mathbf{f}, \perp\}$	$\{\mathbf{f}\}$	$\{\mathbf{f}, \perp\}$
$\top$	$\{\mathbf{f}\}$	$\{\mathbf{f}\}$	$\{\top\}$	$\{\top\}$
$\mathbf{t}$	$\{\mathbf{f}\}$	$\{\mathbf{f}, \perp\}$	$\{\top\}$	$\{\mathbf{t}, \top\}$

$\widetilde{\neg}$	$\mathbf{f}$	$\perp$	$\top$	$\mathbf{t}$
	$\mathbf{t}$	$\perp$	$\top$	$\mathbf{f}$

Intuitively, any legal valuation of $\mathcal{M}^4_I$ represents possible information about values of formulas in a standard source-processor structure. To better understand this, let examine the rather surprising entry in the table for $\widetilde{\vee}$ saying that $\perp \widetilde{\vee} \perp = \{\mathbf{t}, \perp\}$. Suppose that in a source-processor structure $\mathcal{S} = \langle S, g, d\rangle$ we have $d(\varphi) = d(\psi) = \perp$. Then $0 \notin d(\varphi)$ and $0 \notin d(\psi)$, so by (d4) $0 \notin d(\varphi \vee \psi)$. Hence two cases are possible. If also $1 \notin d(\varphi \vee \psi)$ (which is what one might expect in case $1 \notin d(\varphi)$ and $1 \notin d(\psi)$), then $d(\varphi \vee \psi) = \perp$. If $1 \in d(\varphi \vee \psi)$ (e.g. because there is a source s such that $s(\varphi \vee \psi) = 1$, in which case $1 \in g(\varphi \vee \psi)$ in view of the existential globalisation strategy used by the processor), then $d(\varphi \vee \psi) = \mathbf{t}$. This justifies the two options included in this table entry; some other entries are explained in [5].

PROPOSITION 3.10. Let $\mathcal{E}$ denote the class of standard source-processor structures where the processor uses the existential strategy. Then $\vdash_{\mathcal{E}} = \vdash_{\mathcal{M}_I^4}$.

PROOF. Let $\mathcal{V}(\mathcal{M}_I^4)$ be the set of legal valuations of $\mathcal{M}_I^4$. Obviously, v is in $\mathcal{V}(\mathcal{M}_I^4)$ iff it satisfies conditions (d1)-(d6). It follows that

(1) $\forall \mathcal{S} \in \mathcal{E}\ \forall S\ \forall g\ \forall d.\ \mathcal{S} = \langle S, g, d \rangle \to d \in \mathcal{V}(\mathcal{M}_I^4)$

Now assume that $v \in \mathcal{V}(\mathcal{M}_I^4)$. For $i = 0, 1$ and for every $\varphi \in \mathcal{F}$, let $s_v^i(\varphi) = i$ if $i \in v(\varphi)$, and undefined otherwise. It is easy to see that $\mathcal{S} = \langle \{s_v^0, s_v^1\}, v, v \rangle$ is an element of $\mathcal{E}$. Hence:

(2) $\forall v \in \mathcal{V}(\mathcal{M}_I^4)\ \exists \mathcal{S} \in \mathcal{E}\ \exists S\ \exists g. \mathcal{S} = \langle S, g, v \rangle$

The theorem is now immediate from (1) and (2). ∎

3.4. General source-processor logic with complete information

The last case is when the sources provide complete information about all atomic formulas (but they may provide information, not necessarily complete one, about other formulas too). Thus for any atomic formula p of L_C, some source in S must say either that p is true or that p is false. Like in Subsection 3.2, one can easily prove by induction that under this condition no formula is given the value $\perp$. Thus in this case too only three truth-values are employed. However, this time the scenario gives rise to a logic based on a three-valued *Nmatrix*. This is the Nmatrix $\mathcal{M}_I^3 = \langle \mathcal{V}, \mathcal{D}, \mathcal{O} \rangle$, where $\mathcal{V} = \{\mathbf{f}, \top, \mathbf{t}\}, \mathcal{D} = \{\top, \mathbf{t}\}, \mathcal{O} = \{\tilde{\neg}, \tilde{\vee}, \tilde{\wedge}\}$, and the non-deterministic interpretations of the connectives are given by:

