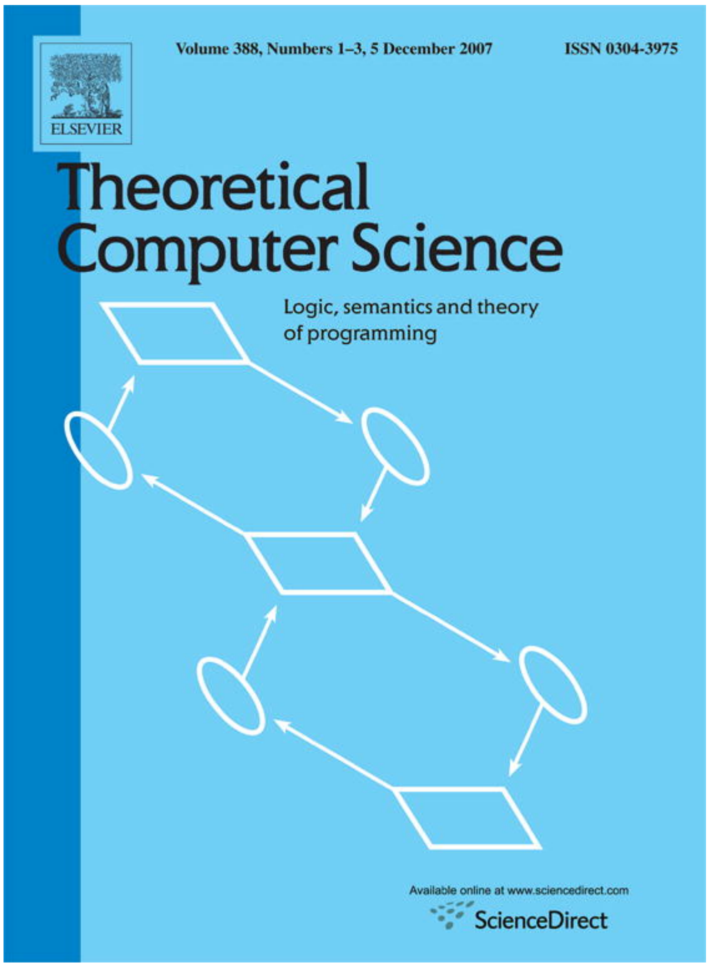
Volume 388, Numbers 1–3, 5 December 2007
ISSN 0304-3975
ELSEVIER
Theoretical Computer Science
Logic, semantics and theory of programming
Available online at www.sciencedirect.com
ScienceDirect

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Theoretical Computer Science 388 (2007) 319–336
Theoretical Computer Science
www.elsevier.com/locate/tcs

Belnap's logic and conditional composition

Alban Ponse, Mark B. van der Zwaag*

Department of Computer Science, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

Received 19 July 2005; received in revised form 10 August 2006; accepted 21 September 2007

Communicated by F. Pfenning

Abstract

We study two alternative bases for Belnap's four-valued logic and provide complete equational axiomatizations for them. One is called *conditional composition logic*. It has a single, ternary if-then-else connective with a sequential, operational reading, and four constants for the truth values. The other logic is called *guard logic*. The main motivation for this logic lies in its technical properties. It admits a useful type of canonical form (term representation), and a relatively simple strategy for equational reasoning.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Belnap's logic; Equational axiomatizations; Conditional composition; Guard logic; Completeness

1. Introduction

In 1977, N.D. Belnap introduced his "useful four-valued logic" in [2], see also the exposition in [1] (1992, Chapter XII). At present this logic is called *Belnap's logic*, and is well-known, mainly in the context of Entailment and Relevance Logic (see, e.g., [1,18]).

Belnap designed his logic to be used by question-answering computers that perform deductions based on information from multiple sources. The logic allows the computer to perform meaningful deductions when confronted with an inconsistency, as may arise when these sources provide mutually conflicting information. In particular, the *ex falso* principle (from an inconsistency, anything follows) is not valid in this logic.

For atomic questions, such a question-answering computer will reply T (*true*) to question p, if p has been asserted by one or more of its sources, and denied by none, and it will reply F (*false*) to question p, if one or more of its sources have denied p, and none has asserted it. If at least one source has asserted p, and at least one (other) source has denied p, the computer will reply B (*both*) to p, and if none of the sources has asserted or denied p, it will answer N (*none*).

This provides an intuition for the four truth values B, T, F, and N. An important notion is the *information ordering* $\leq$ corresponding to the uphill ordering in the *approximation lattice*:

$$\begin{tikzcd}
 & \mathrm{B} \arrow[dl, dash] \arrow[dr, dash] & \\
\mathrm{T} \arrow[dr, dash] & & \mathrm{F} \arrow[dl, dash] \\
 & \mathrm{N} &
\end{tikzcd}$$

* Corresponding author. Tel.: +31 20 525 7584.
E-mail address: mbz@science.uva.nl (M.B. van der Zwaag).

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2007.09.027

Here $a \leq b$ is read as a "approximates the information in" b. The value N does not give information at all, and B gives too much (inconsistent) information.

Belnap further defines the connectives negation, conjunction and disjunction. These definitions follow in a technical sense, and this is argued most clearly in [1], from the following requirements: (1) the connectives should be defined classically on $\{\text{T}, \text{F}\}$, (2) the connectives should be monotonic with respect to the information ordering, and (3) conjunction and disjunction should be duals in that they satisfy

$$a \wedge b = a \text{ iff } a \vee b = b, \quad \text{and} \quad a \wedge b = b \text{ iff } a \vee b = a.$$

The resulting truth tables:

¬	
B	B
T	F
F	T
N	N

∧	B	T	F	N
B	B	B	F	F
T	B	T	F	N
F	F	F	F	F
N	F	N	F	N

∨	B	T	F	N
B	B	T	B	T
T	T	T	T	T
F	B	T	F	N
N	T	T	N	N

In [2,1], these definitions are also given an intuitive motivation in the setting of a question-answering computer as sketched above. A perhaps not so obvious case is the conjunction of B and N. For this case we can envisage that B stands for concurrent T and F answers, and that T answers in conjunction with N vanish, while F answers remain, so that

$$\text{B} \wedge \text{N} = \text{F} \wedge \text{N} = \text{F}.$$

We shall denote Belnap's logic by $\mathbf{B}_4$.

Observe that Belnap's conjunction and disjunction are commutative, which is in line with the classical assumption that in the evaluation of a compound sentence the evaluation of the parts is directly available. In this article we shall consider *sequential* connectives over Belnap's truth values, i.e., connectives in which the evaluation of the subterms is ordered. In computer science there is ample reason for such a more operational perspective, as is witnessed by the sequential evaluation of logical connectives implemented in programming languages as diverse as, e.g., Java, Lisp and Prolog.

A well-known example of a sequential connective is the (non-commutative) conjunction of McCarthy's three-valued logic [16], which has the property that $\text{F} \wedge x = \text{F}$, and $* \wedge x = *$, for all x, where $*$ is the third truth value (next to T and F) in McCarthy's logic. Here it can be said that evaluation is *lazy* and proceeds sequentially from left to right: one only evaluates as far as is needed to evaluate the sentence. In this article we consider another sequential, programming-oriented connective called *conditional composition*. This is a ternary connective which stems from the programming construct *if-then-else*. Usually the *if* part is a (Boolean) condition, while the other two arguments are program fragments. In our variant all three arguments are logical formulas. We use the notation

$$x \lhd y \rhd z$$

(read as *if* y *then* x *else* z), taken from the 1987 article *Laws of programming* by Hayes *et al* [13], and find

$$x \lhd \text{T} \rhd y = x, \quad \text{and} \quad x \lhd \text{F} \rhd y = y.$$

So, in $x \lhd y \rhd z$, first the condition y is evaluated, and depending on the result x or z. We propose a sequential logic over Belnap's truth values, with conditional composition defined further by

$$x \lhd \text{B} \rhd y = x \oplus y, \quad \text{and} \quad x \lhd \text{N} \rhd y = \text{N},$$

where $\oplus$ is defined as the least upper bound operator in the information ordering given above. We can now give the following intuitive reading to $x \lhd y \rhd z$. If there is evidence for both the truth and the falsity of the condition y, continue with the sum of the information that can be obtained from the *alternatives* x and z, and if there is no evidence for y, stop (we are in a situation with no further alternatives). Observe that this reading is rather close to that of the programming construct *if-then-else*. This correspondence of the logical conditional composition with the operational *if-then-else* is made precise in our article [17], see Section 7.2 for a brief discussion.

Belnap demonstrates how useful reasoning with inconsistent information is tied to the monotonicity of operations with respect to the information ordering. We feel that conditional composition, being monotonic, is a natural connective for this kind of reasoning, perhaps even more so than conjunction and disjunction; an intuitive motivation for the classical connectives, although convincingly established in [2,1], is not quite straightforward.

It turns out that the conditional composition operation is definable in $\mathbf{B}_4$, be it in a rather complex way. Conversely, and perhaps even more interestingly, any $\mathbf{B}_4$ term can be expressed using only conditional composition and the four constants B, T, F and N. The crucial identities are rather simple:

$$\begin{aligned}\neg x &= \mathrm{F} \lhd x \rhd \mathrm{T},\\ x \wedge y &= (y \lhd x \rhd \mathrm{F}) \lhd \mathrm{B} \rhd (x \lhd y \rhd \mathrm{F}),\\ x \vee y &= (\mathrm{T} \lhd x \rhd y) \lhd \mathrm{B} \rhd (\mathrm{T} \lhd y \rhd x).\end{aligned}$$

Thus we have a sequential variant of Belnap's logic. We call this logic *conditional composition logic* and denote it by $\mathbf{C}_4$. The establishment of $\mathbf{C}_4$ and its axiomatization is the main contribution of this paper.

