Logic and **L**ogical **P**hilosophy (2022)
DOI: 10.12775/LLP.2022.025
Published online: July 27, 2022

Richard Zach

An Epimorphism between Fine and Ferguson's Matrices for Angell's AC

Abstract. Angell's logic of analytic containment **AC** has been shown to be characterized by a 9-valued matrix **NC** by Ferguson, and by a 16-valued matrix by Fine. It is shown that the former is the image of a surjective homomorphism from the latter, i.e., an epimorphic image. Some candidate 7-valued matrices are ruled out as characteristic of **AC**. Whether matrices with fewer than 9 values exist remains an open question. The results were obtained with the help of the MUltlog system for investigating finite-valued logics; the results serve as an example of the usefulness of techniques from computational algebra in logic. A tableau proof system for **NC** is also provided.

Keywords: analytic containment; many-valued logic; matrix congruence; tableau calculus; computational algebra

1. Introduction

Ferguson [2016] and Fine [2016] have independently provided many-valued matrices which characterize the logic of analytic containment **AC** of [Angell, 1977, 1989]. Ferguson's matrix **NC** has nine truth values, while Fine's matrix **FC** has 16. Fine [2016, fn. 1 on p. 200] suggested that it would be of interest to compare the two. We show below that **NC** is an epimorphic image of **FC**, i.e., **NC** is isomorphic to a factor matrix of **FC**.

Ferguson's **NC** is defined as follows. We start with weak Kleene logic, which has three truth values $K = \{\mathfrak{f}, \mathfrak{u}, \mathfrak{t}\}$. The truth tables for $\wedge$ and $\vee$ are familiar:

Received December 23, 2021. *Accepted* July 24, 2022

$\wedge^{\mathbf{K}_w}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{f}$
$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{t}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{t}$

$\vee^{\mathbf{K}_w}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{t}$
$\mathfrak{f}$	$\mathfrak{f}$	$\mathfrak{u}$	$\mathfrak{t}$
$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$	$\mathfrak{u}$
$\mathfrak{t}$	$\mathfrak{t}$	$\mathfrak{u}$	$\mathfrak{t}$

NC has the truth values $NC = K \times K$, of which $NC^+ = \{\mathfrak{t}\} \times K$ are designated, with the truth functions defined by:

$$\neg^{\mathbf{NC}}(\langle v_0, v_1\rangle) = \langle v_1, v_0\rangle$$
$$\wedge^{\mathbf{NC}}(\langle v_0, v_1\rangle, \langle v_0', v_1'\rangle) = \langle \wedge^{\mathbf{K_w}}(v_0, v_0'), \wedge^{\mathbf{K_w}}(v_1, v_1')\rangle$$
$$\vee^{\mathbf{NC}}(\langle v_0, v_1\rangle, \langle v_0', v_1'\rangle) = \langle \vee^{\mathbf{K_w}}(v_0, v_0'), \vee^{\mathbf{K_w}}(v_1, v_1')\rangle$$

That is, $\wedge$ and $\vee$ are defined component-wise (as in the direct product of two matrices). However, $\neg$ is not the component-wise weak Kleene negation, but instead switches the truth values in a pair.

Dunn and Belnap's matrix for **FDE** has the four truth values $FDE = \{\mathbf{B}, \mathbf{T}, \mathbf{F}, \mathbf{N}\}$ with designated values $FDE^+ = \{\mathbf{B}, \mathbf{T}\}$ and truth functions as follows:[1]

$\neg^{\mathbf{FDE}}$	
B	**B**
T	**F**
F	**T**
N	**N**

$\wedge^{\mathbf{FDE}}$	**B**	**T**	**F**	**N**
B	**B**	**B**	**F**	**F**
T	**B**	**T**	**F**	**N**
F	**F**	**F**	**F**	**F**
N	**F**	**N**	**F**	**N**

$\vee^{\mathbf{FDE}}$	**B**	**T**	**F**	**N**
B	**B**	**T**	**B**	**T**
T	**T**	**T**	**T**	**T**
F	**B**	**T**	**F**	**N**
N	**T**	**T**	**N**	**N**

Fine's matrix is the direct product of **FDE** with another matrix $\mathbf{AC}_2$. $\mathbf{AC}_2$ has four truth values $AC_2 = \{\mathbf{b}, \mathbf{t}, \mathbf{f}, \mathbf{n}\}$ with designated values $AC_2^+ = \{\mathbf{f}, \mathbf{n}\}$ and truth functions as follows:[2]

$\neg^{\mathbf{AC}_2}$	
b	**b**
t	**f**
f	**t**
n	**n**

$\wedge^{\mathbf{AC}_2}$	**b**	**t**	**f**	**n**
b	**b**	**b**	**b**	**b**
t	**b**	**t**	**b**	**t**
f	**b**	**b**	**f**	**f**
n	**b**	**t**	**f**	**n**

$\vee^{\mathbf{AC}_2}$	**b**	**t**	**f**	**n**
b	**b**	**b**	**b**	**b**
t	**b**	**t**	**b**	**t**
f	**b**	**b**	**f**	**f**
n	**b**	**t**	**f**	**n**

(so the truth functions for $\wedge$ and $\vee$ are identical).

The resulting matrix $\mathbf{FC} = \mathbf{FDE} \times \mathbf{AC}_2$ has sixteen truth values and four designated values $FC^+ = FDE^+ \times AC_2^+ = \{\mathbf{Tn}, \mathbf{Bf}, \mathbf{Tf}, \mathbf{Bn}\}$.

[1] We use $\mathbf{B}, \mathbf{T}, \mathbf{F}, \mathbf{N}$ instead of Fine's TF, T$\not$F, $\not$TF, $\not$T$\not$F to save space.

[2] Again, $\mathbf{b}, \mathbf{t}, \mathbf{f}, \mathbf{n}$ correspond to Fine's tf, t$\not$f, $\not$tf, $\not$t$\not$f.

2. Homomorphisms of matrices

A function $f\colon L \to L'$ from the truth values of a matrix $\mathbf{L}$ to those of a matrix $\mathbf{L}'$ is a *strong homomorphism* if it respects truth functions and designated values, that is,

1. $f(\Box^{\mathbf{L}}(v_1, \ldots, v_n)) = \Box^{\mathbf{L}'}(f(v_1), \ldots, f(v_n))$, and
2. $v \in L^+$ iff $f(v) \in L^+$.

A surjective homomorphism is called an *epimorphism*, and a bijective homomorphism an *isomorphism*.

The following facts are elementary results of universal algebra or can easily be verified (see, e.g., Grätzer 1968; Wójcicki 1988):

1. A homomorphism f induces a partition of L consisting of the sets $[v]_f = \{v' \mid f(v) = f(v')\}$.
2. The equivalence relation $\equiv_f$ on L corresponding to the partition induced by a homomorphism f is a congruence of $\mathbf{L}$, that is:
   (a) If $v_1 \equiv_f v_1', \ldots, v_n \equiv_f v_n'$ then $\Box^{\mathbf{L}}(v_1, \ldots, v_n) \equiv_f \Box^{\mathbf{L}}(v_1', \ldots, v_n')$, and
   (b) if $v \equiv_f v'$ then $v \in L^+$ iff $v' \in L^+$.
3. If $\equiv$ is a congruence on $\mathbf{L}$, then $f\colon L \to L/{\equiv}$ defined by $f(v) = [v]_f$ is an epimorphism of $\mathbf{L}$ to the factor matrix $\mathbf{L}/{\equiv}$.
4. Every epimorphism $h\colon \mathbf{L} \to \mathbf{L}'$ factors into the epimorphism $f\colon \mathbf{L} \to \mathbf{L}/{\equiv_h}$ and an isomorphism $g\colon \mathbf{L}/{\equiv_h} \to \mathbf{L}'$.
5. If $f\colon \mathbf{L} \to \mathbf{L}'$ is an epimorphism then the consequence relations of $\mathbf{L}$ and $\mathbf{L}'$ (defined via preservation of designated values) agree, i.e., $\Delta \vDash_{\mathbf{L}} \varphi$ iff $\Delta \vDash_{\mathbf{L}'} \varphi$.