$\tilde{\neg}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
	$\mathbf{t}$	$\top$	$\mathbf{f}$

$\tilde{\vee}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\{\mathbf{f}, \top\}$	$\{\top\}$	$\{\mathbf{t}\}$
$\top$	$\{\top\}$	$\{\top\}$	$\{\mathbf{t}\}$
$\mathbf{t}$	$\{\mathbf{t}\}$	$\{\mathbf{t}\}$	$\{\mathbf{t}\}$

$\tilde{\wedge}$	$\mathbf{f}$	$\top$	$\mathbf{t}$
$\mathbf{f}$	$\{\mathbf{f}\}$	$\{\mathbf{f}\}$	$\{\mathbf{f}\}$
$\top$	$\{\mathbf{f}\}$	$\{\top\}$	$\{\top\}$
$\mathbf{t}$	$\{\mathbf{f}\}$	$\{\top\}$	$\{\mathbf{t}, \top\}$

We leave the proof of the following easy proposition to the reader:

PROPOSITION 3.11. Let $\mathcal{EC}$ denote the class of standard source-processor structures where the sources taken together provide some information about every atomic formula, and the processor uses the existential strategy. Then $\vdash_{\mathcal{EC}} = \vdash_{\mathcal{M}_I^3}$.

4. The universal strategy

In this section we discuss in brief the case in which the processor applies the universal strategy in collecting information from the sources. Note first that if there are at least two sources then $g(\varphi)$ may be $\perp$ in this case, even if all the sources are of the "know all" type (because the processor will assign neither 0 nor 1 to a formula φ which is assigned different values by two sources). On the other hand, it is obvious that with the universal strategy $g(\varphi) \neq \top$ for every formula φ. Without further integrity constraints, this is not necessarily true for the standard extension d of g. One such plausible constraint is that the sources should all be *classically coherent* (see Subsection 2.2.1). Another is again that the sources provide information about atomic formulae only. It is easy to see that the resulting logic in the latter case is that induced by the famous 3-valued matrix $\mathcal{M}^3_K$ of Kleene:

$\widetilde{\vee}$	$\mathbf{f}$	$\perp$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\perp$	$\mathbf{t}$
$\perp$	$\perp$	$\perp$	$\mathbf{t}$
$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$	$\mathbf{t}$

$\widetilde{\wedge}$	$\mathbf{f}$	$\perp$	$\mathbf{t}$
$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$	$\mathbf{f}$
$\perp$	$\mathbf{f}$	$\perp$	$\perp$
$\mathbf{t}$	$\mathbf{f}$	$\perp$	$\mathbf{t}$

$\widetilde{\neg}$	$\mathbf{f}$	$\perp$	$\mathbf{t}$
	$\mathbf{t}$	$\perp$	$\mathbf{f}$

PROPOSITION 4.12. Let $\mathcal{AA}$ be the class of standard source-processor structures $\mathcal{S} = \langle S, g, d\rangle$, where each $s \in S$ is undefined outside $\mathcal{A}$ and the processor uses the universal strategy to obtain g out of the valuations in S. Then $\vdash_{\mathcal{AA}} = \vdash_{\mathcal{M}^3_K}$.

PROOF. Similar to the proof of Proposition 3.8. ■

5. Proof systems for the existential strategy

Now we will proceed to develop proof systems for the four logics discussed in the preceding section. As we explained in Subsection 2.4, to compensate for the weakness of the language, the systems we will provide will be strongly sound and complete sequent calculi.

5.1. The most general source-processor logic

We begin with the proof system for the most general case, from which we will later derive the proof systems for the remaining cases.

DEFINITION 5.1. Let $\mathcal{C}^4_I$ be the sequent calculus defined as follows:

Axioms: $\varphi \Rightarrow \varphi$

Structural inference rules : Weakening, Cut.