To bridge the gap between $\mathbf{B}_4$ and $\mathbf{C}_4$, we define a *guard logic* $\mathbf{G}_4$ over Belnap's truth values. The primitives of this logic are the constant F, negation, $\oplus$ (as mentioned above), and a binary *guard* connective that is strongly related to Dijkstra's *guarded command* [10]; both resemble an *if-then* composition. The guard connective is defined in $\mathbf{C}_4$ by

$$x : y = y \lhd x \rhd \mathrm{N}.$$

The main motivation for $\mathbf{G}_4$ is a technical one: terms can be rewritten to a convenient type of *canonical form* allowing proof strategies that culminate in a highly non-trivial completeness proof of our $\mathbf{G}_4$ axiomatization. The completeness of our $\mathbf{C}_4$ axioms then follows quite easily.

The paper is structured as follows: in the next section we present the three logics mentioned in detail and provide their complete axiomatizations. In Section 3 we provide translations between our three logics, and show that they are equally expressive. In Section 4 we define canonical forms for $\mathbf{G}_4$ and discuss some of their elementary properties. In Section 5 we prove that our $\mathbf{G}_4$ axiomatization is complete, and in Section 6 we derive the completeness of our $\mathbf{C}_4$ axiomatization. In Section 7, we discuss first a relation between $\mathbf{B}_4$ and Kleene's three-valued logic [15], and then some related work on the combination of process algebra and many-valued logics. We end with conclusions in Section 8.

2. Three logics over Belnap's truth values

We present three logics over Belnap's truth values. For each we provide a complete equational axiomatization. We start with Belnap's logic $\mathbf{B}_4$. Then we define $\mathbf{C}_4$ which has conditional composition as its only connective. Finally, we define a so-called guard logic, notation $\mathbf{G}_4$, which is based on a binary *guard* connective.

In this section we prove only for $\mathbf{B}_4$ that the provided axiomatization is complete. The completeness proofs for $\mathbf{G}_4$ and $\mathbf{C}_4$ are based on the completeness of $\mathbf{B}_4$, and are given in Sections 5 and 6.

2.1. Belnap's four-valued logic $\mathbf{B}_4$

In Belnap's Logic $\mathbf{B}_4$, the connectives negation, conjunction and disjunction are defined by the truth tables presented above in Section 1. Observe that conjunction ($\wedge$) and disjunction ($\vee$) can also be characterized as the greatest lower bound and the least upper bound operators of the following distributive lattice called the *truth* [11] or *logical* [2] lattice:

$$\begin{tikzcd} & \mathrm{T} \arrow[dl, dash] \arrow[dr, dash] & \\ \mathrm{B} \arrow[dr, dash] & & \mathrm{N} \arrow[dl, dash] \\ & \mathrm{F} & \end{tikzcd} \qquad (1)$$

Further observe that negation is an *involution*, i.e., $\neg\neg x = x$ is valid, and that the set $\{\wedge, \neg, \mathrm{B}, \mathrm{N}\}$ is a functional basis for $\mathbf{B}_4$: as usual, disjunction is defined in terms of negation and conjunction by

$$x \vee y = \neg(\neg x \wedge \neg y), \qquad (2)$$

and we define $\mathrm{F} = \mathrm{B} \wedge \mathrm{N}$ and $\mathrm{T} = \mathrm{B} \vee \mathrm{N}$.

Table 1
B$_4$ axioms

B1	$x \wedge y = y \wedge x$	B5	$(\mathrm{B} \vee \mathrm{N}) \wedge x = x$
B2	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$	B6	$\neg\neg x = x$
B3	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$	B7	$\neg \mathrm{B} = \mathrm{B}$
B4	$x \vee (x \wedge y) = x$	B8	$\neg \mathrm{N} = \mathrm{N}$

The characterization of **B**$_4$ as a distributive lattice with involution leads directly to the finite and complete equational axiomatization presented in Table 1. Axioms B1–B4 characterize the truth ordering as a distributive lattice, and axiom B6 characterizes negation as an involution. The axioms are easily shown to be sound using truth tables. The dual identities for the axioms (with $\wedge$ and $\vee$ interchanged), and the idempotence of disjunction and conjunction, are derivable using the definition of disjunction (2). These (standard) results are collected in the following lemma.

Lemma 1. *The following identities are derivable in* **B**$_4$*:*

$$x \vee y = y \vee x, \tag{3}$$
$$x \vee (y \vee z) = (x \vee y) \vee z, \tag{4}$$
$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z), \tag{5}$$
$$x \wedge (x \vee y) = x, \tag{6}$$
$$(\mathrm{B} \wedge \mathrm{N}) \vee x = x, \tag{7}$$
$$x \wedge x = x, \tag{8}$$
$$x \vee x = x. \tag{9}$$

Proof. For (6) we find that the left-hand side equals $\neg(\neg x \vee (\neg x \wedge \neg y))$ using (2) and B6. By B4 this term equals $\neg\neg x$. Now use B6. In this vein each of (3)–(7) is easily derived using the dual axiom. Idempotence of conjunction is derived by $x = x \wedge (x \vee (x \wedge x)) = x \wedge x$ using (6) and B4. Idempotence of disjunction is derived similarly. □

The proof of the completeness theorem below is due to Luttik and Rodenburg (personal communication); it is based on [14]. We use this lemma:

Lemma 2. *Every closed term of* **B**$_4$ *is derivably equal to one, and only one, of the four values* B, T, F, *and* N.

Proof. By induction on the size of terms.

Base case. The terms B, T, F, and N are mutually not derivably equal. Suppose that two of them are. Then they should have the same interpretation in any model of the axioms. In the four-element model constituted by the logical lattice (1) with negation, they have a distinct interpretation, a contradiction.

Inductive case. We first look at the negation $\neg t$ for some term t derivably equal to $a \in \{\mathrm{B}, \mathrm{T}, \mathrm{F}, \mathrm{N}\}$. If $a = \mathrm{B}$, use axiom B7, if $a = \mathrm{N}$, use axiom B8. If $a = \mathrm{T}$, recall that $\mathrm{T} = \mathrm{B} \vee \mathrm{N}$ and $\mathrm{F} = \mathrm{B} \wedge \mathrm{N}$ by definition and derive

$$\neg\mathrm{T} = \neg(\mathrm{B} \vee \mathrm{N}) = \neg\mathrm{B} \wedge \neg\mathrm{N} = \mathrm{B} \wedge \mathrm{N} = \mathrm{F},$$

using (2) and axioms B6, B7, and B8. Similarly, for $a = \mathrm{F}$, derive

$$\neg\mathrm{F} = \neg(\mathrm{B} \wedge \mathrm{N}) = \neg(\neg\mathrm{B} \wedge \neg\mathrm{N}) = \mathrm{B} \vee \mathrm{N} = \mathrm{T}.$$

Now consider the conjunction $t \wedge u$ for some terms t and u that are derivably equal to truth values a and b respectively. Note that we have $\mathrm{T} \wedge x = x$ (axiom B5), and

$$\mathrm{F} \wedge x = \neg(\mathrm{T} \vee \neg x) = \neg(\mathrm{T} \vee (\mathrm{T} \wedge \neg x)) = \neg\mathrm{T} = \mathrm{F}$$

using (2), B6, B5, B1, B4. Further using idempotence (8) and commutativity (B1) of conjunction, we find the only remaining case $\mathrm{B} \wedge \mathrm{N}$ which equals F by definition. □

Theorem 3. *The* **B**$_4$ *axioms are complete.*

Table 2
C$_4$ axioms

C1	$x_1 \lhd (y_1 \lhd z \rhd y_2) \rhd x_2 = (x_1 \lhd y_1 \rhd x_2) \lhd z \rhd (x_1 \lhd y_2 \rhd x_2)$		
C2	$(x_1 \lhd u \rhd x_2) \lhd z \rhd (y_1 \lhd u \rhd y_2) = (x_1 \lhd z \rhd y_1) \lhd u \rhd (x_2 \lhd z \rhd y_2)$		
C3	$(x_1 \lhd y \rhd x_2) \lhd y \rhd x_3 = x_1 \lhd y \rhd (x_2 \lhd y \rhd x_3)$		
C4	$\mathrm{T} \lhd x \rhd \mathrm{F} = x$		
C5	$x \lhd \mathrm{T} \rhd y = x$	C8	$x \lhd \mathrm{B} \rhd y = y \lhd \mathrm{B} \rhd x$
C6	$x \lhd \mathrm{F} \rhd y = y$	C9	$x \lhd \mathrm{B} \rhd \mathrm{N} = x$
C7	$x \lhd \mathrm{N} \rhd y = \mathrm{N}$	C10	$\mathrm{B} \lhd \mathrm{B} \rhd x = \mathrm{B}$

Proof. Let the **B**$_4$ axioms in Table 1 denote the variety of algebras with conjunction, disjunction, negation, and the four values B, T, F, and N.

First, it easily follows from Lemma 2 that the initial **B**$_4$ algebra is the four-element distributive lattice (1) with involution and with the two distinct fixed points of negation B and N.