3. The epimorphism of FC to NC

In light of the above, to show that **NC** is an epimorphic image of **FC**, and hence that $\vDash_{\mathbf{NC}}$ coincides with $\vDash_{\mathbf{FC}}$, it suffices to find a congruence relation $\equiv$ of **FC** and an isomorphism between $\mathbf{FC}/{\equiv}$ and **NC**.

FC has two non-trivial congruences, namely:

$$\mathbf{Bt}, \mathbf{Ff}, \mathbf{Fn}, \mathbf{Ft}, \mathbf{Nf}, \mathbf{Nn}, \mathbf{Nt}, \mathbf{Bb} \equiv \mathbf{Fb} \equiv \mathbf{Nb} \equiv \mathbf{Tb}, \mathbf{Tt}, \mathbf{Bf}, \mathbf{Bn}, \mathbf{Tf}, \mathbf{Tn}$$

$$\mathbf{Nn}, \mathbf{Ff} \equiv \mathbf{Nf}, \mathbf{Fn}, \mathbf{Nt} \equiv \mathbf{Tt}, \mathbf{Bb} \equiv \mathbf{Fb} \equiv \mathbf{Nb} \equiv \mathbf{Tb}, \mathbf{Bt} \equiv \mathbf{Ft}, \mathbf{Tn}, \mathbf{Bf} \equiv \mathbf{Tf}, \mathbf{Bn}$$

$\neg_{\mathbf{NC}}$	
$\mathfrak{ff}$	$\mathfrak{ff}$
$\mathfrak{fu}$	$\mathfrak{uf}$
$\mathfrak{ft}$	$\boxed{\mathfrak{tf}}$
$\mathfrak{uf}$	$\mathfrak{fu}$
$\mathfrak{uu}$	$\mathfrak{uu}$
$\mathfrak{ut}$	$\boxed{\mathfrak{tu}}$
$\boxed{\mathfrak{tf}}$	$\mathfrak{ft}$
$\boxed{\mathfrak{tu}}$	$\mathfrak{ut}$
$\boxed{\mathfrak{tt}}$	$\boxed{\mathfrak{tt}}$

$\neg_{\mathbf{FC}/\equiv}$	
Nn	**Nn**
FNf	**NTt**
Fn	$\boxed{\textbf{Tn}}$
NTt	**FNf**
B·Nb	**B·Nb**
BFt	$\boxed{\textbf{BTf}}$
$\boxed{\textbf{Tn}}$	**Fn**
$\boxed{\textbf{BTf}}$	**BFt**
$\boxed{\textbf{Bn}}$	$\boxed{\textbf{Bn}}$

$\neg_{\mathbf{FC}}$	
Nn	**Nn**
Ff	**Tt**
Nf	**Nt**
Fn	$\boxed{\textbf{Tn}}$
Nt	**Nf**
Tt	**Ff**
Fb	**Tb**
Nb	**Nb**
Bb	**Bb**
Tb	**Fb**
Ft	$\boxed{\textbf{Tf}}$
Bt	$\boxed{\textbf{Bf}}$
$\boxed{\textbf{Tn}}$	**Fn**
$\boxed{\textbf{Bf}}$	**Bt**
$\boxed{\textbf{Tf}}$	**Ft**
$\boxed{\textbf{Bn}}$	$\boxed{\textbf{Bn}}$

Table 1. $\neg$ in **NC**, **FC**$/\!\equiv$, and **FC**

The latter has nine equivalence classes, of which the three classes $\{\textbf{Tn}\}$, $\{\textbf{Bf}, \textbf{Tf}\}$, and $\{\textbf{Bn}\}$ are designated. And in fact, the following mapping is an isomorphism between **NC** and **FC**$/\!\equiv$:

v	$\mathfrak{ff}$	$\mathfrak{fu}$	$\mathfrak{ft}$	$\mathfrak{uf}$	$\mathfrak{uu}$	$\mathfrak{ut}$	$\mathfrak{tf}$	$\mathfrak{tu}$	$\mathfrak{tt}$
$g(v)$	**Nn**	**Ff**, **Nf**	**Fn**	**Nt**, **Tt**	**Bb**, **Fb**, **Nb**, **Tb**	**Bt**, **Ft**	**Tn**	**Bf**, **Tf**	**Bn**
		FNf		**NTt**	**B·Nb**	**BFt**		**BTf**	

The last line introduces abbreviations for the congruence classes of **FC**$/\!\equiv$ consisting of more than one truth value. We use these abbreviations merely to save space in tables 1 and 4 below.

The verification of the facts that $\equiv$ is a congruence on **FC** and that g is an isomorphism between **NC** and **FC**$/\!\equiv$ would be extremely tedious. We can make them immediately apparent, however, by displaying the truth tables for each logic in full, with truth values that are isomorphic or equivalent shown in the same color, and by arranging the truth values in corresponding order. For instance, the truth tables for $\neg$ in **NC**, **FC**$/\!\equiv$ and **FC**, respectively, are given in table 1. Reversed colors indicate the designated values. Compare the tables for $\wedge$ in **NC** and **FC** in table 2, and those for $\vee$ in **NC** and **FC** in table 3.

$\wedge^{\mathbf{NC}}$	ff	fu	ft	uf	uu	ut	tf	tu	tt
ff	ff	fu	ft	uf	uu	ut	ff	fu	ft
fu	fu	fu	fu	uu	uu	uu	fu	fu	fu
ft	ft	fu	ft	ut	uu	ut	ft	fu	ft
uf	uf	uu	ut	uf	uu	ut	uf	uu	ut
uu	uu	uu	uu	uu	uu	uu	uu	uu	uu
ut	ut	uu	ut	ut	uu	ut	ut	uu	ut
tf	ff	fu	ft	uf	uu	ut	tf	tu	tt
tu	fu	fu	fu	uu	uu	uu	tu	tu	tu
tt	ft	fu	ft	ut	uu	ut	tt	tu	tt

