A

Taming Paraconsistent (and Other) Logics: An Algorithmic Approach

AGATA CIABATTONI, Vienna University of Technology
ORI LAHAV, Tel Aviv University
LARA SPENDIER, Vienna University of Technology
ANNA ZAMANSKY, University of Haifa

We develop a fully algorithmic approach to "taming" logics expressed Hilbert-style, i.e., reformulating them in terms of analytic sequent calculi, and useful semantics. Our approach applies to Hilbert calculi extending the positive fragment of propositional classical logic with axioms of a certain general form that contain new unary connectives. Our work encompasses various results already obtained for specific logics. It can be applied to new logics, as well as to known logics for which an analytic calculus or a useful semantics has so far not been available. A Prolog implementation of the method is described.

Categories and Subject Descriptors: I.2.3 [**Deduction and Theorem Proving**]: Deduction; F.4.1 [**Mathematical Logic**]: Computational Logic

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Non-classical logics, Proof theory, Sequent Calculus, Non-deterministic matrices, Paraconsistent Logics

ACM Reference Format:
Agata Ciabattoni, Ori Lahav, Lara Spendier and Anna Zamansky. 2013. Taming Paraconsistent (and Other) Logics: An Algorithmic Approach. *ACM Trans. Comput. Logic* V, N, Article A (January YYYY), 23 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Classical logic is often inadequate for dealing with the many challenges posed by emerging applications, which require handling inconsistent, incomplete or vague information, reasoning about dynamic structures, etc. Such applications call for various *non-classical logics*: fuzzy logics, paraconsistent logics, substructural logics and many more. These logics are often described using Hilbert systems, which provide few insights into such important issues as decidability, interpolation, etc. and are extremely cumbersome when it comes to automated deduction.

Figuratively speaking, to make a logic useful for practical purposes, it needs to be *"tamed"*,[1] i.e. reformulated in terms of an analytic calculus, where proof search proceeds by a step-wise decomposition of the formulas to be proved, and equipped with a useful and intuitive semantics. A desirable property of such semantics is *effective-*

[1]To tame: to reduce from a state of native wildness especially so as to be tractable and useful to humans. (Merriam-Webster Dictionary, Encyclopedia Britannica)

Work supported by the Israel Science Foundation (grant no. 280-10), the EC Marie Curie Fellowship (NDMANCL-252314), and FWF START Y544-N23.
Author's addresses: A. Ciabattoni and L. Spendier, Theory and Logic Group, Vienna University of Technology; O. Lahav, Tel Aviv University; A. Zamansky (Current address) Haifa University.

DOI:http://dx.doi.org/10.1145/0000000.0000000

ness in the sense that it naturally induces a decision procedure for the logic. Examples for effective semantics include finite-valued matrices, non-deterministic finite-valued matrices (Nmatrices, [Avron and Lev 2005; Avron and Zamansky 2011])), and partial Nmatrices (PNmatrices, [Baaz et al. 2013]). The latter two are natural generalizations of the former, which allow non-deterministic interpretations of logical connectives (in contrast to Nmatrices, PNmatrices also allow empty sets in the truth tables). Considering the wide variety of existing non-classical logics, and the new logics constantly introduced, it is also desirable to have an *algorithmic* approach to “taming”, that can be automated.

In this paper we provide such an approach for a large class of logics formulated in terms of Hilbert calculi. We introduce an algorithm for a systematic generation of sequent calculi for this class of logics, as well as simple and intuitive semantics for them. The introduced semantics, based on PNmatrices, is effective and also provides a simple sufficient condition for the analyticity of the corresponding calculi.

This class of logics contains many useful and well-known logics, as well as infinitely many new ones: it consists of extensions of the positive fragment of classical propositional logic CL^+ with axioms of a certain natural form. More precisely, they are induced by a family $\mathbf{H}$ of Hilbert calculi obtained by (i) extending the language of CL^+ with finitely many unary connectives, and (ii) adding to HCL^+ (a Hilbert axiomatization of CL^+) axioms over the extended language of a certain general form.

The simplest and best known member of $\mathbf{H}$ is the standard calculus for classical logic, obtained by adding to HCL^+ the usual axioms for negation. Further examples include many C-systems [Da Costa 1974; Carnielli and Marcos 2002] and the paraconsistent logics investigated in [Kamide 2013]. While analytic sequent calculi and/or adequate semantics for some of these logics were already available, the novelty of our approach is that their introduction is *fully automated* (as opposed, e.g., to [Avron et al. 2012; 2013], where the construction of semantics is done manually). Moreover, it applies to infinitely many logics (and is not tailored to the C-systems as [Avron et al. 2012; 2013] or to specific logics as [Kamide 2013]), many of which have had so far no adequate calculi or available semantics.

Given a system $H \in \mathbf{H}$, our “taming” procedure works as follows:

Step 1. transform the Hilbert axioms in H into sequent calculus rules
Step 2. extract a PNmatrix out of the obtained sequent calculus

Step 1 is done by adapting the procedure in [Ciabattoni et al. 2008], where certain Hilbert axioms are transformed into equivalent (sequent and hypersequent) *structural rules* added to a suitable base system. In contrast to [Ciabattoni et al. 2008], the rules extracted from the axioms of $H \in \mathbf{H}$ are *logical rules* in Gentzen's terminology, i.e., they introduce logical connectives. In particular, each of these rules may involve more than one connective, and hence the analyticity of the calculus resulting from addition of such rules depends on the way they interact with all the existing rules mentioning the same connectives. This requires a view on the calculus as a whole, which is provided by the semantics constructed in Step 2 given in the framework of PNmatrices [Baaz et al. 2013]. This framework is non-deterministic and by employing empty sets of options in the truth-tables, makes it possible to “forbid” some combinations of truth values. This type of semantics is still effective, as it guarantees the decidability of the corresponding sequent calculus. As a corollary it follows that each system $H \in \mathbf{H}$ is decidable. Furthermore, we show that if the PNmatrix constructed for H is contains no empty sets, then the corresponding sequent calculus is analytic, as it enjoys a certain generalized subformula property.

This paper extends the results of [Ciabattoni et al. 2013]. However, in contrast to [Ciabattoni et al. 2013], we allow here axioms with possible nesting of unary connectives of any fixed depth. This allows us to capture, e.g., the logics investigated in [Kamide 2013] that could not be dealt with in our the previous work. While Step 1 of our procedure is easily extended to the new axioms, Step 2 required major changes in the construction of our PNmatrices, in the notion of analyticity and in all related theorems.

Our procedure is implemented in the Prolog system *Paralyzer* — PARAconsistent (and other) logics anaLYZER, drawing inspiration from the system MUltlog [Baaz et al. 1996], which introduces analytic calculi for finite-valued logics. For any set of axioms of a certain general[2] form, the system returns as output (a paper containing) the corresponding sequent calculus and the associated PNmatrix.

Our "taming" procedure is a concrete step towards a systematization of the vast variety of existing non-classical logics and the development of tools for designing new application-oriented logics in the spirit of [Ohlbach 1994].

This paper is organized as follows: Section 2 describes the translation of any Hilbert calculus $H \in \mathbf{H}$ into an equivalent sequent calculus G. In Section 3 we obtain a corresponding PNmatrix for each such sequent calculus G and use it to prove the decidability of H. Section 4 provides a way for checking the analyticity of G. Section 5 presents the implementation of *Paralyzer*, while Section 6 contains a summary of the paper and discusses some directions for future work.