We apply the following theorem from [14]: Any distributive lattice with involution is isomorphic with a subdirect product of isomorphic images of the four-element distributive lattice with involution and with two distinct fixed points of negation. From this theorem it follows that the **B**$_4$ axioms completely axiomatize the initial **B**$_4$ algebra. Suppose that $t = u$ is true in the initial algebra. Then this identity holds in any subdirect power of it, and since any **B**$_4$ algebra is isomorphic to such a subdirect power, we may conclude that $\mathbf{B}_4 \models t = u$. Hence $\mathbf{B}_4 \vdash t = u$ follows by Birkhoff's completeness theorem for equational logic [9]. □

2.2. *Conditional composition logic* **C**$_4$

The alternative logic **C**$_4$ over Belnap's truth values has one, ternary operator $\_\lhd\_\rhd\_$ called *conditional composition* as its only connective. A composition $x \lhd y \rhd z$ is read as *if y then x else z*; we define

$$x \lhd \mathrm{T} \rhd y = x, \quad x \lhd \mathrm{F} \rhd y = y, \quad x \lhd \mathrm{N} \rhd y = \mathrm{N},$$

and $x \lhd \mathrm{B} \rhd y = x \oplus y$, where $\oplus$ is the least upper bound operator of the information ordering lattice

$$\begin{tikzcd}[column sep=small] & \mathrm{B} \arrow[dl, no head] \arrow[dr, no head] & \\ \mathrm{T} \arrow[dr, no head] & & \mathrm{F} \arrow[dl, no head] \\ & \mathrm{N} & \end{tikzcd} \tag{10}$$

that we presented already above in Section 1. (The truth table for conditional composition is implicit in this definition.) We use $\oplus$ as an auxiliary operation, keeping in mind that it is defined as $\_\lhd \mathrm{B} \rhd\_$. The set $\{\lhd\rhd, \mathrm{B}, \mathrm{T}, \mathrm{F}, \mathrm{N}\}$ is a functional basis for **C**$_4$.

In Table 2 we give a complete set of axioms for **C**$_4$. Soundness of these axioms is easily verified using truth tables. Axiom C1 expresses that a composition that occurs nested in a condition can be pushed outwards. Axiom C2 says that if the subsequent condition is equal for both alternatives of a conditional composition, then the order in which the conditions are evaluated can be changed. Axiom C3 says that $\_ \lhd x \rhd \_$ is associative for any x. Axiom C4 may be interpreted as expressing idempotence of conditional composition (the identity $x \lhd x \rhd x = x$ is derivable, see Section 4.2). Axioms C5 and C6 characterize the if-then-else reading of conditional composition for the classical truth values. A composition is undefined if its condition is undefined (axiom C7). Finally, the axioms C8–C10 complete the characterization of $\_ \lhd \mathrm{B} \rhd \_$ as the least upper bound operator for the information ordering. In Section 6 we give a detailed proof of the completeness of this **C**$_4$ axiomatization.

We conclude the introduction of **C**$_4$ with some typical identities concerning B, N, and the distribution of conditional composition over the auxiliary connective $\oplus$ (that is, $\_ \lhd \mathrm{B} \rhd \_$).

Table 3
$\mathbf{G}_4$ axioms

G1	$x \oplus y = y \oplus x$	G9	$x = x : \neg\mathrm{F} \oplus \neg x : \mathrm{F}$
G2	$(x \oplus y) \oplus z = x \oplus (y \oplus z)$	G10	$\neg\mathrm{F} : x = x$
G3	$x \oplus x = x$	G11	$\mathrm{F} : x = \mathrm{F} : y$
G4	$(\mathrm{F} \oplus \neg\mathrm{F}) \oplus x = \mathrm{F} \oplus \neg\mathrm{F}$	G12	$x : (y \oplus z) = x : y \oplus x : z$
G5	$\mathrm{F} : x \oplus y = y$	G13	$(x \oplus y) : z = x : z \oplus y : z$
G6	$x : (y : z) = y : (x : z)$	G14	$\neg(x \oplus y) = \neg x \oplus \neg y$
G7	$(x : y) : z = x : (y : z)$	G15	$\neg(x : y) = x : \neg y$
G8	$x : (x : y) = x : y$	G16	$\neg\neg x = x$

Lemma 4. *The following identities are derivable in* $\mathbf{C}_4$*:*

$$x \lhd \mathrm{B} \rhd x = x, \tag{11}$$

$$\mathrm{N} \lhd x \rhd \mathrm{N} = \mathrm{N}, \tag{12}$$

$$(x_1 \oplus x_2) \lhd y \rhd z = (x_1 \lhd y \rhd z) \oplus (x_2 \lhd y \rhd z), \tag{13}$$

$$x \lhd (y_1 \oplus y_2) \rhd z = (x \lhd y_1 \rhd z) \oplus (x \lhd y_2 \rhd z), \tag{14}$$

$$x \lhd y \rhd (z_1 \oplus z_2) = (x \lhd y \rhd z_1) \oplus (x \lhd y \rhd z_2). \tag{15}$$

Proof. For (11) we derive using axiom C9 that $x \lhd \mathrm{B} \rhd x$ equals

$$(x \lhd \mathrm{B} \rhd \mathrm{N}) \lhd \mathrm{B} \rhd (x \lhd \mathrm{B} \rhd \mathrm{N})$$

which is derivably equal to x by axioms C1, C10, and C9. The left-hand side of (12) equals $(y \lhd \mathrm{N} \rhd y) \lhd x \rhd (y \lhd \mathrm{N} \rhd y)$ by axiom C7. Now apply C2 and C7. Eqs. (13) and (15) are derived using (11) and axiom C2. Eq. (14) is an instance of axiom C1. □

2.3. Guard logic $\mathbf{G}_4$

The last logic we consider is based on the binary *guard* connective $\_ : \_$ defined by

$$x : y = y \lhd x \rhd \mathrm{N},$$

where x is called the guard of y. We note that both the guard connective and the $\oplus$ are introduced in [11], where the latter is called the *gullibility operator*. Furthermore, the guard connective corresponds to Dijkstra's *guarded command* construct [10]. Together with $\oplus$ and negation, we obtain the expressiveness of $\mathbf{C}_4$ and $\mathbf{B}_4$, as will be shown in Section 3. We refer to this logic, with primitives $\{\oplus, :, \neg, \mathrm{F}\}$, as *guard logic*, denoted by $\mathbf{G}_4$. The remaining truth values are defined by $\mathrm{T} = \neg\mathrm{F}$, $\mathrm{B} = \mathrm{F} \oplus \neg\mathrm{F}$, and $\mathrm{N} = \mathrm{F} : \mathrm{F}$.

Let $\neg$ bind strongest and $\oplus$ bind weakest. The axioms of $\mathbf{G}_4$ are collected in Table 3. These axioms are sound: this is easily verified using truth tables (for $\oplus$ and $:$ the truth tables follow from their definitions). These axioms are also complete: see Section 5.

Some typical identities characterizing $\mathbf{G}_4$ are collected in the following lemma.

Lemma 5. *The following identities are derivable in* $\mathbf{G}_4$*:*

$$x : \mathrm{N} = \mathrm{N}, \tag{16}$$

$$\mathrm{N} : x = \mathrm{N}, \tag{17}$$

$$\mathrm{B} : x = x, \tag{18}$$

$$\neg\mathrm{B} = \mathrm{B} \tag{19}$$

$$\neg\mathrm{N} = \mathrm{N}. \tag{20}$$

Proof. Eq. (16) is easily derived using the definition $\mathrm{N} = \mathrm{F} : \mathrm{F}$ and axioms G6 and G11. Derivation of (17) is similar using G7 and G11. Eq. (18) is easily derived using the definition $\mathrm{B} = \mathrm{F} \oplus \neg\mathrm{F}$ and axioms G13, G10, and G5. Similarly, (19) is derived using G14, G16, and G1. Eq. (20) is derived using G15 and G11. □

Table 4
Translations; x' stands for the inductive application of the translation on x