$\wedge^{\mathbf{FC}}$	Nn	Ff	Nf	Fn	Nt	Tt	Fb	Nb	Bb	Tb	Ft	Bt	Tn	Bf	Tf	Bn
Nn	Nn	Ff	Nf	Fn	Nt	Nt	Fb	Nb	Fb	Nb	Ft	Ft	Nn	Ff	Nf	Fn
Ff	Ff	Ff	Ff	Ff	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Ff	Ff	Ff	Ff
Nf	Nf	Ff	Nf	Ff	Nb	Nb	Fb	Nb	Fb	Nb	Fb	Fb	Nf	Ff	Nf	Ff
Fn	Fn	Ff	Ff	Fn	Ft	Ft	Fb	Fb	Fb	Fb	Ft	Ft	Fn	Ff	Ff	Fn
Nt	Nt	Fb	Nb	Ft	Nt	Nt	Fb	Nb	Fb	Nb	Ft	Ft	Nt	Fb	Nb	Ft
Tt	Nt	Fb	Nb	Ft	Nt	Tt	Fb	Nb	Bb	Tb	Ft	Bt	Tt	Bb	Tb	Bt
Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb	Fb
Nb	Nb	Fb	Nb	Fb	Nb	Nb	Fb	Nb	Fb	Nb	Fb	Fb	Nb	Fb	Nb	Fb
Bb	Fb	Fb	Fb	Fb	Fb	Bb	Fb	Fb	Bb	Bb	Fb	Bb	Bb	Bb	Bb	Bb
Tb	Nb	Fb	Nb	Fb	Nb	Tb	Fb	Nb	Bb	Tb	Fb	Bb	Tb	Bb	Tb	Bb
Ft	Ft	Fb	Fb	Ft	Ft	Ft	Fb	Fb	Fb	Fb	Ft	Ft	Ft	Fb	Fb	Ft
Bt	Ft	Fb	Fb	Ft	Ft	Bt	Fb	Fb	Bb	Bb	Ft	Bt	Bt	Bb	Bb	Bt
Tn	Nn	Ff	Nf	Fn	Nt	Tt	Fb	Nb	Bb	Tb	Ft	Bt	Tn	Bf	Tf	Bn
Bf	Ff	Ff	Ff	Ff	Fb	Bb	Fb	Fb	Bb	Bb	Fb	Bb	Bf	Bf	Bf	Bf
Tf	Nf	Ff	Nf	Ff	Nb	Tb	Fb	Nb	Bb	Tb	Fb	Bb	Tf	Bf	Tf	Bf
Bn	Fn	Ff	Ff	Fn	Ft	Bt	Fb	Fb	Bb	Bb	Ft	Bt	Bn	Bf	Bf	Bn

Table 2. $\wedge$ in **NC** and **FC**. Values in any rectangle are equivalent and correspond to a single value in **NC**.

4. Complexity

The questions of whether a matrix is an epimorphic image of another, and whether two matrices are isomorphic, are computationally non-trivial to answer. The number of different bijective mappings between two n-valued matrices equals the number of possible permutations of the n truth values, i.e., $n!$. However, since a potential isomorphism has to respect the designated truth values, the number of maps to be checked is actually "only" $k!(n-k)!$ where k is the number of designated values. In

$\vee^{\mathbf{NC}}$	ff	fu	ft	uf	uu	ut	tf	tu	tt
ff	ff	fu	ff	uf	uu	uf	tf	tu	tf
fu	fu	fu	fu	uu	uu	uu	tu	tu	tu
ft	ff	fu	ft	uf	uu	ut	tf	tu	tt
uf	uf	uu	uf	uf	uu	uf	uf	uu	uf
uu	uu	uu	uu	uu	uu	uu	uu	uu	uu
ut	uf	uu	ut	uf	uu	ut	uf	uu	ut
tf	tf	tu	tf	uf	uu	uf	tf	tu	tf
tu	tu	tu	tu	uu	uu	uu	tu	tu	tu
tt	tf	tu	tt	uf	uu	ut	tf	tu	tt

$\vee^{\mathbf{FC}}$	Nn	Ff	Nf	Fn	Nt	Tt	Fb	Nb	Bb	Tb	Ft	Bt	Tn	Bf	Tf	Bn
Nn	Nn	Nf	Nf	Nn	Nt	Tt	Nb	Nb	Tb	Tb	Nt	Tt	Tn	Tf	Tf	Tn
Ff	Nf	Ff	Nf	Ff	Nb	Tb	Fb	Nb	Bb	Tb	Fb	Bb	Tf	Bf	Tf	Bf
Nf	Nf	Nf	Nf	Nf	Nb	Tb	Nb	Nb	Tb	Tb	Nb	Tb	Tf	Tf	Tf	Tf
Fn	Nn	Ff	Nf	Fn	Nt	Tt	Fb	Nb	Bb	Tb	Ft	Bt	Tn	Bf	Tf	Bn
Nt	Nt	Nb	Nb	Nt	Nt	Tt	Nb	Nb	Tb	Tb	Nt	Tt	Tt	Tb	Tb	Tt
Tt	Tt	Tb	Tb	Tt	Tt	Tt	Tb	Tb	Tb	Tb	Tt	Tt	Tt	Tb	Tb	Tt
Fb	Nb	Fb	Nb	Fb	Nb	Tb	Fb	Nb	Bb	Tb	Fb	Bb	Tb	Bb	Tb	Bb
Nb	Nb	Nb	Nb	Nb	Nb	Tb	Nb	Nb	Tb	Tb	Nb	Tb	Tb	Tb	Tb	Tb
Bb	Tb	Bb	Tb	Bb	Tb	Tb	Bb	Tb	Bb	Tb	Bb	Bb	Tb	Bb	Tb	Bb
Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb
Ft	Nt	Fb	Nb	Ft	Nt	Tt	Fb	Nb	Bb	Tb	Ft	Bt	Tt	Bb	Tb	Bt
Bt	Tt	Bb	Tb	Bt	Tt	Tt	Bb	Tb	Bb	Tb	Bt	Bt	Tt	Bb	Tb	Bt
Tn	Tn	Tf	Tf	Tn	Tt	Tt	Tb	Tb	Tb	Tb	Tt	Tt	Tn	Tf	Tf	Tn
Bf	Tf	Bf	Tf	Bf	Tb	Tb	Bb	Tb	Bb	Tb	Bb	Bb	Tf	Bf	Tf	Bf
Tf	Tf	Tf	Tf	Tf	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tb	Tf	Tf	Tf	Tf
Bn	Tn	Bf	Tf	Bn	Tt	Tt	Bb	Tb	Bb	Tb	Bt	Bt	Tn	Bf	Tf	Bn

Table 3. $\vee$ in **NC** and **FC**

our case, this amounts to $3! \times 6! = 4{,}320$ potential isomorphisms; already too large to be checked manually by brute force. To find a candidate epimorphic image in the first place, an exhaustive search would have to enumerate all potential partitions and check if they are congruences. The number of partitions of a size k set is B_k, the k-th Bell number. Since congruences have to again respect the designated values, the number of potential congruences to be checked can be reduced to $B_k B_{n-k}$ (for an n-valued matrix with k designated values). In the case of **FC**, there are

$\wedge^{\mathbf{FC}/\equiv}$	Nn	FNf	Fn	NTt	B·Nb	BFt	Tn	BTf	Bn
Nn	Nn	FNf	Fn	NTt	B·Nb	BFt	Nn	FNf	Fn
FNf	FNf	FNf	FNf	B·Nb	B·Nb	B·Nb	FNf	FNf	FNf
Fn	Fn	FNf	Fn	BFt	B·Nb	BFt	Fn	FNf	Fn
NTt	NTt	B·Nb	BFt	NTt	B·Nb	BFt	NTt	B·Nb	BFt
B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb
BFt	BFt	B·Nb	BFt	BFt	B·Nb	BFt	BFt	B·Nb	BFt
Tn	Nn	FNf	Fn	NTt	B·Nb	BFt	Tn	BTf	Bn
BTf	FNf	FNf	FNf	B·Nb	B·Nb	B·Nb	BTf	BTf	BTf
Bn	Fn	FNf	Fn	BFt	B·Nb	BFt	Bn	BTf	Bn

$\vee^{\mathbf{FC}/\equiv}$	Nn	FNf	Fn	NTt	B·Nb	BFt	Tn	BTf	Bn
Nn	Nn	FNf	Nn	NTt	B·Nb	NTt	Tn	BTf	Tn
FNf	FNf	FNf	FNf	B·Nb	B·Nb	B·Nb	BTf	BTf	BTf
Fn	Nn	FNf	Fn	NTt	B·Nb	BFt	Tn	BTf	Bn
NTt	NTt	B·Nb	NTt	NTt	B·Nb	NTt	NTt	B·Nb	NTt
B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb	B·Nb
BFt	NTt	B·Nb	BFt	NTt	B·Nb	BFt	NTt	B·Nb	BFt
Tn	Tn	BTf	Tn	NTt	B·Nb	NTt	Tn	BTf	Tn
BTf	BTf	BTf	BTf	B·Nb	B·Nb	B·Nb	BTf	BTf	BTf
Bn	Tn	BTf	Bn	NTt	B·Nb	BFt	Tn	BTf	Bn

Table 4. $\wedge$ and $\vee$ in $\mathbf{FC}/\equiv$

$15 \times 4{,}213{,}597 = 63{,}203{,}955$ potential congruences. Thus, finding all epimorphic images of **FC**, and verifying that **NC** is one, is impossible to do by brute force without the help of a computer.