Logical inference rules:

$$(\neg\neg \Rightarrow)\ \frac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma, \neg\neg\varphi \Rightarrow \Delta} \qquad (\Rightarrow \neg\neg)\ \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \neg\neg\varphi}$$

$$(\Rightarrow \vee)\ \frac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}$$

$$(\neg\vee \Rightarrow)\ \frac{\Gamma, \neg\varphi, \neg\psi \Rightarrow \Delta}{\Gamma, \neg(\varphi \vee \psi) \Rightarrow \Delta} \qquad (\Rightarrow \neg\vee)\ \frac{\Gamma \Rightarrow \Delta, \neg\varphi \qquad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \vee \psi)}$$

$$(\wedge \Rightarrow)\ \frac{\Gamma, \varphi, \psi \Rightarrow \Delta,}{\Gamma, \varphi \wedge \psi \Rightarrow \Delta} \qquad (\Rightarrow \wedge)\ \frac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}$$

$$(\Rightarrow \neg\wedge)\ \frac{\Gamma \Rightarrow \Delta, \neg\varphi, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \wedge \psi)}$$

DEFINITION 5.2. Let $\underline{\mathcal{C}}_I^4$ be the calculus obtained from $\mathcal{C}_I^4$ by limiting the applications of the cut rule to formulas occurring in the premises of sequent derivations. In other words: If $S = \{\Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_n \Rightarrow \Delta_n\}$ then $S \vdash_{\underline{\mathcal{C}}_I^4} \Sigma$ if there is a proof of Σ from S in $\mathcal{C}_I^4$ in which all cuts are on formulas from $\bigcup_{i=1}^n (\Gamma_i \cup \Delta_i)$ (in particular: $\vdash_{\underline{\mathcal{C}}_I^4} \Sigma$ iff Σ has a cut-free proof in $\mathcal{C}_I^4$).

THEOREM 5.13. The calculus $\underline{\mathcal{C}}_I^4$ is finitely strongly sound and complete for $\vdash_{\mathcal{M}_I^4}$, i.e., for any finite set of sequents $S \subseteq Seq$ and any sequent $\Sigma \in Seq$, $S \vdash_{\mathcal{M}_I^4} \Sigma$ iff $S \vdash_{\underline{\mathcal{C}}_I^4} \Sigma$.

PROOF. For simplicity, in what follows we drop the decorations on $\models$.

It is easy to see that $\mathcal{C}_I^4$ is strongly sound for $\vdash_{\mathcal{M}_I^4}$ (i.e. if $S \vdash_{\mathcal{C}_I^4} \Sigma$ then $S \vdash_{\mathcal{M}_I^4} \Sigma$). Hence it suffices to prove the strong completeness of $\underline{\mathcal{C}}_I^4$ for finite premise sets.

We argue by contradiction. Suppose that for a finite set of sequents S and a sequent $\Sigma_0 = \Gamma \Rightarrow \Delta$ we have $S \vdash_{\mathcal{M}_I^4} \Sigma_0$, but Σ_0 is not derivable from S in $\underline{\mathcal{C}}_I^4$. We shall construct a counter-valuation v such that $v \models S$ but $v \not\models \Sigma_0$.

Denote by $F(S)$ the set of all formulae belonging to at least one of the sides in some sequent in S. Then $F(S)$ is finite; assume it has l elements. Let

$\varphi_1, \varphi_2, \ldots, \varphi_l$ be an enumeration of formulae in $F(S)$. We shall now define a sequence of sequents $\Gamma_n \Rightarrow \Delta_n, n = 0, 1, \ldots, l$, such that, for $n = 0, 1, \ldots, l$:

(i) $\Gamma \subseteq \Gamma_n$, $\Delta \subseteq \Delta_n$

(ii) If $n \neq 0$ then $\varphi_n \in (\Gamma_n \cup \Delta_n)$.

(iii) $\Gamma_n \Rightarrow \Delta_n$ is not derivable from S in $\underline{\mathcal{C}}_I^4$.