$\mathbf{B}_4$ to $\mathbf{C}_4$			
$\neg t \mapsto \mathrm{F} \lhd t' \rhd \mathrm{T}$	$t \wedge u \mapsto (u' \lhd t' \rhd \mathrm{F}) \lhd \mathrm{B} \rhd (t' \lhd u' \rhd \mathrm{F})$		
$t \mapsto t$ for $t = \mathrm{B}, \mathrm{N}$	$t \vee u \mapsto (\mathrm{T} \lhd t' \rhd u') \lhd \mathrm{B} \rhd (\mathrm{T} \lhd u' \rhd t')$		
$\mathbf{C}_4$ to $\mathbf{B}_4$			
$t \lhd u \rhd v \mapsto ((t' \wedge u') \vee (v' \wedge \neg u')) \vee (((t' \wedge v') \wedge \mathrm{N}) \vee ((u' \wedge \neg u') \wedge \mathrm{N}))$			
$\mathrm{B} \mapsto \mathrm{B}$	$\mathrm{F} \mapsto \mathrm{B} \wedge \mathrm{N}$	$\mathrm{N} \mapsto \mathrm{N}$	$\mathrm{T} \mapsto \mathrm{B} \vee \mathrm{N}$
$\mathbf{C}_4$ to $\mathbf{G}_4$			
$t \lhd u \rhd v \mapsto u' : t' \oplus \neg u' : v'$			
$\mathrm{B} \mapsto \mathrm{F} \oplus \neg \mathrm{F}$	$\mathrm{T} \mapsto \neg \mathrm{F}$	$\mathrm{F} \mapsto \mathrm{F}$	$\mathrm{N} \mapsto \mathrm{F} : \mathrm{F}$
$\mathbf{G}_4$ to $\mathbf{C}_4$			
$t : u \mapsto u' \lhd t' \rhd \mathrm{N}$	$\mathrm{F} \mapsto \mathrm{F}$		
$t \oplus u \mapsto t' \lhd \mathrm{B} \rhd u'$	$\neg t \mapsto \mathrm{F} \lhd t' \rhd \mathrm{T}$		
$\mathbf{G}_4$ to $\mathbf{B}_4$			
$t : u \mapsto ((t' \wedge u') \vee (\neg t' \wedge \mathrm{N})) \vee (u' \wedge \mathrm{N})$	$\mathrm{F} \mapsto \mathrm{B} \wedge \mathrm{N}$		
$t \oplus u \mapsto ((t' \wedge \mathrm{B}) \vee (u' \wedge \mathrm{B})) \vee ((t' \wedge u') \wedge \mathrm{N})$	$\neg t \mapsto \neg t'$		
$\mathbf{B}_4$ to $\mathbf{G}_4$			
$\neg t \mapsto \neg t'$	$t \wedge u \mapsto (t' : u' \oplus \neg t' : \mathrm{F}) \oplus (u' : t' \oplus \neg u' : \mathrm{F})$		
$\mathrm{B} \mapsto \mathrm{F} \oplus \neg \mathrm{F}$	$t \vee u \mapsto (t' : \neg \mathrm{F} \oplus \neg t' : u') \oplus (u' : \neg \mathrm{F} \oplus \neg u' : t')$		
$\mathrm{N} \mapsto \mathrm{F} : \mathrm{F}$			

3. Expressiveness results

We have defined three logics over Belnap's truth values. In Section 3.1 we show that these logics have exactly the same expressiveness, that is, their operators can be defined in terms of the other logics. Hence the logics can be considered the same, but with a different functional basis. So, we can freely use those operators that seem most appropriate. As in $\mathbf{G}_4$, we let $\neg$ bind strongest, and $\oplus$ bind weakest, when we combine operators from these logics.

Then, in Section 3.2, we characterize the expressiveness of the logics: in our three logics we can express exactly those functions on the truth values that are monotonic with respect to the information ordering (10).

Finally, in Section 3.3, we show that with the addition of one non-monotonic operator, we can express every function on the truth values.

3.1. Translations

We present translations between the logics $\mathbf{B}_4$, $\mathbf{C}_4$, and $\mathbf{G}_4$; that is, we show how to express any term in one logic using the primitives of each of the other logics. The translations are given in Table 4. For simplicity we use disjunction, which we defined as an auxiliary connective, in the translations to and from $\mathbf{B}_4$.

Theorem 6. *The translations in Table 4 are sound.*

Proof. Straightforward using truth tables. $\square$

We conclude that the logics

- $\mathbf{B}_4$ with functional basis $\{\wedge, \neg, \mathrm{B}, \mathrm{N}\}$,
- $\mathbf{C}_4$ with functional basis $\{\lhd\rhd, \mathrm{B}, \mathrm{T}, \mathrm{F}, \mathrm{N}\}$, and
- $\mathbf{G}_4$ with functional basis $\{\oplus, :, \neg, \mathrm{F}\}$

are equally expressive.

3.2. Truth-functional completeness for monotonic functions

We show that, with respect to the information ordering defined by lattice (10), the logic $\mathbf{C}_4$ (and hence also $\mathbf{B}_4$ and $\mathbf{G}_4$) is truth functionally complete for monotonic functions. Write T for the set $\{\mathrm{B}, \mathrm{T}, \mathrm{F}, \mathrm{N}\}$ of truth values.

Recall that an n-ary function f over T is monotonic with respect to a partial ordering $\leq$ on T, if whenever $a_i \leq b_i$ for $1 \leq i \leq n$, then

$$f(a_1, \ldots, a_n) \leq f(b_1, \ldots, b_n).$$

Let $\leq$ be the information ordering. First note that conditional composition is monotonic. This follows from the fact that $x \leq y$ if and only if $x \oplus y = y$ and that conditional composition distributes over $\oplus$ (Lemma 4). Furthermore, an n-ary function f over T can be expressed in $\mathbf{C}_4$ if there is a term t with variables $x_1, \ldots, x_n$, and no others, such that

$$f(a_1, \ldots, a_n) = t[a_1/x_1, \ldots, a_n/x_n]$$

for all $a_1, \ldots, a_n \in T$.

Theorem 7. *Functions that are monotonic with respect to the information ordering can be expressed in* $\mathbf{C}_4$.

Proof. Let f be a $(k+1)$-ary monotonic function on T, and write $\bar{x}, y$ for $(k+1)$-tuples ($\bar{x}$ may be empty). We prove that the function f is expressible by induction on k; assume that $f(\bar{x}, a)$ is expressible, for all $a \in T$.

We define

$$f(\bar{x}, y) = f(\bar{x}, \mathrm{N}) \oplus f(\bar{x}, \mathrm{T}) \lhd y \rhd f(\bar{x}, \mathrm{F}) \oplus (\mathrm{N} \lhd y \rhd f(\bar{x}, \mathrm{B})) \lhd y \rhd \mathrm{N}.$$

To see that this indeed defines f, we make a case distinction on the value of y. For the respective cases $y = \mathrm{N}, \mathrm{T}, \mathrm{F}, \mathrm{B}$, we find that the right-hand side equals

(1) $f(\bar{x}, \mathrm{N}) \oplus \mathrm{N} \oplus \mathrm{N}$
(2) $f(\bar{x}, \mathrm{N}) \oplus f(\bar{x}, \mathrm{T}) \oplus \mathrm{N}$,
(3) $f(\bar{x}, \mathrm{N}) \oplus f(\bar{x}, \mathrm{F}) \oplus \mathrm{N}$, and
(4) $f(\bar{x}, \mathrm{N}) \oplus f(\bar{x}, \mathrm{T}) \oplus f(\bar{x}, \mathrm{F}) \oplus f(\bar{x}, \mathrm{B})$.

For each case it is easily found that this term equals $f(\bar{x}, y)$ using the monotonicity of f (for example, $\mathrm{N} < \mathrm{T}$, so $f(\bar{x}, \mathrm{N}) \oplus f(\bar{x}, \mathrm{T}) = f(\bar{x}, \mathrm{T})$). $\square$

3.3. Truth-functional completeness

Because all operators defined so far are monotonic, we cannot express non-monotonic functions on the truth values. We show that with the addition of one non-monotonic operator, we can express every function on the truth values. The unary definedness operator $\downarrow$ (see [3]) is defined by

$$\downarrow\mathrm{B} = \mathrm{F}, \ \downarrow\mathrm{T} = \mathrm{T}, \ \downarrow\mathrm{F} = \mathrm{T}, \ \downarrow\mathrm{N} = \mathrm{F}.$$

This operator is not monotonic; for example, we have $\mathrm{T} \leq \mathrm{B}$ while $\downarrow\mathrm{T} \not\leq \downarrow\mathrm{B}$.

Theorem 8. *With the addition of the definedness operator* $\downarrow$ *to* $\mathbf{B}_4$, $\mathbf{C}_4$, *or* $\mathbf{G}_4$ *we obtain a logic that is truth functionally complete.*

Proof. It is sufficient to prove this for $\mathbf{B}_4$. We introduce auxiliary functions $\kappa_a(\_)$ that satisfy

$$\kappa_a(b) = \begin{cases} \mathrm{T} & \text{if } a = b, \\ \mathrm{F} & \text{otherwise,} \end{cases}$$

for $a, b \in T$:

$$\kappa_{\mathrm{B}}(x) = \downarrow((x \wedge \neg x) \vee \mathrm{N}),$$
$$\kappa_{\mathrm{T}}(x) = \downarrow x \wedge x,$$
$$\kappa_{\mathrm{F}}(x) = \kappa_{\mathrm{T}}(\neg x),$$
$$\kappa_{\mathrm{N}}(x) = \downarrow((x \wedge \neg x) \vee \mathrm{B}).$$

Let f be a $(k+1)$-ary function on T. Write $\bar{x}, y$ for $(k+1)$-tuples. We define

$$f(\bar{x}, y) = \bigvee_{a \in T} (\kappa_a(y) \wedge f(\bar{x}, a)).$$

Hence, the theorem follows by induction on k. $\square$

We remark that another non-monotonic operator which adds full expressiveness is Fitting's unary *conflation* operator $-$, defined by $-\mathrm{B} = \mathrm{N}$, $-\mathrm{T} = \mathrm{T}$, $-\mathrm{F} = \mathrm{F}$, and $-\mathrm{N} = \mathrm{B}$, see [11]. Using conflation, the definedness operator is defined by

$$\downarrow x = (x \vee \neg x) \wedge -(x \vee \neg x).$$

4. Canonical forms for guard logic

We define a canonical form representation for $\mathbf{G}_4$ terms. We prove that we can derive so-called *optimal* canonical forms for terms, and suggest a general strategy for proving equality of terms.

4.1. Definition of canonical forms

Since the guard connective is associative (by G7), we shall not write parentheses in a term

$$u_1 : \cdots : u_n : t.$$

We call the terms $u_1, \ldots, u_n$ the *guards* of t. Since the guards are unordered (by G6), and multiple occurrences of the same guard can be identified (by G9), we shall, when this is convenient, use the set-like notation

$$\{u_1, \ldots, u_n\}t$$

for such a term, where it is understood that the guards are distinct and unordered. We use the letters α, β to stand for a finite set of guards; so αt stands for some $u_1 : \cdots : u_k : t$ with $k \geq 0$. (In particular the set may be empty; we let $\emptyset t = t$.)