Note that the approach taken is already much better than the completely naive approach of checking every surjective function from $FDE \times AC_2$ to NC for whether it is a homomorphism. First of all, this would require the verification of many more candidate mappings. The number of different surjective functions from an n-element set to one of size m is $m!S(n,m)$ (where $S(n,m)$ are the Stirling numbers of the second kind). Since any epimorphism must respect designated values, we can restrict the possible values for designated and non-designated arguments. The total number of candidates in our case would be $6!S(12,6) \times 3!S(4,3)$, or over 34 billion. Our approach also provides more information: since **FC** has no congruence with fewer than 9 classes, there can be no smaller common factor of both **NC** and **FC**. This cannot be ruled out a priori, and would have yielded an interesting result—a matrix with the same consequence relation as **NC** and **FC**, but fewer truth values than either. Merely checking if an epimorphism from **FC** to **NC** exists would not have settled that question.

It is in practice not necessary to run through the entirety of all partitions or all bijections to find congruences and isomorphisms, respectively. A simple idea will cut down the search space to manageable size. Here's the idea for isomorphisms: call an injection $f\colon U \to V'$ (where $U \subseteq V$, and V, V' are the truth value sets of $\mathbf{L}$ and $\mathbf{L}'$, respectively) a *partial isomorphism* if it respects operations, i.e., if $\Box^{\mathbf{L}'}(f(v_1), \ldots, f(v_n)) = f(\Box^{\mathbf{L}}(v_1, \ldots, v_n))$ provided v_i and $\Box(v_1, \ldots, v_n) \in \mathrm{dom}(f)$.[3] A bijective $f\colon V \to V'$ is an isomorphism iff $f \restriction U$ is a partial isomorphism for all $U \subseteq V$. So to find an isomorphism we can proceed as follows: Set $f_0 = \emptyset$ and let $f_{i+1} = f_i \cup (v, v')$ where $v \in V \setminus \mathrm{dom}(f_i)$ and $v' \in V' \setminus \mathrm{ran}(f_i)$. If f_i is not a partial isomorphism, no expansion of f_i can be an isomorphism; in that case, backtrack and pick a different pair (v, v'). In other words, instead of generating and testing all isomorphisms, generate isomorphisms one value at a time. If a particular choice of value results in a conflict with the truth tables, it is guaranteed that no expansion of that sequence of choices to a total bijection is an isomorphism. A similar idea can be used to speed up the search for congruences. Nevertheless, even if there are just a handful of partial isomorphisms to test, this is intractable by hand. For instance, verifying that **NC** with its modest 9 truth values and three operations has no nontrivial automorphisms requires checking 27 partial isomorphisms and computing almost 1,400 individual operations.

5. Application of MUltlog

Finding the congruences of **FC**, i.e., its potential factor matrices, and the verification that $\mathbf{FC}/{\equiv}$ is isomorphic to **NC**, was accomplished using the MUltlog system due to [Salzer, 1996].[4] MUltlog was originally

[3] It must also respect designated values, but we can simply split the bijection into a bijection between designated values and one between undesignated values as before.

[4] The MUltlog software is available at logic.at/multlog. Results reported in this paper were generated on version 1.16a [Salzer et al., 2022]. The supplementary code used is available at github.com/rzach/ncac and is archived with DOI: `10.5281/zenodo.6910023`.

designed to compute optimized n-sided sequent calculus rules for arbitrary n-valued matrices using the methods of [Baaz et al., 1993] and [Salzer, 2000]. As of version v1.5, MUltlog added interactive functionality, including: evaluating formulas in a matrix, testing for and finding tautologies of matrices, defining products and factors of matrices, finding congruences of matrices, and checking for isomorphism between matrices.[5] The solution of Fine's question presented above was found by:

1. Specifying the matrices for **NC**, **FDE**, and $\mathbf{AC}_2$ in MUltlog's format;[6]
2. Defining the product **FC**;
3. Finding the epimorphism from **FC** to **NC**.

Only the specification of **NC** required substantive intervention. MUltlog provided the basis for the LaTeX code of the truth tables with matching orders of truth values and matching colors.[7] The algorithms used are still relatively naive. Much more efficient algorithms exist to solve questions like this, e.g., see [Freese, 2008] for a fast algorithm to find a minimal congruence of a finite algebra. The naive methods, however, have the advantage of being straightforwardly implemented in Prolog, and an extension of MUltlog provides a convenient way to operate on finite-valued matrices.

MUltlog's original purpose, as mentioned, is to compute inference rules for n-valued logics. It does this not just for propositional operators but also for so-called distribution quantifiers. Any associative, commutative, idempotent binary operation induces such a quantifier. E.g., if in a given interpretation, the formula $A(x)$ takes all and only values in $\{v_1, \ldots, v_n\}$ then the induced quantifier $\forall x\, A(x)$ of, say, $\wedge$, takes the value $v_1 \wedge v_2 \wedge \cdots \wedge v_n$. Since **NC**'s $\wedge$ and $\vee$ are associative, commutative, and idemptotent, **NC** has universal and existential quantifiers that generalize these connectives. Their truth tables are unwieldy, as they list the value of $\forall x\, A(x)$ for all potential distributions of $A(x)$, i.e., all $2^9 - 1 = 511$ possible non-empty subsets of NC.[8] MUltlog nevertheless

[5] Implementation of these features was carried out, for the most part, by the author.

[6] See appendix A.

[7] Finding the epimorphism between **FC** and **NC** required less than a second. Checking all potential congruences of **FC** took about 20 minutes. So had we not had the candidate matrix **NC** available, we could have found it automatically. The computations were carried out on an Asus Zenbook 14 (Intel Core i7–8565U CPU, 16GB memory), running Ubuntu Linux 21.10 and SWI-Prolog 8.2.4.