2. FROM HILBERT SYSTEMS TO SEQUENT CALCULI (STEP 1)

In what follows, $\mathcal{L}^+_{cl}$ denotes the language of CL^+, the positive fragment of propositional classical logic, consisting of the binary connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\supset$ (implication), and the atomic formulas $p_1, p_2, \ldots$. We consider propositional languages that extend $\mathcal{L}^+_{cl}$ with finitely many unary connectives. The symbol $\mathcal{L}$ is used to denote such a language. We identify a language with its set of formulas, e.g. when writing $\varphi \in \mathcal{L}$. Several additional notations are used: $\mathcal{U}_\mathcal{L}$ for the set of unary connectives employed in $\mathcal{L}$; $\star, \triangleright$ for arbitrary elements of $\mathcal{U}_\mathcal{L}$; $\mathcal{U}^*_\mathcal{L}$ for the set of all finite sequences of connectives from $\mathcal{U}_\mathcal{L}$, with the empty sequence denoted by ϵ; and $\bar{\star}, \bar{\triangleright}$ for arbitrary such finite sequences. We also employ standard notations for their concatenation (e.g., when writing expressions like $\bar{\star}\bar{\triangleright}$).

2.1. The Family $\mathbf{H}$

Consider the Hilbert axiomatization HCL^+ of CL^+, which in addition to the MP rule includes the following axioms:

I1. $\varphi \supset (\psi \supset \varphi)$
I2. $(\varphi \supset \psi \supset \theta) \supset (\varphi \supset \psi) \supset (\varphi \supset \theta)$
I3. $((\psi \supset \varphi) \supset \psi) \supset \psi$
C1. $\varphi \wedge \psi \supset \varphi$
C2. $\varphi \wedge \psi \supset \psi$
C3. $\varphi \supset (\psi \supset \varphi \wedge \psi)$
D1. $\varphi \supset \varphi \vee \psi$
D2. $\psi \supset \varphi \vee \psi$
D3. $(\varphi \supset \theta) \supset (\psi \supset \theta) \supset (\varphi \vee \psi \supset \theta)$

$\mathbf{H}$ consists of all Hilbert calculi that are obtained by augmenting HCL^+ (or any other axiomatization of CL^+) with axioms in $\mathbf{Ax}_\mathcal{L}$ of the following general form:

Definition 2.1. $\mathbf{Ax}_\mathcal{L}$ is the set of $\mathcal{L}$-formulas that:

[2] *Paralyzer* currently handles the axiom systems of [Ciabattoni et al. 2013].

$(\mathbf{n_1})$	$p_1 \vee \neg p_1$	$(\mathbf{n_2})$	$p_1 \supset (\neg p_1 \supset p_2)$
$(\mathbf{b})$	$p_1 \supset (\neg p_1 \supset (\circ p_1 \supset p_2))$	$(\mathbf{r}_\diamond)$	$\circ(p_1 \diamond p_2) \supset (\circ p_1 \vee \circ p_2)$
$(\mathbf{k})$	$\circ p_1 \vee (p_1 \wedge \neg p_1)$	$(\mathbf{i})$	$\neg\circ p_1 \supset (p_1 \wedge \neg p_1)$
$(\mathbf{o^1_\diamond})$	$\circ p_1 \supset \circ(p_1 \diamond p_2)$	$(\mathbf{o^2_\diamond})$	$\circ p_2 \supset \circ(p_1 \diamond p_2)$
$(\mathbf{a}_\diamond)$	$(\circ p_1 \wedge \circ p_2) \supset \circ(p_1 \diamond p_2)$	$(\mathbf{a}_\neg)$	$\circ p_1 \supset \circ\neg p_1$
$(\mathbf{c_k})$	$\neg\neg\neg^{k-1} p_1 \supset \neg^{k-1} p_1$	$(\mathbf{e_k})$	$\neg^{k-1} p_1 \supset \neg\neg\neg^{k-1} p_1$
$(\mathbf{e^l_{\diamond_k}})$	$\neg^k(p_1 \diamond p_2) \supset (\neg^k p_1 \diamond \neg^k p_2)$	$(\mathbf{e^r_{\diamond_k}})$	$(\neg^k p_1 \diamond \neg^k p_2) \supset \neg^k(p_1 \diamond p_2)$
$(\mathbf{o^l_{\supset_k}})$	$\neg^k(p_1 \supset p_2) \supset (\neg^{k-1} p_1 \wedge \neg^k p_2)$	$(\mathbf{o^r_{\supset_k}})$	$(\neg^{k-1} p_1 \wedge \neg^k p_2) \supset \neg^k(p_1 \supset p_2)$
$(\mathbf{o^l_{\wedge_k}})$	$\neg^k(p_1 \wedge p_2) \supset (\neg^k p_1 \vee \neg^k p_2)$	$(\mathbf{o^r_{\wedge_k}})$	$(\neg^k p_1 \vee \neg^k p_2) \supset \neg^k(p_1 \wedge p_2)$
$(\mathbf{o^l_{\vee_k}})$	$\neg^k(p_1 \vee p_2) \supset (\neg^k p_1 \wedge \neg^k p_2)$	$(\mathbf{o^r_{\vee_k}})$	$(\neg^k p_1 \wedge \neg^k p_2) \supset \neg^k(p_1 \vee p_2)$

Fig. 1. Examples of formulas from $\mathbf{Ax}_\mathcal{L}$ for $\mathcal{U}_\mathcal{L} = \{\neg, \circ\}$ ($\diamond \in \{\vee, \wedge, \supset\}$, and any $k > 0$)

(1) are generated by the following grammar (S is the initial variable):

$S = R_1 \mid R_2$ for $\diamond \in \{\wedge, \vee, \supset\}, \bar\star \in \mathcal{U}_\mathcal{L}^* \setminus \{\epsilon\}$

$R_1 = (R_1 \diamond P_1) \mid (P_1 \diamond R_1) \mid \bar\star p_1$ $\quad P_1 = (P_1 \diamond P_1) \mid \bar\star p_1 \mid p_1 \mid p_2$

$R_2 = (R_2 \diamond P_2) \mid (P_2 \diamond R_2) \mid \bar\star(p_1 \diamond p_2)$ $\quad P_2 = (P_2 \diamond P_2) \mid \bar\star p_1 \mid p_1 \mid \bar\star p_2 \mid p_2$

(2) satisfy the following conditions: for some subformula $\varphi = \bar\star p_1$ of an $\mathcal{L}$-formula arising from the start variable R_1 (and for the subformula $\varphi = \bar\star(p_1 \diamond p_2)$ of an $\mathcal{L}$-formula arising from R_2, resp.): φ must not be contained
(i) in a positively[3] occurring (sub)formula of the form $\psi_1 \wedge \psi_2$, and
(ii) in a negatively occurring (sub)formula of the form $\psi_1 \vee \psi_2$ or $\psi_1 \supset \psi_2$.

The formulas in $\mathbf{Ax}_\mathcal{L}$ are axiom schemata in which p_1, p_2 denote metavariables which are substituted by any $\mathcal{L}$-formula in the instances of the schema. Roughly speaking, the axiom schemata in $\mathbf{Ax}_\mathcal{L}$ contain

(R_1) at least one propositional variable p_1 prefixed with a non-empty sequence of connectives from $\mathcal{U}_\mathcal{L}$ and possibly the propositional variables p_1, p_2, or
(R_2) exactly one formula $(p_1 \diamond p_2)$ prefixed with a sequence of connectives from $\mathcal{U}_\mathcal{L}$ and possibly the propositional variables p_1, p_2, possibly prefixed with sequences of connectives from $\mathcal{U}_\mathcal{L}$.

Example 2.2. The axioms $(\mathbf{n_1})$, $(\mathbf{n_2})$, $(\mathbf{b})$, $(\mathbf{a}_\neg)$ (cf. Figure 1) are generated by (R_1) in the grammar of Definition 2.1, whereas the axioms $(\mathbf{i})$, $(\mathbf{o^1_\wedge})$, $(\mathbf{a}_\vee)$ (cf. Figure 1) are generated by (R_2). Axioms that are *not* in $\mathbf{Ax}_\mathcal{L}$ are $\neg(\neg p_1 \wedge p_1) \supset \circ p_1$ (it cannot be generated by (R_1) or (R_2)) and $p_1 \wedge \neg p_1$ (it does not satisfy the conditions in Definition 2.1.(2)).

Definition 2.3. $\mathbf{H}$ is the family of Hilbert calculi obtained by extending HCL^+ with any finite set of axioms from $\mathbf{Ax}_\mathcal{L}$ for some language $\mathcal{L}$.