The above sequences are defined inductively as follows:

- We put $\Gamma_0 = \Gamma, \Delta_0 = \Delta$. As by our assumption $\Gamma \Rightarrow \Delta$ is not derivable from S in $\underline{\mathcal{C}}_I^4$, (i)–(iii) above are satisfied for $n = 0$.
- Suppose $n \leq l-1$ and we have defined the sequents $\Gamma_i \Rightarrow \Delta_i$ satisfying conditions (i)–(iii) for $i \leq n$. Then the sequents $\Sigma_1 = \Gamma_n \Rightarrow \Delta_n, \varphi_{n+1}$ and $\Sigma_2 = \varphi_{n+1}, \Gamma_n \Rightarrow \Delta_n$ cannot be both derivable from S in $\underline{\mathcal{C}}_I^4$, since then $\Gamma_n \Rightarrow \Delta_n$ would be derivable from them by an allowed cut on the formula $\varphi_{n+1} \in S$. We take $\Gamma_{n+1} \Rightarrow \Delta_{n+1}$ to be Σ_1, if Σ_1 is not derivable, and Σ_2 otherwise. Then, obviously, from the inductive assumption it follows that the sequence $\Gamma_{n+1} \Rightarrow \Delta_{n+1}$ satisfies conditions (i)–(iii).

By induction, the whole sequence $\Gamma_n \Rightarrow \Delta_n, n = 0, 1, \ldots, l$, satisfies the desired conditions (i)–(iii). What is more, from the inductive construction we can see that $\Gamma_n \subseteq \Gamma_{n+1}, \Delta_n \subseteq \Delta_{n+1}$ for $n = 1, 2, \ldots, l-1$.

Let $\Gamma^* \Rightarrow \Delta^*$ be the extension of $\Gamma_l \Rightarrow \Delta_l$ to a saturated sequent, i.e., a minimal sequent containing $\Gamma_l \Rightarrow \Delta_l$ and closed under the logical rules in $\underline{\mathcal{C}}_I^4$ applied backwards. Then we can easily see that:

(i) $\Gamma \subseteq \Gamma^*, \Delta \subseteq \Delta^*$;

(ii) $F(S) \subseteq \Gamma^* \cup \Delta^*$;

(iii) $\Gamma^* \Rightarrow \Delta^*$ is saturated and it is not derivable from S in $\underline{\mathcal{C}}_I^4$.

Now we define the valuation v as follows:

- For any atomic p:

  (v0) $1 \in v(p)$ iff $p \in \Gamma^*$, $0 \in v(p)$ iff $\neg p \in \Gamma^*$;

- For any formulas α, β:

  (v1) $1 \in v(\neg\alpha)$ iff $0 \in v(\alpha)$;

(v2) $0 \in v(\neg\alpha)$ iff $1 \in v(\alpha)$;

(v3) $1 \in v(\alpha \vee \beta)$ iff $1 \in v(\alpha)$ or $1 \in v(\beta)$ or $(\alpha \vee \beta) \in \Gamma^*$;

(v4) $0 \in v(\alpha \vee \beta)$ iff $0 \in v(\alpha)$ and $0 \in v(\beta)$;

(v5) $1 \in v(\alpha \wedge \beta)$ iff $1 \in v(\alpha)$ and $1 \in v(\beta)$;

(v6) $0 \in v(\alpha \wedge \beta)$ iff $0 \in v(\alpha)$ or $0 \in v(\beta)$ or $\neg(\alpha \wedge \beta) \in \Gamma^*$;

It can be easily checked, by considering the truth tables of the Nmatrix $\mathcal{M}_I^4$, that v defined as above is legal valuation for that Nmatrix. It remains to prove that v is indeed the desired counter-valuation, i.e., that:

(I) $v \models \Sigma$ for each $\Sigma \in S$; (II) $v \not\models (\Gamma \Rightarrow \Delta)$;

We start with (II). As $\Gamma \subseteq \Gamma^*, \Delta \subseteq \Delta^*$, then in order to prove (II) it suffices to prove that $v \not\models (\Gamma^* \Rightarrow \Delta^*)$. To this end, we have to show that:

(A) $v \models \gamma$ for each $\gamma \in \Gamma^*$; (B) $v \not\models \delta$ for each $\delta \in \Delta^*$

We argue by induction on the complexity of formulas.