[8] All but 113 of which result in the value $\mathfrak{uu}$.

finds optimal inference rules in minutes. The tableaux calculus for **NC** generated by MUltlog can be found in appendix B.[9]

6. The mystery of the 7-valued matrix

Fine [2016, p. 223] suggested that "The 16 values [of **FC**] can, in fact, be reduced to 7 since we may just differentiate one designated value, $\mathbf{B}$ or $\mathbf{T}$, into four values, when paired with the values $\mathbf{b}$, $\mathbf{t}$, $\mathbf{f}$, or $\mathbf{n}$." It is not perfectly clear what he has in mind here, but one way of making it precise is this: If we let V be either $\mathbf{B}$ or $\mathbf{T}$, then we can differentiate V into the four values $V\mathbf{b}$, $V\mathbf{t}$, $V\mathbf{f}$, and $V\mathbf{n}$. The remaining three truth values in $\{\mathbf{B}, \mathbf{T}, \mathbf{F}, \mathbf{N}\} \setminus \{V\}$ would then be paired with a fixed value $v \in \{\mathbf{b}, \mathbf{t}, \mathbf{f}, \mathbf{n}\}$, resulting in 7 values altogether. In other words, the truth values FC_{Vv} of $\mathbf{FC}_{Vv}$ are

$$FC_{Vv} = (\{V\} \times AC_2) \cup ((FDE \setminus \{V\}) \times \{v\}).$$

The truth functions of the corresponding matrix $\mathbf{FC}_{Vv}$ would be given by

$$\begin{aligned} \neg^{\mathbf{FC}_{Vv}}(\langle v_0, v_1\rangle) &= h_{Vv}(\langle \neg^{\mathbf{FDE}}(v_0), \neg^{\mathbf{AC}_2}(v_1)\rangle) \\ \wedge^{\mathbf{FC}_{Vv}}(\langle v_0, v_1\rangle, \langle v_0', v_1'\rangle) &= h_{Vv}(\langle \wedge^{\mathbf{FDE}}(v_0, v_0'), \wedge^{\mathbf{AC}_2}(v_1, v_1')\rangle) \\ \vee^{\mathbf{FC}_{Vv}}(\langle v_0, v_1\rangle, \langle v_0', v_1'\rangle) &= h_{Vv}(\langle \vee^{\mathbf{FDE}}(v_0, v_0'), \vee^{\mathbf{AC}_2}(v_1, v_1')\rangle) \end{aligned}$$

where the function h_{Vv} guarantees that the result is in the corresponding set of truth values, by letting

$$h_{Vv}(\langle v_0, v_1\rangle) = \begin{cases} \langle v_0, v_1\rangle & \text{if } v_0 = V \\ \langle v_0, v\rangle & \text{otherwise.} \end{cases}$$

In other words, $\mathbf{FC}_{Vv}$ is like **FC**, except that only V is paired with all four truth values of $\mathbf{AC}_2$, and the remaining truth values of **FDE** are each paired with the single value v of $\mathbf{AC}_2$. The value v is "dominant," in the sense that if the value of one of the truth functions of **FC** would yield a value $\langle v_0, v_1\rangle$ where $v_0 \neq V$ and $v_1 \neq v$, then the value taken is $\langle v_0, v\rangle$ instead of $\langle v_0, v_1\rangle$.

This leaves the question of which truth values to designate in $\mathbf{FC}_{Vv}$. It is most natural to let $FC^+_{Vv} = FC^+ \cap FC_{Vv}$, i.e., a truth value is designated in $\mathbf{FC}_{Vv}$ iff it is designated in **FC**. This gives us eight different

[9] Ferguson [2021] has recently given a much more elegant tableau system for **NC**.

matrices, one for each of the possible ways of combining either B or T with one of the four values of $\mathbf{AC}_2$.

Let $V' = $ B if $V = $ T, and $=$ T if $V = $ B (so, the designated value of **FDE** that's not V). If $v \in AC_2^+$, i.e., if $v = $ f or $=$ n, then $V'v$ is designated in $\mathbf{FC}_{Vv}$, and otherwise it is not. But $V'v$ has to play the role of *all* the pairs $V'u$ with $u \in AC_2$. So we should consider designating $V'v$ even when $v \notin AC_2^+$, and also consider *not* designating it when $v \in AC^+$. So, let the matrix $\mathbf{FC}^*_{Vv}$ be just like $\mathbf{FC}_{Vv}$ except that $V'v$ is designated if $v = $ t or $=$ b, or undesignated if $v = $ f or $=$ n. For instance, in $\mathbf{FC_{Tb}}$, the truth value Bb is not designated (since b $\notin AC^+$). But, e.g, the designated value Bf $\in FC^+$ doesn't exist in $\mathbf{FC_{Tb}}$. If Bb has to play its role in $\mathbf{FC_{Tb}}$, it would have to be designated after all. So in $\mathbf{FC^*_{Tb}}$, the designated values are Tf, Tn, and Bb. Conversely, if $v \in AC_2^+$, the value $V'v$ is designated in $\mathbf{FC}_{Vv}$. We can choose to not designate it, since it has to play the role of all values $V'u$. E.g., the designated values of $\mathbf{FC^*_{Tf}}$ are Tf, Tn, but not Bf. In general, if $v \in AC^+$, then $\mathbf{FC}_{Vv}$ has three designated values while $\mathbf{FC}^*_{Vv}$ has two; otherwise $\mathbf{FC}_{Vv}$ has two designated values and $\mathbf{FC}^*_{Vv}$ has three. This results in another eight matrices.

For all of the resulting logics other than the four corresponding to Bt and Bb, already one of the axioms of **AC** given in [Ferguson, 2016] fails. For instance, in $\mathbf{FC^*_{Tf}}$, Tf is designated but $\neg\neg$Tf $=$ Tt is not, so $A \nvDash \neg\neg A$ and thus axiom AC1a fails.

The 7-valued logics $\mathbf{FC_{Bb}}$, $\mathbf{FC^*_{Bb}}$, $\mathbf{FC_{Bt}}$, and $\mathbf{FC^*_{Bt}}$, however, are sound for those axioms. To show that here the reverse inclusions do not hold, we have to find examples of entailments $\varphi \models \psi$ that hold in the logics considered but not in **AC**, and this can be done by finding countervaluations in **NC**. For $V = $ B and $v = $ t or $v = $ b we have $A \vee B \vDash_{\mathbf{FC}_{Vv}} B$ and $A \wedge (B \vee \neg A) \vDash_{\mathbf{FC}^*_{Vv}} B$. Neither consequence holds in **NC**. For the first, observe that $\mathfrak{tf} \vee \mathfrak{ff} = \mathfrak{tf}$ is designated in **NC**, but $\mathfrak{ff}$ is not. For the second, we have $\mathfrak{tt} \wedge (\mathfrak{ff} \vee \neg\mathfrak{tt}) = \mathfrak{tt}$ is designated in **NC**, but $\mathfrak{ff}$, again, isn't.[10] In short, none of the sixteen potential 7-valued matrices in fact characterize **AC**.

Whether a matrix with fewer than **NC**'s nine truth values characterizes **AC** remains an open question. Fine's suggested 7-valued matrices do not. If a smaller matrix exists, it will have to be proved to be adequate by independent methods, e.g., by a direct soundness and completeness

[10] These counterexamples were found by MUltlog as well. $\mathbf{AC}_2$ happens to be an epimorphic image of $\mathbf{FC_{Bb}}$. No two of the resulting logics are isomorphic, and other than $\mathbf{FC_{Bb}}$, none of them have epimorphisms to **FDE** or $\mathbf{AC}_2$.

proof (they cannot be epimorphic images of **NC** or **FC**, as these have no congruences with fewer than nine classes).

Acknowledgments. I would like to thank an anonymous referee for their helpful suggestions.