Remark 2.4. The family of Hilbert calculi handled in [Ciabattoni et al. 2013] is properly contained in $\mathbf{H}$. The subformulas in the axiom schemata of $\mathbf{Ax}_\mathcal{L}$ can indeed be prefixed by any finite sequence of connectives from $\mathcal{U}_\mathcal{L}$, and not only by *one* (or, in particular cases, by *two*) as in the axioms considered in [Ciabattoni et al. 2013].

$\mathbf{H}$ includes many known Hilbert calculi as shown in the examples below.

Example 2.5. A standard calculus for (propositional) classical logic is obtained by adding the axioms $(\mathbf{n_1})$ and $(\mathbf{n_2})$ from Figure 1 to HCL^+.

[3]Recall that a subformula φ occurs *negatively* (*positively*, resp.) in an $\mathcal{L}$-formula ψ if there is an odd (even, resp.) number of implications $\supset$ in ψ having φ as a subformula of its antecedent, see e.g., [Buss 1998].

Example 2.6. Paraconsistent logics are logics which are not trivialized in the presence of inconsistency, i.e. for which there are some formulas ψ, φ, such that: $\psi, \neg\psi \nvdash \varphi$. The family of C-systems [Da Costa 1974; Carnielli and Marcos 2002; Carnielli et al. 2007; Avron 2007; Avron et al. 2012; 2013] is a well-known family of paraconsistent logics in which the notion of consistency is internalized into the object language by employing a unary consistency operator $\circ$, the intuitive meaning of $\circ\psi$ being "ψ is consistent". C-systems are defined by adding to HCL^+ the axioms $(\mathbf{b})$ and $(\mathbf{n_1})$, as well as different subsets of the axioms $(\mathbf{r}_\diamond), (\mathbf{k}), (\mathbf{i}), (\mathbf{o}^1_\diamond), (\mathbf{o}^2_\diamond), (\mathbf{a}_\neg), (\mathbf{e_k}), (\mathbf{a}_\diamond)$ and $(\mathbf{c_k})$ in Figure 1.

Example 2.7. For each $n \geq 0$ the logics $L(2^{n+2})$ are obtained by adding to HCL^+ the following axioms from Figure 1: $(\mathbf{c_n}), (\mathbf{e_n})$, and for every $0 < m \leq n+1$:

— $(\mathbf{e}^{\mathbf{l}}_{\diamond_\mathbf{m}})$, $(\mathbf{e}^{\mathbf{r}}_{\diamond_\mathbf{m}})$, if m is even
— $(\mathbf{o}^{\mathbf{l}}_{\wedge_\mathbf{m}})$, $(\mathbf{o}^{\mathbf{r}}_{\wedge_\mathbf{m}})$, $(\mathbf{o}^{\mathbf{l}}_{\vee_\mathbf{m}})$, $(\mathbf{o}^{\mathbf{r}}_{\vee_\mathbf{m}})$, $(\mathbf{o}^{\mathbf{l}}_{\supset_\mathbf{m}})$, $(\mathbf{o}^{\mathbf{r}}_{\supset_\mathbf{m}})$, if m is odd.

[Kamide 2013] introduces analytic sequent calculi and suitable semantics with ad-hoc proofs of soundness, completeness and analyticity for $L(2^{n+2})$.

Definition 2.8. For an $\mathcal{L}$-formula ψ, let Θ_ψ denote the set of all prefixes (including the empty one ϵ) of the maximal sequences of connectives from $\mathcal{U}_\mathcal{L}$ that occur in ψ. Formally, Θ_ψ is defined inductively as follows:

— If $\psi = \bar{\star}p$ for some $\bar{\star} \in \mathcal{U}^*_\mathcal{L}$ and atomic formula p, then Θ_ψ consists of all prefixes of $\bar{\star}$.
— If $\psi = \bar{\star}(\psi_1 \diamond \psi_2)$ for some $\bar{\star} \in \mathcal{U}^*_\mathcal{L}$, $\diamond \in \{\vee, \wedge, \supset\}$ and $\mathcal{L}$-formulas ψ_1, ψ_2, then Θ_ψ consists of all prefixes of $\bar{\star}$, together with the sequences in $\Theta_{\psi_1} \cup \Theta_{\psi_2}$.

Example 2.9. For $\mathcal{U}_\mathcal{L} = \{\circ, *, \neg\}$ and $\psi = * \circ \neg p_1 \supset p_1$, we have $\Theta_\psi = \{\epsilon, *, *\circ, * \circ \neg\}$.

NOTATION 1. *For* $H \in \mathbf{H}$, $\Theta_H = \bigcup_{\psi \in H} \Theta_\psi$.

We note that Θ_H will be used in the second step of our procedure for introducing the PNmatrix. In particular, Θ_H will determine the number and the form of the truth values in the PNmatrix for H.

2.2. From axioms to logical sequent rules

By suitably adapting the procedure in [Ciabattoni et al. 2008] we transform each $H \in \mathbf{H}$ into an equivalent sequent calculus.

To simplify our presentation, we use a label-based formulation of sequent calculi, and recall below the relevant definitions and notations.

Definition 2.10. Let $\mathcal{L}$ be a propositional language.

(1) A *labelled $\mathcal{L}$-formula* has the form $b : \psi$, where $b \in \{f, t\}$ and ψ is an $\mathcal{L}$-formula.
(2) An *$\mathcal{L}$-sequent* is a finite set of labelled $\mathcal{L}$-formulas. The usual sequent notation $\psi_1, \ldots, \psi_n \Rightarrow \varphi_1, \ldots, \varphi_m$ corresponds to the set $\{f : \psi_1, \ldots, f : \psi_n, t : \varphi_1, \ldots, t : \varphi_m\}$ of labelled formulas.
(3) An *$\mathcal{L}$-substitution* is a function $\sigma : \mathcal{L} \to \mathcal{L}$, that satisfies the following conditions:
(a) $\sigma(\star\psi) = \star(\sigma(\psi))$ for every unary connective $\star$ of $\mathcal{L}$ and formula ψ.
(b) $\sigma(\psi_1 \diamond \psi_2) = \sigma(\psi_1) \diamond \sigma(\psi_2)$ for every $\diamond \in \{\vee, \wedge, \supset\}$ and formulas ψ_1, ψ_2.
$\mathcal{L}$-substitutions are naturally extended to labelled $\mathcal{L}$-formulas, $\mathcal{L}$-sequents, and sets of $\mathcal{L}$-sequents.
(4) An *$\mathcal{L}$-rule* is an expression of the form Q/s, where Q is a finite set of $\mathcal{L}$-sequents (called *premises*) and s is an $\mathcal{L}$-sequent (called *conclusion*). An *application* of an $\mathcal{L}$-rule Q/s is any inference step inferring the $\mathcal{L}$-sequent $\sigma(s) \cup c$ from the set of $\mathcal{L}$-sequents $\{\sigma(q) \cup c \mid q \in Q\}$, where σ is an $\mathcal{L}$-substitution and c is an $\mathcal{L}$-sequent.

(5) A sequent calculus G for $\mathcal{L}$ consists of a finite set of $\mathcal{L}$-rules. We write $\mathcal{S} \vdash_G s$ whenever the $\mathcal{L}$-sequent s is derivable from the set $\mathcal{S}$ of $\mathcal{L}$-sequents in G.

Note that since we defined sequents as *sets*, the structural rules of contraction and exchange are implicitly included in all sequent calculi studied in this paper.