Proposition 9. *For all terms $t, u_1, \ldots, u_n$ we have*

$$u_1 : \cdots : u_n : t = (u_1 \wedge \cdots \wedge u_n) : t = (u_1 \otimes \cdots \otimes u_n) : t,$$

where $\otimes$ is the greatest lower bound operation for the information ordering lattice.

We define *simple canonical forms* as follows: the truth values T and F are simple canonical forms; if t is a simple canonical form, then $u : t$ is a simple canonical form for any term u. We see that every simple canonical form is of the form

$$u_1 : \cdots : u_n : a$$

with $n \geq 0$ and $a \in \{\mathrm{T}, \mathrm{F}\}$, and using the convention introduced above this may be written as $\{u_1, \ldots, u_n\}a$.

A simple canonical form $\{u_1, \ldots, u_n\}a$ is *optimal* if all of its guards are *literals*, where a literal is defined as being either a variable or a negated variable.

A *canonical form* is either N, in which case we say it is *empty*, or a least upper bound

$$t_0 \oplus \cdots \oplus t_n$$

of simple canonical forms $t_0, \ldots, t_n$ with $n \geq 0$. A canonical form is *optimal* if either it is empty or each of its simple canonical forms is optimal.

Proposition 10. *A canonical form can be written as*

$$\bigoplus_i \alpha_i \mathrm{T} \oplus \bigoplus_j \beta_j \mathrm{F},$$

in which the finite sets α_i, β_j may be seen as the support *for* T *and* F*, respectively.*

4.2. Deriving canonical forms

Deriving a canonical form is trivial: any term t equals the canonical form $t : \mathrm{T} \oplus \neg t : \mathrm{F}$ by G9. Canonical forms are *optimized*, i.e., rewritten towards an optimal canonical form, by pushing $\oplus$ outwards, and negation inwards.

Lemma 11. *The term $t : u$ is derivably equal to an optimal canonical form, for all optimal simple canonical forms t and u.*

Proof. Take optimal simple canonical forms $t = \alpha a$ and $u = \beta b$, where $a, b \in \{\mathrm{T}, \mathrm{F}\}$, and α, β finite sets of literals.

Using G7 we derive that $t : u$ equals $\alpha(a : u)$. If $a = \mathrm{T}$, then, by G10, $t : u = \alpha u = (\alpha \cup \beta)b$, which is an optimal canonical form. Otherwise, if $a = \mathrm{F}$, then, by G7, G11, and (16), $t : u = \alpha\mathrm{N} = \mathrm{N}$, which is an optimal canonical form. □

Lemma 12. *The term $t : u$ is derivably equal to an optimal canonical form, for all optimal canonical forms t and u.*

Proof. Take optimal canonical forms $t = \bigoplus_i t_i$ and $u = \bigoplus_j u_j$, where the t_i and u_j are optimal simple canonical forms.

If t and/or u is empty, i.e., equal to N, then we find by (16) and (17) that $t : u$ equals N, which is an optimal canonical form. So we further assume that both are non-empty. It is easy to derive that t equals

$$\bigoplus_{i,j}(t_i : u_j)$$

using G12 and G13. Each $t_i : u_j$ is derivably equal to an optimal canonical form by Lemma 11. □

Lemma 13. *The term $\neg t$ is derivably equal to an optimal canonical form, for all optimal canonical forms t.*

Proof. Easy, using induction on the number of symbols in t, and G14 and G15. □

Theorem 14. *Every* $\mathbf{G}_4$ *term is derivably equal to an optimal canonical form.*

Proof. Let t be a term. We use induction on the number of symbols in t.

If $t \in \{\mathrm{T}, \mathrm{F}, \mathrm{N}\}$, then it is an optimal canonical form.

If $t = \mathrm{B}$, then $t = \mathrm{B} : \mathrm{T} \oplus \neg\mathrm{B} : \mathrm{F} = \mathrm{T} \oplus \mathrm{F}$ by G9, (18), and (19), and this right-hand side is an optimal canonical form.

If t is a variable, then it equals the optimal canonical form $t : \mathrm{T} \oplus \neg t : \mathrm{F}$ by axiom G9.

If $t = \neg t_1$ then it is derivably equal to an optimal canonical form by the induction hypothesis and application of Lemma 13.

If $t = t_1 : t_2$, then we use the induction hypothesis and Lemma 12.

Finally, if $t = t_1 \oplus t_2$ we only use the induction hypothesis. □

For example, we derive for variables x, y, z that

$$\begin{aligned} x \lhd y \rhd z &= y : x \oplus \neg y : z \\ &= y : (x : \mathrm{T} \oplus \neg x : \mathrm{F}) \oplus \neg y : (z : \mathrm{T} \oplus \neg z : \mathrm{F}) \\ &= y : x : \mathrm{T} \oplus y : \neg x : \mathrm{F} \oplus \neg y : z : \mathrm{T} \oplus \neg y : \neg z : \mathrm{F}, \end{aligned}$$

where we eliminated the conditional composition according to its translation (see Table 4), and we used G9 and G12. Using our set-like notation for guards we may write

$$x \lhd y \rhd z = \{x, y\}\mathrm{T} \oplus \{\neg x, y\}\mathrm{F} \oplus \{\neg y, z\}\mathrm{T} \oplus \{\neg y, \neg z\}\mathrm{F} \tag{21}$$

for this identity.

Lemma 15 (*Absorption*). *The following identity is derivable:*

$$x : y : z \oplus y : z = y : z. \tag{Abs}$$

Proof. Using (18), G13, G1, and G4, we derive

$$x : y : z \oplus y : z = x : y : z \oplus \text{B} : y : z = (x \oplus \text{B}) : y : z = \text{B} : y : z = y : z. \quad \square$$

In the next section, where we frequently use the set-like notation for canonical forms, we shall use this absorption property to let a summand αx absorb a summand $(\alpha \cup \beta)x$. As an example, we derive the identity $x \lhd x \rhd x = x$:

$$\begin{aligned} x \lhd x \rhd x &= \{x\}\text{T} \oplus \{x, \neg x\}\text{F} \oplus \{x, \neg x\}\text{T} \oplus \{\neg x\}\text{F} \\ &= \{x\}\text{T} \oplus \{\neg x\}\text{F} \\ &= x, \end{aligned}$$

using (21), absorption, and G9. Note that, to apply this, the canonical forms need not be optimal. Our general strategy for proving equations between open terms is to write both sides as canonical forms, optimize them as far as is needed, and then apply absorption.

Finally, the following identity is easy to derive using G9:

$$\alpha x = (\alpha \cup \{x\})\text{T} \oplus (\alpha \cup \{\neg x\})\text{F}. \tag{22}$$

5. Completeness of guard logic

We have presented the three equally expressive logics $\mathbf{B}_4$, $\mathbf{G}_4$, and $\mathbf{C}_4$, and we proved completeness for the $\mathbf{B}_4$ axiomatization (Theorem 3). Also, we have defined sound translations between these logics in Table 4. In this section we prove the completeness of $\mathbf{G}_4$ from the completeness of $\mathbf{B}_4$. In Section 6 we prove completeness of $\mathbf{C}_4$ in the same way, but then relative to the completeness of $\mathbf{G}_4$.

We argue as follows. If the translation of each $\mathbf{B}_4$ axiom is derivable in $\mathbf{G}_4$, then each $\mathbf{B}_4$ derivation can be mimicked in $\mathbf{G}_4$. To complete the proof we argue that the translations are invariant with respect to derivability. We explain this in some more detail: for t a term in the $\mathbf{G}_4$ signature, we write t' for its translation to $\mathbf{B}_4$ and for t a term in the $\mathbf{B}_4$ signature, we write t^* for its translation to $\mathbf{G}_4$. Now assume $\mathbf{G}_4 \models u = v$. Then, by translation and the completeness of $\mathbf{B}_4$, we have $\mathbf{B}_4 \vdash u' = v'$. Since we have proved that every derivation for this identity can be mimicked in $\mathbf{G}_4$, it follows that $\mathbf{G}_4 \vdash (u')^* = (v')^*$. Finally, invariance of our back-and-forth translation, i.e., $\mathbf{G}_4 \vdash t = (t')^*$, yields $\mathbf{G}_4 \vdash u = v$, as was to be shown.

We repeat here the translations between $\mathbf{B}_4$ and $\mathbf{G}_4$ as defined above in Table 4. From $\mathbf{B}_4$ to $\mathbf{G}_4$:

$$\begin{aligned} \text{B}^* &= \text{F} \oplus \neg\text{F}, \\ \text{N}^* &= \text{F} : \text{F}, \\ (\neg x)^* &= \neg x^*, \\ (x \wedge y)^* &= (x^* : y^* \oplus \neg x^* : \text{F}) \oplus (y^* : x^* \oplus \neg y^* : \text{F}), \\ (x \vee y)^* &= (x^* : \neg\text{F} \oplus \neg x^* : y^*) \oplus (y^* : \neg\text{F} \oplus \neg y^* : x^*). \end{aligned}$$

And vice versa:

$$\begin{aligned} \text{F}' &= \text{B} \wedge \text{N}, \\ (\neg x)' &= \neg x', \\ (x : y)' &= ((x' \wedge y') \vee (\neg x' \wedge \text{N})) \vee (y' \wedge \text{N}), \\ (x \oplus y)' &= ((x' \wedge \text{B}) \vee (y' \wedge \text{B})) \vee ((x' \wedge y') \wedge \text{N}). \end{aligned}$$

Variables translate to themselves.