Proof of (A):

- Assume γ is atomic. Then $1 \in v(\gamma)$ by (v0) in the definition of v, whence $v \models \gamma$.
- Assume $\gamma = \neg\gamma'$. This case splits in the following four subcases:

  $\gamma' = p$ (where p is atomic): Then $\neg p = \gamma \in \Gamma^*$, and by (v0) in the definition of v we have $0 \in v(p)$, whence by (v1) of that definition we get $1 \in v(\gamma)$;

  $\gamma' = \neg\alpha$**:** Then $\neg\neg\alpha = \gamma \in \Gamma^*$. As $\Gamma^* \Rightarrow \Delta^*$ is saturated, then by rule $(\neg\neg \Rightarrow)$ we have $\alpha \in \Gamma^*$, whence by the inductive assumption $1 \in v(\alpha)$, which in turn yields $1 \in v(\neg\neg\alpha) = v(\gamma)$ by applications of (v2) and (v1);

  $\gamma' = \alpha \vee \beta$**:** Then $\neg(\alpha \vee \beta) = \gamma \in \Gamma^*$. As the sequent $\Gamma^* \Rightarrow \Delta^*$ is saturated, then by rule $(\neg\vee \Rightarrow)$ we have $\neg\alpha, \neg\beta \in \Gamma^*$, whence by the inductive assumption $1 \in v(\neg\alpha), 1 \in v(\neg\beta)$ Thus $0 \in v(\alpha), 0 \in v(\beta)$ by (v1), whence $0 \in v(\alpha \vee \beta)$ by (v4), and finally $1 \in v(\neg(\alpha \vee \beta)) = v(\gamma)$ by (v1);

  $\gamma' = \alpha \wedge \beta$**:** Then $\neg(\alpha \wedge \beta) = \gamma \in \Gamma^*$, whence by (v6) $0 \in v(\alpha \wedge \beta)$, which yields $1 \in v(\neg(\alpha \wedge \beta)) = v(\gamma)$.

- Assume $\gamma = \gamma_1 \vee \gamma_2$. Then $\gamma_1 \vee \gamma_2 = \gamma \in \Gamma^*$, so by (v3) we have $1 \in v(\gamma_1 \vee \gamma_2) = v(\gamma)$.
- Assume $\gamma = \gamma_1 \wedge \gamma_2$. As $\gamma \in \Gamma^*$ and the sequent $\Gamma^* \Rightarrow \Delta^*$ is saturated, then by rule $(\wedge \Rightarrow)$ we have $\gamma_1, \gamma_2 \in \Gamma^*$, whence by the inductive assumption $1 \in v(\gamma_1), 1 \in v(\gamma_2)$, which yields $1 \in v(\gamma_1 \wedge \gamma_2) = v(\gamma)$ by (v5).

Proof of (B):

Assume first that $\delta = p$ where p is atomic. As $\delta \in \Delta^*$ and $\Gamma^* \Rightarrow \Delta^*$ is not derivable, then $p \notin \Gamma^*$, whence $1 \notin v(\delta)$ by (v0). In turn, if $\delta = \neg p$, then $\neg p \notin \Gamma^*$, for $\Gamma^* \Rightarrow \Delta^*$ is not derivable. Thus by (v0) $0 \notin v(p)$, whence by (v1) we have $1 \notin v(\neg p) = v(\delta)$.

The proof that (B) holds for δ's which are not literals is carried out by induction, following a single schema analogous to the proof of, e.g., the last case in (A). Each time, from the fact that $\Gamma^* \Rightarrow \Delta^*$ is saturated and from the right hand side rule of the sequent calculus corresponding to the given complex formula α we conclude that the appropriate component formulas of α must also be in Δ^*, whence by the inductive assumption they are not assigned 1 by v. From the latter we deduce that $1 \notin v(\alpha)$ using the appropriate clauses of the definition of v.