Example 2.11. Examples of rules and their applications (in standard sequent notation) are:

$(\neg\neg \Rightarrow)$ $\{\{f : p_1\}\}/\{f : \neg\neg p_1\}$ $\dfrac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma, \neg\neg\varphi \Rightarrow \Delta}$

$(\Rightarrow \neg\neg\neg)$ $\{\{t : \neg p_1\}\}/\{t : \neg\neg\neg p_1\}$ $\dfrac{\Gamma \Rightarrow \neg\varphi, \Delta}{\Gamma \Rightarrow \neg\neg\neg\varphi, \Delta}$

$(\Rightarrow \neg\neg\wedge)$ $\{\{t : \neg\neg p_1\}, \{t : \neg\neg p_2\}\}/\{t : \neg\neg(p_1 \wedge p_2)\}$ $\dfrac{\Gamma \Rightarrow \neg\neg\varphi, \Delta \quad \Gamma \Rightarrow \neg\neg\psi, \Delta}{\Gamma \Rightarrow \neg\neg(\varphi \wedge \psi), \Delta}$

Example 2.12. Formulated according to Definition 2.10, the standard sequent calculus LK^+ for CL^+ consists of the following set of $\mathcal{L}_{cl}^+$-rules:

(id)	$\emptyset/\{f : p_1, t : p_1\}$	(cut)	$\{\{f : p_1\}, \{t : p_1\}\}/\emptyset$
$(W\Rightarrow)$	$\{\emptyset\}/\{f : p_1\}$	$(\Rightarrow W)$	$\{\emptyset\}/\{t : p_1\}$
$(\wedge \Rightarrow)$	$\{\{f : p_1, f : p_2\}\}/\{f : p_1 \wedge p_2\}$	$(\Rightarrow \wedge)$	$\{\{t : p_1\}, \{t : p_2\}\}/\{t : p_1 \wedge p_2\}$
$(\vee \Rightarrow)$	$\{\{f : p_1\}, \{f : p_2\}\}/\{f : p_1 \vee p_2\}$	$(\Rightarrow \vee)$	$\{\{t : p_1, t : p_2\}\}/\{t : p_1 \vee p_2\}$
$(\supset\Rightarrow)$	$\{\{t : p_1\}, \{f : p_2\}\}/\{f : p_1 \supset p_2\}$	$(\Rightarrow\supset)$	$\{\{f : p_1, t : p_2\}\}/\{t : p_1 \supset p_2\}$

$(\diamond \Rightarrow)$ and $(\Rightarrow \diamond)$ with $\diamond \in \{\wedge, \vee, \supset\}$ are the *logical rules* of LK^+.

In what follows, given a Hilbert system $H \in \mathbf{H}$ we show how to construct a sequent calculus G_H equivalent to H in the following sense:

Definition 2.13. A sequent calculus G is *equivalent* to a Hilbert system H if for every finite set $\Gamma \cup \{\varphi\}$ of formulas: φ is provable in H from Γ (in symbols $\Gamma \vdash_H \varphi$) iff $\Gamma \Rightarrow \varphi$ is provable in G (in symbols $\vdash_G \Gamma \Rightarrow \varphi$).

FACT 1. *LK^+ (see Example 2.12) is equivalent to HCL^+.*

NOTATION 2. *We denote by $H \cup \{\varphi\}$ ($H \setminus \{\varphi\}$ resp.) the Hilbert system obtained from H by adding (removing) the axiom φ, and by $G \cup R$ the sequent calculus extending G with the set R of $\mathcal{L}$-rules.*

Definition 2.14. Let R and R' be two finite sets of $\mathcal{L}$-rules, and G be a sequent calculus for $\mathcal{L}$. R and R' are *equivalent in* G if $Q \vdash_{G\cup R'} s$ for every $Q/s \in R$, and $Q \vdash_{G\cup R} s$ for every $Q/s \in R'$.

Clearly, the above definition of equivalence between rules could be reformulated by considering rule applications.

PROPOSITION 2.15. *Let R and R' be two finite sets of $\mathcal{L}$-rules, and G be a sequent calculus for $\mathcal{L}$. R and R' are equivalent iff the following hold for every $\mathcal{L}$-sequent c and $\mathcal{L}$-substitution σ: $\sigma(Q) \cup c \vdash_{G\cup R'} \sigma(s) \cup c$ for every $Q/s \in R$, and $\sigma(Q) \cup c \vdash_{G\cup R} \sigma(s) \cup c$ for every $Q/s \in R'$.*

PROOF. One direction directly follows from the following properties of every sequent calculus G for $\mathcal{L}$:

— If $\mathcal{S} \vdash_G s$, then $\sigma(\mathcal{S}) \vdash_G \sigma(s)$ for every set $\mathcal{S}$ of $\mathcal{L}$-sequents, $\mathcal{L}$-sequent s, and $\mathcal{L}$-substitution σ.
— If $\mathcal{S} \vdash_G s$, then $\{s' \cup c \mid s' \in \mathcal{S}\} \vdash_G s \cup c$ for every set $\mathcal{S}$ of $\mathcal{L}$-sequents, and $\mathcal{L}$-sequents s and c.

Both properties are proved by a straightforward induction on the length of derivation in G. The converse follows by taking σ to be identity and c to be the empty context sequent. □

Definition 2.16. An $\mathcal{L}_{cl}^{+}$-rule Q/s is *invertible in* LK^+ if $s \vdash_{LK^+} q$ for every $q \in Q$.

FACT 2. *The logical rules of* LK^+ *are invertible in* LK^+.

To transform each $H \in \mathbf{H}$ into an equivalent sequent calculus the idea is to extract suitable logical rules out of the axioms of H belonging to $\mathbf{Ax}_{\mathcal{L}}$ and add the obtained rules to LK^+. The procedure – which is contained in the proof of Theorem 2.22 – is inspired by the method in [Ciabattoni et al. 2008] and is roughly described below. Given any axiom $\varphi \in \mathbf{Ax}_{\mathcal{L}}$, it works as follows:

(Step i) We start from the rule $\emptyset/\{t : \varphi\}$. By utilizing the invertibility of the logical rules of LK^+ as much as possible, we obtain an equivalent set R of rules each having the form $\emptyset/\{b_1 : \varphi_1, \ldots, b_n : \varphi_n\}$ with $b_i \in \{t, f\}$. Note that due to the shape of φ (see Definition 2.1) it must be the case that each φ_i is either of the form $\bar{\star}p_1$ or $\bar{\star}p_2$ with $\bar{\star} \in \Theta_H$, and there is at most one φ_i of the form $\bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$.
(Step ii) Next, we remove each rule $r \in R$ whose conclusion contains $\{t : p_i, f : p_i\}$ for $i \in \{1, 2\}$. Moreover, for each remaining rule if the conclusion does not contain $\bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$, we remove all variables p_2 and use Lemma 2.21 below to ensure that the resulting rule is equivalent to r.
(Step iii) Choose a labelled formula β of the form $\bar{\star}p_1$ or $\bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$, and move the remaining formulas to the premises of the rule while switching their corresponding sequent sides (see Lemma 2.20 below). β is the formula introduced by the rule and has .

We illustrate the steps to obtain sequent rules out of axioms first with an example.

Example 2.17. Let φ be the axiom $(\mathbf{n_2})$ $p_1 \supset (\neg p_1 \supset p_2)$ from Figure 1. The algorithm works as follows:

$$
\begin{array}{lll}
 & \emptyset/\{t : p_1 \supset (\neg p_1 \supset p_2)\} & \\
\text{(i)} & \longrightarrow^{\text{Invertibility of } (\Rightarrow\supset)} & \emptyset/\{f : p_1, t : \neg p_1 \supset p_2)\} \\
\text{(i)} & \longrightarrow^{\text{Invertibility of } (\Rightarrow\supset)} & \emptyset/\{f : p_1, f : \neg p_1, t : p_2\} \\
\text{(ii)} & \longrightarrow^{\text{Lemma 2.21}} & \emptyset/\{f : p_1, f : \neg p_1\} \\
\text{(iii)} & \longrightarrow^{\text{Lemma 2.20}} & \{\{t : p_1\}\}/\{f : \neg p_1\}
\end{array}
$$

Due to the special format of φ ($\in \mathbf{Ax}_{\mathcal{L}}$) the rules generated by the algorithm have the general form depicted in Figure 2. The Θ-unary rules arise from axioms generated starting from R_1 in the grammar of Definition 2.1, while the Θ-binary rules are generated starting from R_2.