References

Angell, Richard Bradshaw, 1977, "Three systems of first degree entailment (abstract)", *The Journal of Symbolic Logic* 42 (1): 147. DOI: 10.2307/2272332

Angell, Richard Bradshaw, 1989, "Deducibility, entailment and analytic containment", pages 119–143 in J. Norman and R. Sylvan (eds.), *Directions in Relevant Logic*, Dordrecht: Kluwer. DOI: 10.1007/978-94-009-1005-8_8

Baaz, Matthias, Christian G. Fermüller, and Richard Zach, 1993, "Elimination of cuts in first-order finite-valued logics", *Journal of Information Processing and Cybernetics EIK* 29 (6): 333–355. DOI: 10.11575/PRISM/38801

Ferguson, Thomas Macaulay, 2016, "Faulty Belnap computers and subsystems of FDE", *Journal of Logic and Computation* 26 (5): 1617–1636. DOI: 10.1093/logcom/exu048

Ferguson, Thomas Macaulay, 2021, "Tableaux and restricted quantification for systems related to weak Kleene logic", pages 3–19 in A. Das and S. Negri (eds.), *Automated Reasoning with Analytic Tableaux and Related Methods*, no. 12842 in Lecture Notes in Computer Science, Cham: Springer. DOI: 10.1007/978-3-030-86059-2_1

Fine, Kit, 2016, "Angellic content", *Journal of Philosophical Logic* 45 (2): 199–226. DOI: 10.1007/s10992-015-9371-9

Freese, Ralph, 2008, "Computing congruences efficiently", *Algebra Universalis* 59 (3): 337–343. DOI: 10.1007/s00012-008-2073-1

Grätzer, George, 1968, *Universal Algebra*, Princeton, NJ: van Nostrand.

Hähnle, Reiner, 1993, *Automated Deduction in Multiple-Valued Logics*, Oxford: Oxford University Press.

Salzer, Gernot, 1996, "MUltlog: An expert system for multiple-valued logics", pages 50–55 in *Collegium Logicum*, Collegium Logicum, Vienna: Springer. DOI: 10.1007/978-3-7091-9461-4_3

Salzer, Gernot, 2000, "Optimal axiomatizations of finitely valued logics", *Information and Computation* 162 (1–2): 185–205. DOI: 10.1006/inco.1999.2862

Salzer, Gernot et al., 2022, MUltlog v1.16a. Software program. https://github.com/rzach/multlog. DOI: 10.5281/zenodo.6893267

Wójcicki, Ryszard, 1988, *Theory of Logical Calculi: Basic Theory of Consequence Operations*, No. 147 in Synthese Library, Dordrecht: Kluwer.

Richard Zach
Department of Philosophy
University of Calgary, Canada
rzach@ucalgary.ca richardzach.org

A. MUltlog specification files

Specification of NC

```
logic "NC".

truth_values { ff , fu , ft , uf , uu , ut , tf , tu , tt }.
designated_truth_values { tf , tu , tt }.

operator(neg/1, mapping {
          (ff) : ff,
          (fu) : uf,
          (ft) : tf,
          (uf) : fu,
          (uu) : uu,
          (ut) : tu,
          (tf) : ft,
          (tu) : ut,
          (tt) : tt
        }
).

operator(and/2, table [
                ff, fu, ft, uf, uu, ut, tf, tu, tt,
          ff,   ff, fu, ft, uf, uu, ut, ff, fu, ft,
          fu,   fu, fu, fu, uu, uu, uu, fu, fu, fu,
          ft,   ft, fu, ft, ut, uu, ut, ft, fu, ft,
          uf,   uf, uu, ut, uf, uu, ut, uf, uu, ut,
          uu,   uu, uu, uu, uu, uu, uu, uu, uu, uu,
          ut,   ut, uu, ut, ut, uu, ut, ut, uu, ut,
          tf,   ff, fu, ft, uf, uu, ut, tf, tu, tt,
          tu,   fu, fu, fu, uu, uu, uu, tu, tu, tu,
          tt,   ft, fu, ft, ut, uu, ut, tt, tu, tt
        ]
).
```

```
operator(or/2, table [
                 ff, fu, ft, uf, uu, ut, tf, tu, tt,
            ff,  ff, fu, ff, uf, uu, uf, tf, tu, tf,
            fu,  fu, fu, fu, uu, uu, uu, tu, tu, tu,
            ft,  ff, fu, ft, uf, uu, ut, tf, tu, tt,
            uf,  uf, uu, uf, uf, uu, uf, uf, uu, uf,
            uu,  uu, uu, uu, uu, uu, uu, uu, uu, uu,
            ut,  uf, uu, ut, uf, uu, ut, uf, uu, ut,
            tf,  tf, tu, tf, uf, uu, uf, tf, tu, tf,
            tu,  tu, tu, tu, uu, uu, uu, tu, tu, tu,
            tt,  tf, tu, tt, uf, uu, ut, tf, tu, tt
          ]
).

quantifier(forall, induced_by and/2).
quantifier(exists, induced_by or/2).
```

Specification of FDE

```
logic "FDE".

truth_values { b, t, f, n }.
designated_truth_values { b, t }.

ordering(truth, "f < {n, b} < t").

operator(neg/1, mapping {
           (t) : f,
           (b) : b,
           (n) : n,
           (f) : t
          }
).

operator(and/2, inf(truth)).
operator(or/2, sup(truth)).
```

Specification of AC_2

```
logic "AC2".

truth_values { b, t, f, n }.
designated_truth_values { n , f }.

operator(neg/1, mapping {
```

```
          (t) : f,
          (b) : b,
          (n) : n,
          (f) : t
        }
).

operator(or/2, table [
              t, b, n, f,
          t,  t, b, t, b,
          b,  b, b, b, b,
          n,  t, b, n, f,
          f,  b, b, f, f
        ]
).

operator(and/2, table [
              t, b, n, f,
          t,  t, b, t, b,
          b,  b, b, b, b,
          n,  t, b, n, f,
          f,  b, b, f, f
        ]
).
```

B. Tableaux for first-order NC

Definition B.1. A *signed formula* is an expression of the form $v\colon A$, where $v \in NC$ and A is a formula.

Definition B.2. A *tableau* for a set of signed formulas Δ is a downward rooted tree of signed formulas where each one is either an element of Δ or results from a signed formula in the branch above it by a branch expansion rule. A tableau is *closed* if every branch contains, for some formula A, the signed formulas $v\colon A$ for all $v \in NC$.

The branch expansion rules for connective $\neg$ are given by

$$\frac{\mathfrak{ff}\colon \neg A}{\mathfrak{ff}\colon A} \qquad \frac{\mathfrak{fu}\colon \neg A}{\mathfrak{uf}\colon A} \qquad \frac{\mathfrak{ft}\colon \neg A}{\mathfrak{tf}\colon A} \qquad \frac{\mathfrak{uf}\colon \neg A}{\mathfrak{fu}\colon A} \qquad \frac{\mathfrak{uu}\colon \neg A}{\mathfrak{uu}\colon A} \qquad \frac{\mathfrak{ut}\colon \neg A}{\mathfrak{tu}\colon A}$$

$$\frac{\mathfrak{tf}\colon \neg A}{\mathfrak{ft}\colon A} \qquad \frac{\mathfrak{tu}\colon \neg A}{\mathfrak{ut}\colon A} \qquad \frac{\mathfrak{tt}\colon \neg A}{\mathfrak{tt}\colon A}$$