5.1. *Derivation of the* $\mathbf{B}_4$ *axioms in* $\mathbf{G}_4$

We prove for every $\mathbf{B}_4$ axiom $t = u$ that $\mathbf{G}_4 \vdash t^* = u^*$. In the cases of the axioms B2–B4, it is not easy to find a “direct” derivation; in these cases we use rewriting to canonical forms, after which application of absorption (Abs) yields the required identity. We have omitted the details of this straightforward rewriting.

B1. This axiom translates to

$$(x : y \oplus \neg x : \text{F}) \oplus (y : x \oplus \neg y : \text{F}) = (y : x \oplus \neg y : \text{F}) \oplus (x : y \oplus \neg x : \text{F}),$$

which is an instance of axiom G1.

B2. It is easy to derive using straightforward rewriting towards canonical forms and absorption, that the translations of both sides equal the canonical form

$$\{x, y\}z \oplus \{x, z\}y \oplus \{z, y\}x \oplus \{\neg x\}\text{F} \oplus \{\neg y\}\text{F} \oplus \{\neg z\}\text{F}.$$

B3. Both sides rewrite to the optimal canonical form

$$\{\neg x\}\text{F} \oplus \{\neg y, \neg z\}\text{F} \oplus \{x, y\}\text{T} \oplus \{x, z\}\text{T},$$

using absorption and (22) during optimization.

B4. It is not difficult to derive both sides equal to the optimal canonical form $\{x\}\text{T} \oplus \{\neg x\}\text{F}$.

B5. First, the translation of $\text{B} \vee \text{N}$ equals $(\text{T} \oplus \text{N}) \oplus (\text{N} \oplus \text{N})$, where $\text{T} = \neg\text{F}$ and $\text{N} = \text{F} : \text{F}$ by definition, using the identities of Lemma 5. This term is derivably equal to T by axioms G1, G2, and G5. Next, we find that $(\text{B} \vee \text{N}) \wedge x$ translates to

$$(\text{T} : x \oplus \neg\text{T} : \text{F}) \oplus (x : \text{T} \oplus \neg x : \text{F})$$

which equals

$$x \oplus (x : \text{T} \oplus \neg x : \text{F})$$

by G10, G16, G1, G5. The proof is finished straightforwardly using axiom G9 and G3.

B6. Equals G16.

B7 and B8. Have been derived in Lemma 5.

5.2. *Translation invariance*

We give a proof of the translation invariance: we show that every term t of $\mathbf{G}_4$ is derivably equal to $(t')^*$.

We consider the cases $t = u \oplus v$, $t = u : v$, and $t = \neg u$, where u and v are arbitrary terms. We prove that $(t')^*$ is derivably equal to t in $\mathbf{G}_4$ using induction on terms: we assume that $(x')^*$ is derivably equal to x for $x = u, v$. The last case (negation) is trivial.

Let $t = u \oplus v$. First we translate to $\mathbf{B}_4$:

$$t' = ((s_1 \vee s_2) \vee s_3),$$

where

$$\begin{aligned} s_1 &= u' \wedge \text{B},\\ s_2 &= v' \wedge \text{B},\\ s_3 &= (u' \wedge v') \wedge \text{N}. \end{aligned}$$

Now, we translate t' back to $\mathbf{G}_4$. We apply this translation bottom-up: we first translate the s_i. Using B, T, and N as abbreviations in $\mathbf{G}_4$, we find

$$\begin{aligned} s_1^* &= ((u')^* : \text{B} \oplus \neg((u')^*) : \text{F}) \oplus (\text{B} : (u')^* \oplus \neg\text{B} : \text{F})\\ &= (u : \text{B} \oplus \neg u : \text{F}) \oplus (\text{B} : u \oplus \neg\text{B} : \text{F})\\ &= (u : \text{B} \oplus \neg u : \text{F}) \oplus (u \oplus \text{F})\\ &= (u : \text{T} \oplus u : \text{F} \oplus \neg u : \text{F}) \oplus (u : \text{T} \oplus \neg u : \text{F} \oplus \text{F})\\ &= \{u\}\text{T} \oplus \text{F}, \end{aligned}$$

using first the induction hypothesis and Lemma 5, then G9, and finally absorption. In the same way we find

$$s_2^* = \{v\}\text{T} \oplus \text{F},$$

and, using straightforward rewriting and absorption,

$$s_3^* = \{\neg u\}\text{F} \oplus \{\neg v\}\text{F}.$$

Now, we compute a canonical form for $(s_1 \vee s_2)^*$. We find that

$$(s_1 \vee s_2)^* = (s_1^* : \text{T} \oplus \neg s_1^* : s_2^*) \oplus (s_2^* : \text{T} \oplus \neg s_2^* : s_1^*).$$

We derive

$$\begin{aligned} s_1^* : \text{T} \oplus \neg s_1^* : s_2^* &= (\{u\}\text{T} \oplus \text{F}) : \text{T} \oplus \neg(\{u\}\text{T} \oplus \text{F}) : (\{v\}\text{T} \oplus \text{F}) \\ &= \{u\}\text{T} \oplus \{v\}\text{T} \oplus \text{F}. \end{aligned}$$

Similarly, we find that

$$s_2^* : \text{T} \oplus \neg s_2^* : s_1^* = \{u\}\text{T} \oplus \{v\}\text{T} \oplus \text{F},$$

so that

$$(s_1 \vee s_2)^* = \{u\}\text{T} \oplus \{v\}\text{T} \oplus \text{F}.$$

For $(t')^*$ we find

$$(t')^* = ((s_1 \vee s_2) \vee s_3)^* = r_1 \oplus r_2 \oplus r_3 \oplus r_4,$$

where

$$\begin{aligned} r_1 &= (s_1 \vee s_2)^* : \text{T}, \\ r_2 &= \neg(s_1 \vee s_2)^* : s_3^*, \\ r_3 &= s_3^* : \text{T}, \\ r_4 &= \neg s_3^* : (s_1 \vee s_2)^*. \end{aligned}$$

It is easy to derive

$$\begin{aligned} r_1 &= \{u\}\text{T} \oplus \{v\}\text{T}, \\ r_2 &= \{\neg u\}\text{F} \oplus \{\neg v\}\text{F}, \\ r_3 &= \text{N}, \\ r_4 &= \{\neg u, u\}\text{T} \oplus \{\neg u, v\}\text{T} \oplus \{\neg u\}\text{F} \oplus \{\neg v, u\}\text{T} \oplus \{\neg v, v\}\text{T} \oplus \{\neg v\}\text{F}. \end{aligned}$$

Using absorption we find

$$(t')^* = \{u\}\text{T} \oplus \{v\}\text{T} \oplus \{\neg u\}\text{F} \oplus \{\neg v\}\text{F}.$$

Now we finish this case using G9.

For the next case let $t = u : v$. First we translate to $\mathbf{B}_4$:

$$t' = ((s_1 \vee s_2) \vee s_3),$$

where

$$\begin{aligned} s_1 &= u' \wedge v', \\ s_2 &= \neg u' \wedge \text{N}, \\ s_3 &= v' \wedge \text{N}. \end{aligned}$$

Now, we translate t' back to $\mathbf{G}_4$. We apply this translation bottom-up: we first translate the s_i. We find

$$s_1^* = \{u\}v \oplus \{\neg u\}\text{F} \oplus \{v\}u \oplus \{\neg v\}\text{F},$$

using the induction hypothesis, i.e., $u = (u')^*$, $v = (v')^*$. Similarly we find $s_2^* = \{u\}\text{F}$, and $s_3^* = \{\neg v\}\text{F}$.

Now, we compute a canonical form for $(s_1 \vee s_2)^*$. We find that

$$(s_1 \vee s_2)^* = (s_1^* : \text{T} \oplus \neg s_1^* : s_2^*) \oplus (s_2^* : \text{T} \oplus \neg s_2^* : s_1^*).$$

We derive optimal canonical forms for each of these summands:

$$\begin{aligned} s_1^* : \text{T} &= \{u, v\}\text{T}, \\ \neg s_1^* : s_2^* &= \{u, \neg v\}\text{F} \oplus \{u, \neg u\}\text{F}, \\ s_2^* : \text{T} &= \text{N}, \\ \neg s_2^* : s_1^* &= \{u, v\}\text{T} \oplus \{u, \neg v\}\text{F} \oplus \{u, \neg u\}\text{F}, \end{aligned}$$

so that

$$(s_1 \vee s_2)^* = \{u, v\}\text{T} \oplus \{u, \neg v\}\text{F} \oplus \{u, \neg u\}\text{F}.$$

For $(t')^*$ we find

$$(t')^* = ((s_1 \vee s_2) \vee s_3)^* = r_1 \oplus r_2 \oplus r_3 \oplus r_4,$$

where

$$\begin{aligned} r_1 &= (s_1 \vee s_2)^* : \text{T}, \\ r_2 &= \neg(s_1 \vee s_2)^* : s_3^*, \\ r_3 &= s_3^* : \text{T}, \\ r_4 &= \neg s_3^* : (s_1 \vee s_2)^*. \end{aligned}$$

It is easy to derive

$$\begin{aligned} r_1 &= \{u, v\}\text{T}, \\ r_2 &= \{u, \neg v\}\text{F}, \\ r_3 &= \text{N}, \\ r_4 &= \{u, v, \neg v\}\text{T} \oplus \{u, \neg v\}\text{F} \oplus \{u, \neg u, \neg v\}\text{F}. \end{aligned}$$

Using absorption we find

$$(t')^* = \{u, v\}\text{T} \oplus \{u, \neg v\}\text{F}.$$

Now we finish this case using (22).