Remark 2.18. Distinguishing between the two types of rules in Figure 2 will be crucial for the semantic definitions in Section 3. As we shall see, rules of different types will play different semantic roles: the Θ-unary rules will determine the truth values for the PNmatrix, while the Θ-binary rules will refine the truth tables for the binary connectives of the PNmatrix.

Definition 2.19. Let Θ be a non-empty subset of $\mathcal{U}_{\mathcal{L}}^{*}$ that is closed under prefixes (in particular, $\epsilon \in \Theta$). An $\mathcal{L}$-rule Q/s is called Θ-*simple* if it is either Θ-unary or Θ-binary

	Rule	Application form
Θ-unary	$\mathcal{P}/\{t : \bar{\star}p_1\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi \Rightarrow \Delta \quad \Gamma \Rightarrow \bar{\bullet}_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi, \Delta}{\Gamma \Rightarrow \bar{\star}\varphi, \Delta}$
	$\mathcal{P}/\{f : \bar{\star}p_1\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi \Rightarrow \Delta \quad \Gamma \Rightarrow \bar{\bullet}_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi, \Delta}{\Gamma, \bar{\star}\varphi \Rightarrow \Delta}$
	where $\mathcal{P} = \{\{f : \bar{\triangleright}_1 p_1\}, \dots, \{f : \bar{\triangleright}_n p_1\}, \{t : \bar{\bullet}_1 p_1\}, \dots, \{t : \bar{\bullet}_m p_1\}\}$	
Θ-binary	$\mathcal{Q}/\{t : \bar{\star}(p_1 \diamond p_2)\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi_{i_1} \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi_{i_n} \Rightarrow \Delta \quad \Gamma \Rightarrow \bar{\bullet}_1\varphi_{j_1}, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi_{j_m}, \Delta}{\Gamma \Rightarrow \bar{\star}(\varphi_1 \diamond \varphi_2), \Delta}$
	$\mathcal{Q}/\{f : \bar{\star}(p_1 \diamond p_2)\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi_{i_1} \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi_{i_n} \Rightarrow \Delta \quad \Gamma \Rightarrow \bar{\bullet}_1\varphi_{j_1}, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi_{j_m}, \Delta}{\Gamma, \bar{\star}(\varphi_1 \diamond \varphi_2) \Rightarrow \Delta}$
	where $\mathcal{Q} = \{\{f : \bar{\triangleright}_1 p_{i_1}\}, \dots, \{f : \bar{\triangleright}_n p_{i_n}\}, \{t : \bar{\bullet}_1 p_{j_1}\}, \dots, \{t : \bar{\bullet}_m p_{j_m}\}\}$	

Fig. 2. The general form of our rules ($\bar{\star} \in \Theta \setminus \{\epsilon\}, \bar{\triangleright}_i, \bar{\bullet}_j \in \Theta, \diamond \in \{\wedge, \vee, \supset\}, i_1, \dots, i_n, j_1, \dots, j_m \in \{1, 2\}$)

(cf. Figure 2). A sequent calculus for $\mathcal{L}$ is called Θ-*simple* if it is obtained by augmenting LK^+ with a finite set of Θ-simple $\mathcal{L}$-rules. We shall omit Θ when it is clear from the context.

The lemma below, known as Ackermann's lemma and used, e.g., in [Ciabattoni et al. 2008] for substructural logics and in [Conradie and Palmigiano 2012] for modal logics, allows us to "move" formulas from the conclusions of sequent rules to their premises (while making the formulas switch the sequent hand-side).

LEMMA 2.20. *Let G be a sequent calculus for $\mathcal{L}$ extending LK^+,*

$$r = \emptyset/\{b_1 : \varphi_1, \dots, b_n : \varphi_n\} \qquad \textit{and} \qquad r' = \{\{\underline{b_2} : \varphi_2\}, \dots, \{\underline{b_n} : \varphi_n\}\}/\{b_1 : \varphi_1\}$$

be $\mathcal{L}$-rules where $\underline{f} = t$ and $\underline{t} = f$. Then $\{r\}$ and $\{r'\}$ are equivalent in G.

PROOF. In order to show $\{\{\underline{b_2} : \varphi_2\}, \dots, \{\underline{b_n} : \varphi_n\}\} \vdash_{G \cup \{r\}} \{b_1 : \varphi_1\}$ we use an application of r, $n-1$ cuts and weakenings. For the other direction, we use (id) to obtain $\{f : \varphi_i, t : \varphi_i\}$ for every $2 \leq i \leq n$ and then apply weakenings and r'. □

When using the logical rules of LK^+ to decompose axioms generated via (R_1) in the grammar of Definition 2.1, p_2 might appear only as $b : p_2$ with $b \in \{t, f\}$ (see, e.g., Example 2.17). The next lemma ensures that we can safely remove $b : p_2$ in these cases.

LEMMA 2.21. *Let G be a sequent calculus for $\mathcal{L}$ extending LK^+. Let s be an $\mathcal{L}$-sequent, and let $s' = s \cup \{b : p\}$, where $b \in \{f, t\}$ and p is an atomic formula that does not occur in s. Then, $\vdash_{G \cup \{\emptyset/s'\}} \Gamma \Rightarrow \varphi$ iff $\vdash_{G \cup \{\emptyset/s\}} \Gamma \Rightarrow \varphi$, for every $\mathcal{L}$-sequent $\Gamma \Rightarrow \varphi$.*

PROOF. Suppose that $\vdash_{G \cup \{\emptyset/s'\}} \Gamma \Rightarrow \varphi$. Since applications of $\emptyset/s'$ can be simulated using weakenings and $\emptyset/s$, $\vdash_{G \cup \{\emptyset/s\}} \Gamma \Rightarrow \varphi$ clearly holds. For the converse direction, let P be a derivation of $\Gamma \Rightarrow \varphi$ in $G \cup \{\emptyset/s\}$; we distinguish two cases:

— $b = f$. Then every application of $\emptyset/s$ in P deriving $\sigma(s)$ can be simulated in $G \cup \{\emptyset/s'\}$ by using (cut) on $\sigma(s) \cup \{f : p_1 \supset p_1\}$ (obtained by $\emptyset/s'$ in which p is substituted with $p_1 \supset p_1$) and $\sigma(s) \cup \{t : p_1 \supset p_1\}$ (which is derivable in LK^+).

— $b = t$. Then every application of $\emptyset/s$ in P is replaced with an application of $\emptyset/s'$, in which p is substituted with φ. $t : \varphi$ is then propagated until the end sequent. □

THEOREM 2.22. *Let $H \in \mathbf{H}$ be a Hilbert calculus for $\mathcal{L}$. There is an algorithm for constructing an equivalent Θ_H-simple sequent calculus G_H for $\mathcal{L}$.*

PROOF. Let $H \in \mathbf{H}$, $\psi \in \mathbf{Ax}_{\mathcal{L}} \cap H$ and G_H^- be the sequent calculus equivalent to $H \backslash \{\psi\}$ in the sense of Definition 2.13. We construct a sequent calculus G_H equivalent to H by extending G_H^- with Θ_H-simple rules. The theorem follows by repetitive applications of this construction, starting from LK^+ (that is equivalent to HCL^+, see Fact 1). Thus we transform ψ into a set R_ψ of Θ_H-simple rules such that H and $G_H = G_H^- \cup R_\psi$ are equivalent.