The branch expansion rules for connective $\land$ are given by

$$\begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{ff}\colon A \land B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{ff}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{tf}\colon B & \mathfrak{ff}\colon B & \mathfrak{tf}\colon A \end{array} \qquad \begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{fu}\colon A \land B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{ff}\colon A & \mathfrak{fu}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{fu}\colon B & \mathfrak{fu}\colon A & \mathfrak{tu}\colon A & \mathfrak{fu}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{ft}\colon A & \mathfrak{fu}\colon B & \mathfrak{ft}\colon A \\ \mathfrak{tf}\colon B & \mathfrak{ff}\colon B & \mathfrak{tu}\colon B & \mathfrak{tf}\colon A \\ \mathfrak{tu}\colon B & \mathfrak{fu}\colon B & & \mathfrak{tu}\colon A \\ \mathfrak{tt}\colon B & \mathfrak{ft}\colon B & & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{ft}\colon A \land B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{ff}\colon A & \mathfrak{ft}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{ft}\colon A & \mathfrak{tt}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{tf}\colon B & \mathfrak{ff}\colon B & \mathfrak{ft}\colon B & \mathfrak{tf}\colon A \\ \mathfrak{tt}\colon B & \mathfrak{ft}\colon B & \mathfrak{tt}\colon B & \mathfrak{tt}\colon A \end{array} \qquad \begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{uf}\colon A \land B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{uf}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{uf}\colon B & \mathfrak{uf}\colon B & \mathfrak{uf}\colon A \\ \mathfrak{tf}\colon B & & \mathfrak{tf}\colon A \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{uu}\colon A \land B} \\ \hline \mathfrak{uf}\colon A & \mathfrak{fu}\colon A \\ \mathfrak{uu}\colon A & \mathfrak{uu}\colon A \\ \mathfrak{ut}\colon A & \mathfrak{tu}\colon A \\ \mathfrak{uf}\colon B & \mathfrak{fu}\colon B \\ \mathfrak{uu}\colon B & \mathfrak{uu}\colon B \\ \mathfrak{ut}\colon B & \mathfrak{tu}\colon B \end{array} \qquad \begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{ut}\colon A \land B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{uf}\colon A & \mathfrak{ft}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{ut}\colon A & \mathfrak{ut}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{uf}\colon B & \mathfrak{uf}\colon B & \mathfrak{tt}\colon A & \mathfrak{uf}\colon A \\ \mathfrak{ut}\colon B & \mathfrak{ut}\colon B & \mathfrak{ft}\colon B & \mathfrak{ut}\colon A \\ \mathfrak{tf}\colon B & & \mathfrak{ut}\colon B & \mathfrak{tf}\colon A \\ \mathfrak{tt}\colon B & & \mathfrak{tt}\colon B & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{tf}\colon A \land B} \\ \hline \mathfrak{tf}\colon B & \mathfrak{tf}\colon A \end{array} \qquad \begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{tu}\colon A \land B} \\ \hline \mathfrak{tf}\colon B & \mathfrak{tu}\colon A & \mathfrak{tf}\colon A \\ \mathfrak{tu}\colon B & \mathfrak{tu}\colon B & \mathfrak{tu}\colon A \\ \mathfrak{tt}\colon B & & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{tt}\colon A \land B} \\ \hline \mathfrak{tf}\colon B & \mathfrak{tt}\colon A & \mathfrak{tf}\colon A \\ \mathfrak{tt}\colon B & \mathfrak{tt}\colon B & \mathfrak{tt}\colon A \end{array}$$

The branch expansion rules for connective $\lor$ are given by

$$\begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{ff}\colon A \lor B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{ff}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{ff}\colon B & \mathfrak{ft}\colon A \end{array} \qquad \begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{fu}\colon A \lor B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{fu}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{fu}\colon B & \mathfrak{fu}\colon B & \mathfrak{fu}\colon A \\ \mathfrak{ft}\colon B & & \mathfrak{ft}\colon A \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{ft}\colon A \vee B} \\ \hline \mathfrak{ft}\colon B & \mathfrak{ft}\colon A \end{array}$$

$$\begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{uf}\colon A \vee B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{uf}\colon A & \mathfrak{ff}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{ut}\colon A & \mathfrak{uf}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{uf}\colon B & \mathfrak{uf}\colon B & \mathfrak{tf}\colon A & \mathfrak{uf}\colon A \\ \mathfrak{ut}\colon B & \mathfrak{ut}\colon B & \mathfrak{ff}\colon B & \mathfrak{ut}\colon A \\ \mathfrak{tf}\colon B & & \mathfrak{uf}\colon B & \mathfrak{tf}\colon A \\ \mathfrak{tt}\colon B & & \mathfrak{tf}\colon B & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{uu}\colon A \vee B} \\ \hline \mathfrak{uf}\colon A & \mathfrak{fu}\colon A \\ \mathfrak{uu}\colon A & \mathfrak{uu}\colon A \\ \mathfrak{ut}\colon A & \mathfrak{tu}\colon A \\ \mathfrak{uf}\colon B & \mathfrak{fu}\colon B \\ \mathfrak{uu}\colon B & \mathfrak{uu}\colon B \\ \mathfrak{ut}\colon B & \mathfrak{tu}\colon B \end{array}$$

$$\begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{ut}\colon A \vee B} \\ \hline \mathfrak{ft}\colon B & \mathfrak{ut}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{ut}\colon B & \mathfrak{ut}\colon B & \mathfrak{ut}\colon A \\ \mathfrak{tt}\colon B & & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{tf}\colon A \vee B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{ff}\colon A & \mathfrak{tf}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{tf}\colon A & \mathfrak{tt}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{tf}\colon B & \mathfrak{ff}\colon B & \mathfrak{tf}\colon B & \mathfrak{tf}\colon A \\ \mathfrak{tt}\colon B & \mathfrak{tf}\colon B & \mathfrak{tt}\colon B & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{tu}\colon A \vee B} \\ \hline \mathfrak{ff}\colon B & \mathfrak{fu}\colon A & \mathfrak{tf}\colon A & \mathfrak{ff}\colon A \\ \mathfrak{fu}\colon B & \mathfrak{tu}\colon A & \mathfrak{tu}\colon A & \mathfrak{fu}\colon A \\ \mathfrak{ft}\colon B & \mathfrak{fu}\colon B & \mathfrak{tt}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{tf}\colon B & \mathfrak{tu}\colon B & \mathfrak{tf}\colon B & \mathfrak{tf}\colon A \\ \mathfrak{tu}\colon B & & \mathfrak{tu}\colon B & \mathfrak{tu}\colon A \\ \mathfrak{tt}\colon B & & \mathfrak{tt}\colon B & \mathfrak{tt}\colon A \end{array}$$

$$\begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{tt}\colon A \vee B} \\ \hline \mathfrak{ft}\colon B & \mathfrak{tt}\colon A & \mathfrak{ft}\colon A \\ \mathfrak{tt}\colon B & \mathfrak{tt}\colon B & \mathfrak{tt}\colon A \end{array}$$