6. Completeness of conditional composition logic

By the same argument as used above to prove the completeness of $\mathbf{G}_4$ relative to the completeness of $\mathbf{B}_4$, we prove the completeness of $\mathbf{C}_4$ relative to the completeness of $\mathbf{G}_4$.

We repeat here the translations between $\mathbf{G}_4$ and $\mathbf{C}_4$ as defined in Table 4. From $\mathbf{G}_4$ to $\mathbf{C}_4$:

$$\begin{aligned} \text{F}^* &= \text{F}, \\ (\neg x)^* &= \text{F} \lhd x^* \rhd \text{T}, \\ (x \oplus y)^* &= x^* \lhd \text{B} \rhd y^*, \\ (x : y)^* &= y^* \lhd x^* \rhd \text{N}. \end{aligned}$$

And vice versa:

$$\begin{aligned} (x \lhd y \rhd z)' &= y' : x' \oplus \neg y' : z', \\ \text{B}' &= \text{F} \oplus \neg \text{F}, \\ \text{T}' &= \neg \text{F}, \\ \text{F}' &= \text{F}, \\ \text{N}' &= \text{F} : \text{F}. \end{aligned}$$

Variables translate to themselves.

6.1. *Derivation of the* $\mathbf{G}_4$ *axioms in* $\mathbf{C}_4$

We prove for every $\mathbf{G}_4$ axiom $t = u$ that $\mathbf{C}_4 \vdash t^* = u^*$.

Axiom G1 translates to an instance of axiom C8, and G2 translates to an instance of axiom C3. G3 is derived above as (11). G4 translates to an instance of C10. Derivation of G5 is easy by C8, C9. In the case of G6, the left-hand side equals $(z \lhd y \rhd \text{N}) \lhd x \rhd (\text{N} \lhd y \rhd \text{N})$ by definition and by (12). Now apply C2 and again (12). Derivation of G7 is easy using C1 and C7, and derivation of G8 is easy by C3 and (12).

Axiom G9 is derived as follows.

$$\begin{aligned}
x &= \text{T} \lhd x \rhd \text{F} && \text{(by C4)}\\
&= (\text{T} \lhd \text{B} \rhd \text{N}) \lhd x \rhd (\text{N} \lhd \text{B} \rhd \text{F}) && \text{(by C9, C8)}\\
&= (\text{T} \lhd x \rhd \text{N}) \lhd \text{B} \rhd (\text{N} \lhd x \rhd \text{F}) && \text{(by C2)}\\
&= (\text{T} \lhd x \rhd \text{N}) \lhd \text{B} \rhd (\text{F} \lhd (\text{F} \lhd x \rhd \text{T}) \rhd \text{N}) && \text{(by C1, C5, C6)}\\
&= (x : \text{T} \oplus \neg x : \text{F})^* && \text{(definition).}
\end{aligned}$$

G10 translates to an instance of C5; G11 to an instance of C6; and G12 to an instance of (13). G13 and G14 translate to instances of C1 (and of (14)). G15 is easy to derive using axioms C1 and C7. G16 is derived using axioms C1, C5, C6, and C4.

6.2. *Translation invariance*

As in Section 5.2, we apply induction on terms. The cases for the truth constants are trivial. Take $\mathbf{C}_4$ term $t = u \lhd v \rhd w$. We show that $\mathbf{C}_4 \vdash (t')^* = t$. First,

$$t' = v' : u' \oplus \neg v' : w'.$$

Back to $\mathbf{C}_4$:

$$\begin{aligned}
(t')^* &= ((u')^* \lhd (v')^* \rhd \text{N}) \lhd \text{B} \rhd ((w')^* \lhd (\text{F} \lhd (v')^* \rhd \text{T}) \rhd \text{N})\\
&= (u \lhd v \rhd \text{N}) \lhd \text{B} \rhd (w \lhd (\text{F} \lhd v \rhd \text{T}) \rhd \text{N})\\
&= (u \lhd v \rhd \text{N}) \lhd \text{B} \rhd (\text{N} \lhd v \rhd w)\\
&= (u \lhd \text{B} \rhd \text{N}) \lhd v \rhd (\text{N} \lhd \text{B} \rhd w)\\
&= u \lhd v \rhd w,
\end{aligned}$$

using the induction hypothesis, then C1, C5, C6, then C2, and finally C8, C9.

7. Digression

We discuss two matters which fall outside the main line of this article. First, in Section 7.1, we show that $\mathbf{B}_4$ arises naturally from Kleene's three-valued logic [15]. In itself, this is well-known, see [11], but our argument starts from the preservation of the equational theory. Second, in Section 7.2 we discuss some related work on process algebra with many-valued logics.

7.1. *Belnap's logic extends Kleene's logic*

Kleene's three-valued logic [15] — denoted here by $\mathbf{K}_3$ and also known as *partial logic* — has, besides the values *true* (T) and *false* (F), a third truth value *undefined*, for which we shall use the symbol $*$. Negation, disjunction and conjunction are defined by the following truth tables.

$\neg$	
T	F
F	T
$*$	$*$

$\wedge$	T	F	$*$
T	T	F	$*$
F	F	F	F
$*$	$*$	F	$*$

$\vee$	T	F	$*$
T	T	T	T
F	T	F	$*$
$*$	T	$*$	$*$

This logic was designed in order to deal with partial recursive functions: if a partial function f is not defined for argument a, and the truth value of a term ϕ depends on $f(a)$, then ϕ may be classified as $*$. However, a term may still make sense, that is, have a definite truth value, even if it has indefinite subterms; for example, $\mathrm{F} \wedge \phi$ equals F, even if ϕ is classified as $*$. Observe that this intuition also applies to Belnap's truth values B and N. Above we introduced Belnap's logic as the logic characterizing the logical lattice (1). Alternatively, one can start with Kleene's logic and argue that the natural extension of this logic with distinct interpretations of the third truth value while respecting its equational theory is, in fact, Belnap's logic.

Consider extensions of Kleene's logic that are obtained by distinguishing distinct readings of $*$. We require the following:

(1) For an extension with X one of the interpretations of $*$, the subalgebra over $\{\mathrm{T}, \mathrm{F}, \mathrm{X}\}$ should be isomorphic to $\mathbf{K}_3$. For example, this leads to the requirement that $\neg\mathrm{X} = \mathrm{X}$ should be valid for any interpretation X of $*$.

(2) The extension should preserve the equational theory of $\mathbf{K}_3$. If the identity $t = u$ is valid in $\mathbf{K}_3$, and X is one of the interpretations of $*$, then $t[* := \mathrm{X}] = u[* := \mathrm{X}]$ should be valid in the extension. In particular, commutativity, associativity and idempotence of conjunction and disjunction, and absorption and distributivity, are valid in $\mathbf{K}_3$ and should also be valid in the extension. Moreover, negation should remain an involution.

We shall show that Belnap's logic is the only possible extension of $\mathbf{K}_3$ satisfying these requirements.

Let B and N be two distinct readings of the value $*$. It is easy to verify that the requirements lead to the following (incomplete) truth tables:

¬	
B	B
T	F
F	T
N	N

∧	B	T	F	N
B	B	B	F	
T	B	T	F	N
F	F	F	F	F
N		N	F	N

In the following we argue that $\mathrm{B} \wedge \mathrm{N} = \mathrm{N} \wedge \mathrm{B} = \mathrm{F}$ (and hence that $\mathrm{B} \vee \mathrm{N} = \mathrm{N} \vee \mathrm{B} = \mathrm{T}$), and that there are no more than two possible readings of the third truth value $*$. Observe that absorption $(x = x \wedge (x \vee y))$ is valid in $\mathbf{K}_3$, and so are commutativity, associativity and idempotence of conjunction. Now $\mathrm{B} \wedge \mathrm{N} \notin \{\mathrm{B}, \mathrm{N}\}$ by absorption and the identity $\mathrm{B} \vee \mathrm{N} = \neg(\mathrm{B} \wedge \mathrm{N})$. Suppose $\mathrm{B} \wedge \mathrm{N} = \mathrm{N}$, then

$$\mathrm{B} = \mathrm{B} \wedge (\mathrm{B} \vee \mathrm{N}) = \mathrm{B} \wedge \neg(\mathrm{B} \wedge \mathrm{N}) = \mathrm{B} \wedge \neg\mathrm{N} = \mathrm{N}.$$

(In the same way, $\mathrm{B} \wedge \mathrm{N} = \mathrm{B}$ can be refuted.) By associativity and idempotence of conjunction, $\mathrm{B} \wedge \mathrm{N} \neq \mathrm{T}$ (consider $\mathrm{B} \wedge \mathrm{B} \wedge \mathrm{N}$). Now assume that $*$ admits a third interpretation, say X, and $\mathrm{B} \wedge \mathrm{N} = \mathrm{X}$ (and thus $\mathrm{B} \vee \mathrm{N} = \mathrm{X}$). Then we derive $\mathrm{X} = \mathrm{B}$ as follows. First, we have that

$$\mathrm{B} = \mathrm{B} \wedge (\mathrm{B} \vee \mathrm{N}) = \mathrm{B} \wedge \mathrm{X} = \mathrm{X} \wedge \mathrm{B},$$

and hence

$$\mathrm{B} = \neg\mathrm{B} = \neg(\mathrm{B} \wedge \mathrm{X}) = \neg\mathrm{B} \vee \neg\mathrm{X} = \mathrm{B} \vee \mathrm{X} = \mathrm{X} \vee \mathrm{B}.$$