First, let $r_\psi = \emptyset/\{t : \psi\}$. We start by showing that H is equivalent to $G_H^- \cup \{r_\psi\}$. The direction $\Gamma \vdash_H \varphi$ implies $\vdash_{G_H^- \cup \{r_\psi\}} \Gamma \Rightarrow \varphi$ is easy and proceeds by induction on the length of the derivation in H. For the converse direction, suppose to have a proof P in $G_H^- \cup \{r_\psi\}$ of the sequent $\Gamma \Rightarrow \varphi$. Then there are substitutions $\sigma_1, \ldots, \sigma_n$, for which we can transform P into a proof of $\Gamma, \sigma_1(\psi), \ldots, \sigma_n(\psi) \Rightarrow \varphi$ in G_H^-, by replacing every application of r_ψ with the identity axiom $\{f : \sigma_i(\psi), t : \sigma_i(\psi)\}$ (and weakening), and propagating $f : \sigma_i(\psi)$ through the derivation until the end sequent. The equivalence of $H \backslash \{\psi\}$ and G_H^- entails that $\Gamma, \sigma_1(\psi), \ldots, \sigma_n(\psi) \vdash_{H \backslash \{\psi\}} \varphi$, and it immediately follows that $\Gamma \vdash_H \varphi$.

The algorithm to transform r_ψ into a set of Θ_H-simple rules works in three steps:

(Step i): We use the logical rules for $\wedge, \vee$ and $\supset$ of LK^+ to obtain a finite set R of rules such that (i) R is equivalent to $\{r_\psi\}$ in $G_H^- \cup \{r_\psi\}$, and (ii) each $r \in R$ has the form $\emptyset/s$, where s has one of the following forms (depending on whether ψ is generated by R_1 or R_2 in the grammar of Definition 2.1):

(1) s consists of at least one labelled formula of the form $b : \bar{\star} p_1$ for $b \in \{f, t\}$ and $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$ and any number of labelled formulas of the form $c : p_2$ or $c : \bar{\triangleright} p_1$ for $c \in \{f, t\}$ and $\bar{\triangleright} \in \Theta_H$.
(2) s consists of exactly one labelled formula of the form $b : \bar{\star}(p_1 \diamond p_2)$ for $b \in \{f, t\}$, $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$ and $\diamond \in \{\wedge, \vee, \supset\}$, and any number of labelled formulas of the form $c : \bar{\triangleright} p_i$ for $i \in \{1, 2\}, c \in \{f, t\}$ and $\bar{\triangleright} \in \Theta_H$.

The equivalence between $\{r_\psi\}$ and R easily follows by the invertibility of the logical rules in LK^+ (and, hence, in G_H^-). We prove that, when ψ is generated by R_2, s has the form (2) above (the proof for (1) is similar). Indeed if $(*)$ each $r'_\psi \in R$ contains exactly one labelled formula of the form $b : \bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$, then we are done. Otherwise, we apply the logical rules of LK^+ according to the outermost binary connective of some $b : \psi_j$ in $r'_\psi \in R$ until we reach condition $(*)$. We distinguish the following cases:

$b : \psi_j = t : \varphi_1 \supset \varphi_2$ (or $t : \varphi_1 \vee \varphi_2$ or $f : \varphi_1 \wedge \varphi_2$, resp.). By using $(\Rightarrow\supset)$ (or $(\Rightarrow \vee)$ or $(\wedge \Rightarrow)$, resp.), we obtain a new rule $r^1_\psi = r'_\psi$ where $b : \psi_j$ is replaced by $f : \varphi_1, t : \varphi_2$ (or $t : \varphi_1, t : \varphi_2$ or $f : \varphi_1, f : \varphi_2$, resp.) and hence it contains one binary connective less.

$b : \psi_j = t : \varphi_1 \wedge \varphi_2$ (or $f : \varphi_1 \supset \varphi_2$ or $f : \varphi_1 \vee \varphi_2$, resp.). By using $(\Rightarrow \wedge)$ (or $(\supset\Rightarrow)$ or $(\vee \Rightarrow)$, resp.), we obtain two rules $\{r^1_\psi, r^2_\psi\}$. $r^1_\psi = r'_\psi$ where $b : \psi_j$ is replaced by $t : \varphi_1$ (or $t : \varphi_1$ or $f : \varphi_1$, resp.) and $r^2_\psi = r'_\psi$ where $b : \psi_j$ is replaced by $t : \varphi_2$ (or $f : \varphi_2$ or $f : \varphi_2$, resp.). Note that $\bar{\star}(p_1 \diamond p_2)$ is *not* a subformula of ψ_j by condition (i) ((ii), resp.) in Definition 2.1.

(Step ii): Obviously, we can discard from R all rules $\emptyset/s$ for which $\{f : p_i, t : p_i\} \subseteq s$ for $i \in \{1, 2\}$, keeping the equivalence with $\{r_\psi\}$. For each rule $\emptyset/s$ remaining in R: if s

has the form (1) and it contains some $b : p_2$, by Lemma 2.21 we remove these labelled formulas and obtain an equivalent set of rules.

(Step iii): For each rule $\emptyset/s \in R$, we take (a) $b : \bar{\star}p_1, \bar{\star} \in \Theta_H \setminus \{\epsilon\}, b \in \{t, f\}$ if s is of form (1), or (b) $b : \bar{\star}(p_1 \diamond p_2), \bar{\star} \in \Theta_H \setminus \{\epsilon\}, b \in \{t, f\}$, if s is of form (2). We use Lemma 2.20 to move all remaining labelled formulas to the premises of the rule, changing their side of the sequent, to obtain a set R_ψ of Θ_H-simple rules equivalent to $\{r_\psi\}$ (the rules are Θ_H-unary, in case (a) and Θ_H-binary in case (b)).

Let G_H be $G_H^- \cup R_\psi$. G_H is equivalent to $G_H^- \cup \{r_\psi\}$ and, hence, equivalent to H. □

Example 2.23. The simple rules equivalent to the axioms $(\mathbf{c_1})$ $\neg\neg p_1 \supset p_1$ and $(\mathbf{e}^{\mathbf{r}}_{\wedge_2})$ $(\neg\neg p_1 \wedge \neg\neg p_2) \supset \neg\neg(p_1 \wedge p_2)$ (see Figure 1) are constructed as follows:

$$\begin{array}{rl} \emptyset/\{t : \neg\neg p_1 \supset p_1\} & \\ \longrightarrow^{\text{Invertibility of } (\Rightarrow\supset)} & \emptyset/\{f : \neg\neg p_1, t : p_1\} \\ \longrightarrow^{\text{Lemma 2.20}} & \{\{f : p_1\}\}/\{f : \neg\neg p_1\} \end{array}$$

$$\begin{array}{rl} \emptyset/\{t : (\neg\neg p_1 \wedge \neg\neg p_2) \supset \neg\neg(p_1 \wedge p_2))\} & \\ \longrightarrow^{\text{Invertibility of } (\Rightarrow\supset)} & \emptyset/\{f : \neg\neg p_1 \wedge \neg\neg p_2, t : \neg\neg(p_1 \wedge p_2)\} \\ \longrightarrow^{\text{Invertibility of } (\wedge\Rightarrow)} & \emptyset/\{f : \neg\neg p_1, f : \neg\neg p_2, t : \neg\neg(p_1 \wedge p_2)\} \\ \longrightarrow^{\text{Lemma 2.20}} & \{\{t : \neg\neg p_1\}, \{t : \neg\neg p_2\}\}/\{t : \neg\neg(p_1 \wedge p_2)\} \end{array}$$

These are the rules $(\neg\neg \Rightarrow)$ and $(\Rightarrow \neg\neg\wedge)$ of Example 2.11.

3. EXTRACTING EFFECTIVE SEMANTICS (STEP 2)

In what follows, we show how to algorithmically obtain effective semantics for any given Θ-simple sequent calculus. Henceforth, let Θ denote an arbitrary non-empty subset of $\mathcal{U}^*_{\mathcal{L}}$ that is closed under prefixes (in particular, $\epsilon \in \Theta$). We note that the results of this section are formulated independently of the previous one. In the context of the full "taming" procedure, the method presented in this section should be applied to G_H, the Θ_H-simple sequent calculus that is constructed from a given Hilbert calculus H as described above.

The semantic framework that we use is that of *partial non-deterministic matrices* (PNmatrices) [Baaz et al. 2013]. These structures provide a natural generalization of the notion of ordinary multi-valued logical matrices, in which connectives can have non-deterministic and partial interpretations. Therefore, truth values assigned to compound formulas can be chosen non-deterministically out of a given set of options. In PNmatrices one also allows this set of options to be empty, in order to forbid some combinations of truth values. Below we briefly adapt the basic definitions from [Baaz et al. 2013] to our context.