This ends the proof of (II) above. It remains to prove (I), i.e., to show that $v \models \Sigma$ for each $\Sigma \in S$. So let $\Sigma \in S$. Then $\Sigma = \varphi_1, \ldots, \varphi_k \Rightarrow \psi_1, \ldots, \psi_l$ for some integers k, l and formulas $\varphi_i, \psi_j, i = 1, \ldots, k, j = 1, \ldots, l$. Clearly, we cannot have both $\{\varphi_1, \ldots, \varphi_k\} \subseteq \Gamma^*$ and $\{\psi_1, \ldots, \psi_l\} \subseteq \Delta^*$, for then $\Gamma^* \Rightarrow \Delta^*$ would be derivable from Σ, and hence from S, by weakening. Since $F(S) \subseteq \Gamma^* \cup \Delta^*$, this implies that either $\varphi_i \in \Delta^*$ for some i, or $\psi_j \in \Gamma^*$ for some j. Hence by (A) and (B), which we have already proved, we have either $v \not\models \varphi_i$ for some i, or $v \models \psi_j$ for some j, which implies that $v \models \Sigma$. ■

Corollary 5.3. *The calculus $\mathcal{C}_I^4$ is (weakly) sound and complete for $\vdash_{\mathcal{M}_I^4}$, and the cut rule is admissible in it. In particular: If Γ is a finite set of formulas, and φ is a formula, then $\Gamma \vdash_{\mathcal{M}_I^4} \varphi$ if the sequent $\Gamma \Rightarrow \varphi$ has a cut-free proof in $\mathcal{C}_I^4$.* [5]

Proof. This follows from the last theorem and Note 3.4. ■

[5] This corollary was first proved (using a different method) in [5].

Note 5.14. The finiteness assumption can in fact be dropped from the formulation of Theorem 5.13, for the theorem holds for infinite premise sets too. However, we skip the proof of this fact here, for such a generalization seems to be of little practical usefulness for the purposes of this paper.

5.2. The other logics

Strongly finitely sound and complete calculi for the other main logics investigated in this paper can be obtained by extending $\mathcal{C}_I^4$ and $\underline{\mathcal{C}}_I^4$ with appropriate rules and axiom:

General source-processor logic with complete information
Calculi corresponding to this case are obtained by augmenting $\mathcal{C}_I^4$ and $\underline{\mathcal{C}}_I^4$ with either the excluded middle axiom $\Rightarrow \varphi, \neg\varphi$, or by the left-to-right swap rule

$$\frac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}$$

(However, the addition of the swap rule does not allow us to eliminate any of the previous negation rules, for none of them is derivable from it).

Dunn-Belnap's logic
To obtain the calculi for Belnap's logic, we augment $\mathcal{C}_I^4$ and $\underline{\mathcal{C}}_I^4$ by the two symmetric rules "missing" from them, i.e.

$$(\vee \Rightarrow)\ \frac{\Gamma, \varphi \Rightarrow \Delta \qquad \Gamma, \psi, \Rightarrow \Delta}{\Gamma, \varphi \vee \psi \Rightarrow \Delta} \qquad (\neg\wedge \Rightarrow)\ \frac{\Gamma, \neg\varphi \Rightarrow \Delta \qquad \Gamma, \neg\psi \Rightarrow \Delta}{\Gamma, \neg(\varphi \wedge \psi) \Rightarrow \Delta}$$

D'Ottaviano and da Costa's logic
The calculi for the above paraconsistent logic are obtained by adding either the excluded middle axiom $\Rightarrow \varphi, \neg\varphi$, or the left-to-right swap rule, to the calculi for Belnap's logic.

Kleene's logic As is well known, calculi for Kleene's 3-valued logic are obtained by adding to the calculi for Belnap's logic either the axiom $\varphi, \neg\varphi \Rightarrow$ (corresponding to the law of contradiction), or the right-to-left swap rule $\dfrac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma, \neg\varphi \Rightarrow \Delta}$.

Theorem 5.15. The obvious analogues of Theorem 5.13 (and its corrolary 5.3) hold for each of the calculi introduced above with respect to their associate matrix/Nmatrix.

The proofs are similar to that of Theorem 5.13, and are left to the reader.

6. Future research

One direction of future research is to explore the general case of the universal strategy, namely, one where the sources can also provide information about complex formulas. As the introduction to Section 4 implies, it will split in two subcases: one when the final processor valuation d can also take the value $\top$, and one where this is not possible due to an additional constraint, like classical coherence (see Section 2.2.1).

Another directions is to investigate other variants of the framework, especially those signalled in Subsections 2.2.1 and 2.2.3. [6]

Finally, a major goal will be to upgrade our framework and results to first-order languages.