It follows that

$$\mathrm{X} = \mathrm{X} \wedge (\mathrm{X} \vee \mathrm{B}) = \mathrm{X} \wedge \mathrm{B} = \mathrm{B}.$$

This shows that $\mathrm{B} \wedge \mathrm{N} = \mathrm{F}$, and it remains to be shown that with this identity the assumption above, i.e., the existence of a third reading X, is not compatible with B and N. Suppose the contrary. Then, as above, it follows that $\mathrm{B} \wedge \mathrm{X} = \mathrm{N} \wedge \mathrm{X} = \mathrm{F}$. Because distributivity is valid in $\mathbf{K}_3$, we can derive

$$\mathrm{B} = \mathrm{B} \wedge \mathrm{T} = \mathrm{B} \wedge (\mathrm{N} \vee \mathrm{X}) = (\mathrm{B} \wedge \mathrm{N}) \vee (\mathrm{B} \wedge \mathrm{X}) = \mathrm{F} \vee \mathrm{F} = \mathrm{F},$$

which concludes our argument.

7.2. *Process algebra and many-valued logics*

As a second digression, we discuss some related work. This article emerged as a result of research on the combination of process algebra and non-standard propositional logics, see, e.g., [5–8,17]. The main motivation for this research is the characterization of erroneous behavior using propositional logics with non-standard truth values in process algebra with conditional composition.

In [3], Bergstra, Bethke and Rodenburg defined a four-valued propositional logic comprising the special values N (in [3] called D, abbreviating deadlock or divergence) and M (meaningless) and proposed an "information ordering lattice" in which M majorizes T and F, while N is their greatest lower bound. Furthermore, like Fitting in [11], these authors introduced special connectives for the sequential interpretation of $\wedge$ and $\vee$. In particular, *left-sequential conjunction*, notation $\circ\!\!\wedge$, can be motivated as providing an interpretation of conjunction with an operational, sequential reading (for instance suitable to represent lazy, left-sequential evaluation of conditions in imperative programming). Finally, the truth value M represents a catastrophic notion of *meaningless*: typically, $x \wedge \text{M} = x \vee \text{M} = \neg\text{M} = \text{M}$, whereas for instance $\text{F} \circ\!\!\wedge \text{M} = \text{F}$. The truth values N and M can be motivated as covering all types of "errors" that one would want to characterize in error modelling. This four-valued logic, with truth values {M, T, F, N}, is combined with the process algebra ACP [4] in [7], where a strict correspondence between the truth value N and the deadlock process δ was established.

In [6], Bergstra and Ponse defined a five-valued propositional logic with truth values {M, B, T, F, N}, where M majorizes B (in [6] the value B was called C, for 'choice') in the information ordering lattice. Furthermore, in that paper conditional composition is introduced as a logical connective, making left-sequential conjunction, as well as the associated right-sequential and dual connectives, definable:

$$x \circ\!\!\wedge\, y = y \lhd x \rhd \text{F}.$$

The correspondence of logical conditional composition with the programming construct *if-then-else* is the subject of [8] and of our article [17]. In the latter we present a generalization of the process algebra ACP [4] with conditions over $\mathbf{C}_4$. We write

$$P +_\phi Q$$

for *if* ϕ *then* P *else* Q, where P and Q are process terms. Process-algebraic conditional composition is defined by

$$P +_{\text{B}} Q = P + Q,$$
$$P +_{\text{T}} Q = P,$$
$$P +_{\text{F}} Q = Q,$$
$$P +_{\text{N}} Q = \delta.$$

Here $+$ is the well-known ACP operator that stands for *choice* (or *alternative composition*), and δ is the constant that models *inaction* (also called *deadlock*). The intuition is that the choice is non-deterministic if there is evidence for both the truth and the falsity of the condition, and that nothing happens if the value of the condition is undefined, e.g., because the evaluation diverges. This leads to the identification of the summand inclusion ordering $\subseteq$ defined by

$$P \subseteq Q \quad \text{iff} \quad Q = P + Q$$

as the process-algebraic counterpart of the information ordering of Belnap's logic. The process constant δ is the bottom element in the summand inclusion ordering ($x + \delta = x$ is an axiom of ACP). ACP does not include a process constant for the top element of the summand inclusion. Such a constant is Hoare's *chaos* constant χ [12], which is combined with δ in a single framework as the *meaningless* constant μ in [5,7]. In [7], a correspondence is established between the process constant μ and the truth value M discussed above.

8. Conclusion

We have presented the three equationally axiomatized logics $\mathbf{B}_4$, $\mathbf{C}_4$ and $\mathbf{G}_4$ over Belnap's truth values. Here, $\mathbf{B}_4$ has Belnap's classical connectives for conjunction and negation as primitives. This logic has a well-known characterization as a distributive lattice with involution, from which a complete axiomatization follows directly. The logics $\mathbf{C}_4$ and $\mathbf{G}_4$

have a sequential character (i.e., they have sequential connectives as primitives) and are, to our knowledge, new, although the guard connective of $\mathbf{G}_4$ is also discussed by Fitting [11].

We motivated these logics as following naturally from an operational perspective on the evaluation of compound logical propositions. This is worked out further in our article [17] in which we show the correspondence of logical conditional composition with the notion of *choice* in process algebra.

We demonstrated the expressive equivalence between the three logics by providing translations. We proved completeness of the axiomatizations: we started with a completeness proof for $\mathbf{B}_4$ and then based the completeness proof for $\mathbf{G}_4$ on that of $\mathbf{B}_4$, and, consequently, that of $\mathbf{C}_4$ on the completeness of $\mathbf{G}_4$. The logic $\mathbf{G}_4$ thus played a role as an intermediate in which we were able to do quite effective equational reasoning based on a certain type of canonical forms.

References

[1] A.R. Anderson, N.D. Belnap, J.M. Dunn, Entailment: The Logic of Relevance and Necessity, Princeton University Press, 1992.

[2] N.D. Belnap, A useful four-valued logic, in: J.M. Dunn, G. Epstein (Eds.), Modern Uses of Multiple-Valued Logic, D. Reidel, 1977, pp. 8–37.

[3] J.A. Bergstra, I. Bethke, P.H. Rodenburg, A propositional logic with 4 values: True, false, divergent and meaningless, Journal of Applied and Non-Classical Logics 5 (2) (1995) 199–218.

[4] J.A. Bergstra, J.W. Klop, Process algebra for synchronous communication, Information and Control 60 (1–3) (1984) 109–137.

[5] J.A. Bergstra, A. Ponse, Bochvar-McCarthy logic and process algebra, Notre Dame Journal of Formal Logic 39 (4) (1998) 464–484.

[6] J.A. Bergstra, A. Ponse, Process algebra with five-valued logic, in: C.S. Calude, M.J. Dinneen (Eds.), Combinatorics, Computation and Logic, in: Australian Computer Science Communications, vol. 21(3), Springer-Verlag, 1999, pp. 128–143.

[7] J.A. Bergstra, A. Ponse, Process algebra with four-valued logic, Journal of Applied Non-Classical Logics 10 (1) (2000) 27–53.

[8] J.A. Bergstra, A. Ponse, Process algebra and conditional composition, Information Processing Letters 80 (1) (2001) 41–49.

[9] G. Birkhoff, On the structure of abstract algebras, Proceedings of the Cambridge Philosophical Society 31 (4) (1935) 433–454.

[10] E.W. Dijkstra, Cooperating sequential processes, in: F. Genuys (Ed.), Programming Languages, Academic Press, New York, 1968, pp. 43–112.

[11] M.C. Fitting, Kleene's three valued logics and their children, Fundamenta Informaticae 20 (1994) 113–131.

[12] S.D. Brookes, C.A.R. Hoare, A.W. Roscoe, A theory of communicating sequential processes, Journal of the ACM 31 (3) (1984) 560–599.

[13] I.J. Hayes, He Jifeng, C.A.R. Hoare, C.C. Morgan, A.W. Roscoe, J.W. Sanders, I.H. Sorensen, J.M. Spivey, B.A. Sufrin, Laws of programming, Communications of the ACM 3 (8) (1987) 672–686.

[14] J.A. Kalman, Lattices with involution, Transactions of the American Mathematical Society 87 (1958) 485–491.

[15] S.C. Kleene, On a notation for ordinal numbers, Journal of Symbolic Logic 3 (1938) 150–155.

[16] J. McCarthy, A basis for a mathematical theory of computation, in: P. Braffort, D. Hirschberg (Eds.), Computer Programming and Formal Systems, North-Holland, Amsterdam, 1963, pp. 33–70.

[17] A. Ponse, M.B. van der Zwaag, A generalization of ACP using Belnap's logic, Journal of Logic and Algebraic Programming 70 (2) (2007) 222–235.

[18] E.N. Zalta (principal editor), The Stanford Encyclopedia of Philosophy, The Metaphysics Research Lab, Stanford University, 2006. http://plato.stanford.edu/contents.html.