Definition 3.1. A *partial non-deterministic matrix* (*PNmatrix*) $\mathcal{M}$ for a propositional language $\mathcal{L}$ consists of:

(1) A set $\mathcal{V}_\mathcal{M}$ of truth values.
(2) A subset $\mathcal{D}_\mathcal{M} \subseteq \mathcal{V}_\mathcal{M}$ of designated truth values.
(3) A truth table $\diamond_\mathcal{M} : {\mathcal{V}_\mathcal{M}}^n \to P(\mathcal{V}_\mathcal{M})$ for every n-ary connective $\diamond$ of $\mathcal{L}$.

Example 3.2. The (positive fragment of the) standard classical matrix can be identified with the PNmatrix $\mathcal{M}_{CL^+}$ for $\mathcal{L}^+_{cl}$ given by: $\mathcal{V}_{\mathcal{M}_{CL^+}} = \{t, f\}$, $\mathcal{D}_{\mathcal{M}_{CL^+}} = \{t\}$, and the truth tables $\wedge, \vee, \supset$ are defined according to the classical ones (singletons are used instead of values, e.g. $\wedge_{\mathcal{M}_{CL^+}}(t, f) = \{f\}$).

Definition 3.3. Let $\mathcal{M}$ be a PNmatrix for $\mathcal{L}$.

(1) An $\mathcal{M}$-*valuation* for $\mathcal{L}$ is a function $v : \mathcal{L} \to \mathcal{V}_{\mathcal{M}}$ that respects the truth tables of $\mathcal{M}$, i.e. $v(\diamond(\varphi_1, \dots, \varphi_n)) \in \diamond_{\mathcal{M}}(v(\varphi_1), \dots, v(\varphi_n))$ for every compound formula $\diamond(\varphi_1, \dots, \varphi_n) \in \mathcal{L}$.
(2) An $\mathcal{M}$-valuation v for $\mathcal{L}$ *satisfies* (with respect to $\mathcal{M}$):
— an $\mathcal{L}$-formula φ (denoted by $v \models_{\mathcal{M}} \varphi$) if $v(\varphi) \in \mathcal{D}_{\mathcal{M}}$;
— a finite set Γ of $\mathcal{L}$-formulas (denoted by $v \models_{\mathcal{M}} \Gamma$) if $v \models_{\mathcal{M}} \varphi$ for every $\varphi \in \Gamma$;
— an $\mathcal{L}$-sequent s (denoted by $v \models_{\mathcal{M}} s$) if either $v \models_{\mathcal{M}} \varphi$ for some $t : \varphi \in s$, or $v \not\models_{\mathcal{M}} \varphi$ for some $f : \varphi \in s$.
(3) Given a set Γ of $\mathcal{L}$-formulas and a single $\mathcal{L}$-formula φ, $\Gamma \vdash_{\mathcal{M}} \varphi$ if for every $\mathcal{M}$-valuation v for $\mathcal{L}$: $v \models_{\mathcal{M}} \varphi$ whenever $v \models_{\mathcal{M}} \Gamma$.
(4) Given an $\mathcal{L}$-sequent s, $\vdash_{\mathcal{M}} s$ if $v \models_{\mathcal{M}} s$ for every $\mathcal{M}$-valuation v for $\mathcal{L}$.

FACT 3. *$\mathcal{M}_{CL^+}$ (see Example 3.2) is sound and complete for HCL^+ (i.e. $\Gamma \vdash_{HCL^+} \varphi$ iff $\Gamma \vdash_{\mathcal{M}_{CL^+}} \varphi$), as well as for LK^+ (i.e. $\vdash_{LK^+} s$ iff $\vdash_{\mathcal{M}_{CL^+}} s$).*

The following result on effectiveness of PNmatrices was established in [Baaz et al. 2013]:

FACT 4. *Let $\mathcal{M}$ be a finite PNmatrix for a propositional language $\mathcal{L}$. Given an $\mathcal{L}$-sequent s, it is decidable whether $\vdash_{\mathcal{M}} s$ or not. Similarly, given a finite set $\Gamma \cup \{\varphi\}$ of $\mathcal{L}$-formulas, it is decidable whether $\Gamma \vdash_{\mathcal{M}} \varphi$ or not.*

Next we describe an algorithmic extraction of a PNmatrix $\mathcal{M}_G$ from a Θ-simple sequent calculus G, such that $\vdash_G s$ iff $\vdash_{\mathcal{M}_G} s$. The intuitive idea is to use truth values of $\mathcal{M}_G$ as "information carriers" in the following sense. Usually, the truth values t, f determine whether a formula φ is "true" or "false". Here, in addition to this information, the truth value of φ contains also information about the "truth/falsity" of all the formulas of the form $\bar{\star}\varphi$ for every $\bar{\star} \in \Theta$. Thus, instead of using just t and f, we use *functions* from Θ to $\{t, f\}$. Since Θ always contains ϵ, the standard information whether φ is "true" or "false" is also included.

NOTATION 3. *We denote by F_Θ the set of functions $\Theta \to \{f, t\}$. For $\bar{\star} \in \Theta$ and $u \in \mathsf{F}_\Theta$, we write $u^{\bar{\star}}$ to denote $u(\bar{\star})$. For $\Theta = \{\bar{\star}_1, \dots, \bar{\star}_n\}$ we shall represent the function u such that $u^{\bar{\star}_i} = b_i$ for $1 \leq i \leq n$ by $\langle \bar{\star}_1 : b_1, \dots, \bar{\star}_n : b_n \rangle$.*

The following *consistency* property is essential:

Definition 3.4. A function $v : \mathcal{L} \to \mathsf{F}_\Theta$ is called *consistent* if $v(\bar{\triangleright}\varphi)^{\bar{\star}} = v(\varphi)^{\bar{\star}\bar{\triangleright}}$ for every formula φ and $\bar{\star}, \bar{\star}\bar{\triangleright} \in \Theta$.

Intuitively speaking, using F_Θ as a set of truth values, information about a formula may occur in more than one "place" in truth values assigned by a valuation. For example, if $\neg \in \Theta$, then the information whether $\neg\varphi$ is "true" occurs both in $v(\neg\varphi)^\epsilon$ and in $v(\varphi)^\neg$. The consistency property ensures that there are no contradictions between two "places" that store the same information.

The construction of the PNmatrix $\mathcal{M}_G$ is based on the following observations:

— The Θ-unary rules of G (Definition 2.19) "dictate" certain relationships between various formulas of the form $\bar{\star}\varphi$ for $\bar{\star} \in \Theta$. Thus in constructing $\mathcal{M}_G$, not all the possible functions in F_Θ are used as truth values, but only those that *respect the Θ-unary rules*, cf. Definition 3.5 below.
— The truth tables of the unary connectives are constructed using the information "contained" in each of the truth values concerning each connective in a way described below. These tables will guarantee that $\mathcal{M}_G$-valuations are consistent.
— The truth tables of the binary connectives are constructed using the Θ-binary rules of G.

To define $\mathcal{M}_G$, we use the following additional notions:

Definition 3.5. Let $\bar{\star} \in \Theta$ and $u_1, u_2 \in \mathsf{F}_\Theta$.

— u_1 ***satisfies*** an $\mathcal{L}$-sequent of the form $\{b : \bar{\star}p_1\}$ if $u_1^{\bar{\star}} = b$.
— u_1 ***respects*** a Θ-unary rule $Q/\{b : \bar{\star}\, p_1\}$ if it satisfies $\{b : \bar{\star}p_1\}$ whenever it satisfies every $q \in Q$.
— The ordered pair $\langle u_1, u_2 \rangle$ ***satisfies*** an $\mathcal{L}$-sequent of the form $\{b : \bar{\star}p_i\}$ if $u_i^{\bar{\star}} = b$ for $i \in \{1, 2\}$.