The branch expansion rules rules for quantifier $\forall$ are given by

$$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{ff}\colon (\forall x)A(x)} \\ \hline \mathfrak{ff}\colon A(\tau) & \mathfrak{ff}\colon A(\alpha) \\ & \mathfrak{tf}\colon A(\alpha) \end{array}$$

$$\begin{array}{ccc} \multicolumn{3}{c}{\mathfrak{fu}\colon (\forall x)A(x)} \\ \hline \mathfrak{ff}\colon A(\tau_1) & \mathfrak{fu}\colon A(\tau_2) & \mathfrak{ff}\colon A(\alpha) \\ \mathfrak{ft}\colon A(\tau_1) & \mathfrak{tu}\colon A(\tau_2) & \mathfrak{ft}\colon A(\alpha) \\ \mathfrak{fu}\colon A(\tau_1) & & \mathfrak{fu}\colon A(\alpha) \\ & & \mathfrak{tf}\colon A(\alpha) \\ & & \mathfrak{tt}\colon A(\alpha) \\ & & \mathfrak{tu}\colon A(\alpha) \end{array}$$

$$\frac{\mathfrak{ft}\colon (\forall x)A(x)}{\begin{array}{lll} \mathfrak{ff}\colon A(\tau_1) & \mathfrak{ft}\colon A(\tau_2) & \mathfrak{ff}\colon A(\alpha) \\ \mathfrak{ft}\colon A(\tau_1) & \mathfrak{tt}\colon A(\tau_2) & \mathfrak{ft}\colon A(\alpha) \\ & & \mathfrak{tf}\colon A(\alpha) \\ & & \mathfrak{tt}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{uf}\colon (\forall x)A(x)}{\begin{array}{ll} \mathfrak{uf}\colon A(\tau) & \mathfrak{ff}\colon A(\alpha) \\ & \mathfrak{tf}\colon A(\alpha) \\ & \mathfrak{uf}\colon A(\alpha) \end{array}}$$

$$\frac{\mathfrak{ut}\colon (\forall x)A(x)}{\begin{array}{lll} \mathfrak{ft}\colon A(\tau_1) & \mathfrak{uf}\colon A(\tau_2) & \mathfrak{ff}\colon A(\alpha) \\ \mathfrak{tt}\colon A(\tau_1) & \mathfrak{ut}\colon A(\tau_2) & \mathfrak{ft}\colon A(\alpha) \\ \mathfrak{ut}\colon A(\tau_1) & & \mathfrak{tf}\colon A(\alpha) \\ & & \mathfrak{tt}\colon A(\alpha) \\ & & \mathfrak{uf}\colon A(\alpha) \\ & & \mathfrak{ut}\colon A(\alpha) \end{array}}$$

$$\frac{\mathfrak{uu}\colon (\forall x)A(x)}{\begin{array}{ll} \mathfrak{fu}\colon A(\tau_1) & \mathfrak{uf}\colon A(\tau_2) \\ \mathfrak{tu}\colon A(\tau_1) & \mathfrak{ut}\colon A(\tau_2) \\ \mathfrak{uu}\colon A(\tau_1) & \mathfrak{uu}\colon A(\tau_2) \end{array}}$$

$$\frac{\mathfrak{tf}\colon (\forall x)A(x)}{\mathfrak{tf}\colon A(\alpha)} \qquad \frac{\mathfrak{tu}\colon (\forall x)A(x)}{\begin{array}{ll} \mathfrak{tu}\colon A(\tau) & \mathfrak{tf}\colon A(\alpha) \\ & \mathfrak{tt}\colon A(\alpha) \\ & \mathfrak{tu}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{tt}\colon (\forall x)A(x)}{\begin{array}{ll} \mathfrak{tt}\colon A(\tau) & \mathfrak{tf}\colon A(\alpha) \\ & \mathfrak{tt}\colon A(\alpha) \end{array}}$$

The branch expansion rules rules for quantifier $\exists$ are given by

$$\frac{\mathfrak{ff}\colon (\exists x)A(x)}{\begin{array}{ll} \mathfrak{ff}\colon A(\tau) & \mathfrak{ff}\colon A(\alpha) \\ & \mathfrak{ft}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{fu}\colon (\exists x)A(x)}{\begin{array}{ll} \mathfrak{fu}\colon A(\tau) & \mathfrak{ff}\colon A(\alpha) \\ & \mathfrak{ft}\colon A(\alpha) \\ & \mathfrak{fu}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{ft}\colon (\exists x)A(x)}{\mathfrak{ft}\colon A(\alpha)}$$

$$\frac{\mathfrak{uf}\colon (\exists x)A(x)}{\begin{array}{lll} \mathfrak{ff}\colon A(\tau_1) & \mathfrak{uf}\colon A(\tau_2) & \mathfrak{ff}\colon A(\alpha) \\ \mathfrak{tf}\colon A(\tau_1) & \mathfrak{ut}\colon A(\tau_2) & \mathfrak{ft}\colon A(\alpha) \\ \mathfrak{uf}\colon A(\tau_1) & & \mathfrak{tf}\colon A(\alpha) \\ & & \mathfrak{tt}\colon A(\alpha) \\ & & \mathfrak{uf}\colon A(\alpha) \\ & & \mathfrak{ut}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{uu}\colon (\exists x)A(x)}{\begin{array}{ll} \mathfrak{fu}\colon A(\tau_1) & \mathfrak{uf}\colon A(\tau_2) \\ \mathfrak{tu}\colon A(\tau_1) & \mathfrak{ut}\colon A(\tau_2) \\ \mathfrak{uu}\colon A(\tau_1) & \mathfrak{uu}\colon A(\tau_2) \end{array}}$$

$$\frac{\mathfrak{ut}\colon (\exists x)A(x)}{\begin{array}{ll} \mathfrak{ut}\colon A(\tau) & \mathfrak{ft}\colon A(\alpha) \\ & \mathfrak{tt}\colon A(\alpha) \\ & \mathfrak{ut}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{tf}\colon (\exists x)A(x)}{\begin{array}{lll} \mathfrak{ff}\colon A(\tau_1) & \mathfrak{tf}\colon A(\tau_2) & \mathfrak{ff}\colon A(\alpha) \\ \mathfrak{tf}\colon A(\tau_1) & \mathfrak{tt}\colon A(\tau_2) & \mathfrak{ft}\colon A(\alpha) \\ & & \mathfrak{tf}\colon A(\alpha) \\ & & \mathfrak{tt}\colon A(\alpha) \end{array}}$$

$$\frac{\mathfrak{tu}\colon (\exists x)A(x)}{\begin{array}{ccc} \mathfrak{tf}\colon A(\tau_1) & \mathfrak{fu}\colon A(\tau_2) & \mathfrak{ff}\colon A(\alpha) \\ \mathfrak{tt}\colon A(\tau_1) & \mathfrak{tu}\colon A(\tau_2) & \mathfrak{ft}\colon A(\alpha) \\ \mathfrak{tu}\colon A(\tau_1) & & \mathfrak{fu}\colon A(\alpha) \\ & & \mathfrak{tf}\colon A(\alpha) \\ & & \mathfrak{tt}\colon A(\alpha) \\ & & \mathfrak{tu}\colon A(\alpha) \end{array}} \qquad \frac{\mathfrak{tt}\colon (\exists x)A(x)}{\begin{array}{cc} \mathfrak{tt}\colon A(\tau) & \mathfrak{ft}\colon A(\alpha) \\ & \mathfrak{tt}\colon A(\alpha) \end{array}}$$

Definition B.3. An interpretation $\mathfrak{I}$ *satisfies* a signed formula $v\colon A$ iff $\mathrm{val}_{\mathfrak{I}}(A) \neq v$. A set of signed formulas is *satisfiable* if some interpretation $\mathfrak{I}$ satisfies all signed formulas in it.

Theorem B.1. *A set of signed formulas is unsatisfiable iff it has a closed tableau.*

Proof. Apply Theorems 4.14 and 4.21 of [Hähnle, 1993]; interpreting $v\colon A$ as $\mathsf{S}\ A$, where $\mathsf{S} = NC \setminus \{v\}$. ⊣

Corollary B.2. *In* **NC**, $\Delta \vDash A$ *iff* $\{v\colon B \mid v \in NC \setminus NC^+, B \in \Delta\} \cup \{v\colon A \mid v \in NC^+\}$ *has a closed tableau.*