Acknowledgment

This research was supported by THE ISRAEL SCIENCE FOUNDATION founded by The Israel Academy of Sciences and Humanities.

References

[1] A. Avron, *Natural 3-valued logics : characterization and proof theory*, The Journal of Symbolic Logic, Vol.56, No. 1, pp. 276-294, 1991.

[2] A. Avron, *Logical Non-determinism as a Tool for Logical Modularity: An Introduction*, in **We Will Show Them: Essays in Honor of Dov Gabbay**, Vol 1 (S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, and J. Woods, eds.), 105-124, College Publications, 2005.

[3] A. Avron and I. Lev, *Canonical Propositional Gentzen-Type Systems*, in **Proceedings of the 1st International Joint Conference on Automated Reasoning (IJCAR 2001)** (R. Goré, A Leitsch, T. Nipkow, Eds), LNAI 2083, 529-544, Springer Verlag, 2001.

[4] A. Avron, and I. Lev, *Non-deterministic Multiple-valued Structures*, Journal of Logic and Computation, Vol. 15 (2005), 241-261.

[5] A. Avron, J. Ben-Naim, and B. Konikowska, *Cut-free Ordinary Sequent Calculi for Logics Having Finite-Valued Semantics*, Logica Universalis Vol. 1 (2006), 41-69.

[6] Some partial results in this direction have already been obtained concerning the important case in which all the sources are classically closed. The resulting logic is quite interesting: it is paraconsistent, but every classical tautology is valid in it. It respects many important classical equivalences, but not the distributive law. Especially important is the fact that there is no N such that a processor valuation is "good" if it is obtained in a framework of this type with at most N sources (while just two suffice for the cases investigated in this paper).

[6] N. D. Belnap, *How computers should think*, In G. Ryle, editor, **Contemporary Aspects of Philosophy**, 30-56. Oriel Press, Stocksfield, England, 1977.

[7] N. D. Belnap, *A useful four-valued logic*, In G. Epstein and J. M. Dunn, editors, **Modern Uses of Multiple-Valued Logic**, 7-37. Reidel, Dordrecht, 1977.

[8] W. A. Carnielli and M. Lima.-Marques, *Society semantics for multiple-valued logics*, in Proceedings of the XII EBL- Advances in Contemporary Logic and Computer Science (Walter A. Carnielli and Itala M. L. D'Ottaviano, eds.), American Mathematical Society, Series Contemporary Mathematics, Volume 235, 33-52, 1999.

[9] W. A. Carnielli and J. Marcos and S. de Amo, *Formal inconsistency and evolutionary databases*, Logic and Logical Philosophy, Vol. 8, 115-152, 2000.

[10] M. Fitting, *Kleene's three-valued logics and their children.* Fundamenta Informaticae, Vol.20, 113-131, 1994.

[11] M. L. Ginsberg, *Multivalued logics: A uniform approach to reasoning in AI.* Computer Intelligence, Vol.4, 256-316, 1988.

[12] I.M.L. D'Ottaviano and N.C.A. da Costa, *Sur un problème de Jaśkowski.* In Comptes Rendus de l'Académie des Sciences de Paris, Vol. 270, 1349-1353, 1970.

[13] I. L. M. D'Ottaviano, *The completeness and compactness of a three-valued first-order logic*, Revista Colombiana de Matematicas, vol. XIX (1985), 31–42.

[14] R. L. Epstein, **The semantic foundation of logic**, 2nd ed., vol. I: Propositional Logics, ch. IX, Kluwer Academic Publisher, 1995.

[15] J. M. Dunn, *Intuitive Semantics for First Degree Entailments and "coupled trees"*, Philosophical Studies 29 (1976), 149-168.

Arnon Avron
School of Computer Science
Tel-Aviv University
Ramat Aviv 69978, Israel
aa@post.tau.ac.il

Jonathan Ben-Naim
Computer Science and Communications Research Unit
University of Luxembourg
L-1359 Luxembourg
jonathan.ben-naim@uni.lu

Beata Konikowska
Institute of Computer Science
Polish Academy of Sciences
Warsaw, Poland
beatak@ipipan.waw.pl