Definition 3.6. Given a Θ-simple sequent calculus G, the PNmatrix $\mathcal{M}_G$ is defined as follows:

— The set of truth values $\mathcal{V}_{\mathcal{M}_G}$ contains all functions in F_Θ that respect all Θ-unary rules of G.
— The set of designated truth values $\mathcal{D}_{\mathcal{M}_G}$ is $\{u \in \mathcal{V}_{\mathcal{M}_G} \mid u^\epsilon = t\}$.
— For any unary connective $\star \in \mathcal{U}_\mathcal{L}$, the truth table for $\star$ is given by:

$$\star_{\mathcal{M}_G}(u_1) = \{u \in \mathcal{V}_{\mathcal{M}_G} \mid u^{\bar{\star}} = u_1^{\bar{\star}\star} \text{ whenever } \bar{\star}\star \in \Theta\}.$$

— For $\diamond \in \{\wedge, \vee, \supset\}$ and $u_1, u_2 \in \mathcal{V}_{\mathcal{M}_G}$, $\diamond_{\mathcal{M}_G}(u_1, u_2)$ is the set of all $u \in \mathcal{V}_{\mathcal{M}_G}$ satisfying:
  (1) $u^\epsilon \in \diamond_{\mathcal{M}_{CL^+}}(u_1^\epsilon, u_2^\epsilon)$ (where $\diamond_{\mathcal{M}_{CL^+}}$ is the classical truth table of $\diamond$; see Example 3.2).
  (2) For every Θ-binary rule of G of the form $Q/\{b : \bar{\star}(p_1 \diamond p_2)\}$, if $\langle u_1, u_2 \rangle$ satisfies every $q \in Q$ then $u^{\bar{\star}} = b$.

Example 3.7. Consider the calculus H_0 that extends HCL^+ with the two axioms from Example 2.23. Then, $\Theta_{H_0} = \{\epsilon, \neg, \neg\neg\}$, and the corresponding Θ_{H_0}-simple sequent calculus G_{H_0} extends LK^+ with one unary rule $r_u = \{\{f : p_1\}\}/\{f : \neg\neg p_1\}$ and one binary rule $r_b = \{\{t : \neg\neg p_1\}, \{t : \neg\neg p_2\}\}/\{t : \neg\neg(p_1 \wedge p_2)\}$. We construct the PNmatrix $\mathcal{M} = \mathcal{M}_{G_{H_0}}$ according to Definition 3.6. We start by listing $\mathsf{F}_{\Theta_{H_0}}$:

$$\begin{aligned}\mathsf{F}_{\Theta_{H_0}} = \{&\langle \epsilon : f, \neg : f, \neg\neg : f\rangle, \langle \epsilon : f, \neg : f, \neg\neg : t\rangle, \langle \epsilon : f, \neg : t, \neg\neg : f\rangle, \langle \epsilon : f, \neg : t, \neg\neg : t\rangle,\\ &\langle \epsilon : t, \neg : f, \neg\neg : f\rangle, \langle \epsilon : t, \neg : f, \neg\neg : t\rangle, \langle \epsilon : t, \neg : t, \neg\neg : f\rangle, \langle \epsilon : t, \neg : t, \neg\neg : t\rangle\}.\end{aligned}$$

Next we need to determine the set $\mathcal{V}_\mathcal{M}$ of truth values that respect the unary rules of G_{H_0}, the only relevant such rule being r_u. Since $u \in \mathcal{V}_\mathcal{M}$ respects r_u iff $u^{\neg\neg} = f$ whenever $u^\epsilon = f$, we delete the values $\{\langle \epsilon : f, \neg : f, \neg\neg : t\rangle, \langle \epsilon : f, \neg : t, \neg\neg : t\rangle\}$ and obtain:

$$\begin{aligned}\mathcal{V}_\mathcal{M} = \{&\langle \epsilon : f, \neg : f, \neg\neg : f\rangle, \langle \epsilon : f, \neg : t, \neg\neg : f\rangle, \langle \epsilon : t, \neg : f, \neg\neg : f\rangle,\\ &\langle \epsilon : t, \neg : f, \neg\neg : t\rangle, \langle \epsilon : t, \neg : t, \neg\neg : f\rangle, \langle \epsilon : t, \neg : t, \neg\neg : t\rangle\}.\end{aligned}$$

The set of designated truth values is:

$$\mathcal{D}_\mathcal{M} = \{\langle \epsilon : t, \neg : f, \neg\neg : f\rangle, \langle \epsilon : t, \neg : f, \neg\neg : t\rangle, \langle \epsilon : t, \neg : t, \neg\neg : f\rangle, \langle \epsilon : t, \neg : t, \neg\neg : t\rangle\}.$$

Next we define the truth table for $\neg$. For every $u_1 \in \mathcal{V}_\mathcal{M}$, we take all $u \in \mathcal{V}_\mathcal{M}$ that satisfy the condition $u^{\bar{\star}} = u_1^{\bar{\star}\neg}$ (for all $\bar{\star} \in \Theta_{H_0}$ such that $\bar{\star}\neg \in \Theta_{H_0}$). For instance, let $u_1 = \langle \epsilon : f, \neg : t, \neg\neg : f\rangle$. We consider those elements u from $\mathcal{V}_\mathcal{M}$ in which $u^\epsilon = u_1^\neg = t$ and $u^\neg = u_1^{\neg\neg} = f$. The only two such elements are $\langle \epsilon : t, \neg : f, \neg\neg : f\rangle$ and $\langle \epsilon : t, \neg : f, \neg\neg : t\rangle$. The truth table for $\neg$ is thus defined as follows (we write below

$\langle x, y, z \rangle$ instead of $\langle \epsilon : x, \neg : y, \neg\neg : z \rangle$):

$\neg_{\mathcal{M}}$	
$\langle f,f,f \rangle$	$\{\langle f,f,f \rangle\}$
$\langle f,t,f \rangle$	$\{\langle t,f,f \rangle, \langle t,f,t \rangle\}$
$\langle t,f,f \rangle$	$\{\langle f,f,f \rangle\}$
$\langle t,f,t \rangle$	$\{\langle f,t,f \rangle\}$
$\langle t,t,f \rangle$	$\{\langle t,f,f \rangle, \langle t,f,t \rangle\}$
$\langle t,t,t \rangle$	$\{\langle t,t,f \rangle, \langle t,t,t \rangle\}$

Finally, we obtain the truth tables for the binary connectives. We show the case of $\wedge$. The only binary rule of G_{H_0} that involves $\wedge$ is r_b, which imposes the requirement that for every $u \in \wedge_{\mathcal{M}}(u_1, u_2)$ we have $u^{\neg\neg} = t$ whenever $u_1^{\neg\neg} = t$ and $u_2^{\neg\neg} = t$. In addition, u^{ϵ} must meet the requirements arising from the classical truth table of $\wedge$. Thus we obtain the following truth table (let $\mathcal{F}_{\mathcal{M}} = \mathcal{V}_{\mathcal{M}} \setminus \mathcal{D}_{\mathcal{M}}$):

$\wedge_{\mathcal{M}}$	$\langle f,f,f \rangle$	$\langle f,t,f \rangle$	$\langle t,f,f \rangle$	$\langle t,f,t \rangle$	$\langle t,t,f \rangle$	$\langle t,t,t \rangle$
$\langle f,f,f \rangle$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$
$\langle f,t,f \rangle$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$
$\langle t,f,f \rangle$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$
$\langle t,f,t \rangle$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\{\langle t,f,t \rangle, \langle t,t,t \rangle\}$	$\mathcal{D}_{\mathcal{M}}$	$\{\langle t,f,t \rangle, \langle t,t,t \rangle\}$
$\langle t,t,f \rangle$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$
$\langle t,t,t \rangle$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{F}_{\mathcal{M}}$	$\mathcal{D}_{\mathcal{M}}$	$\{\langle t,f,t \rangle, \langle t,t,t \rangle\}$	$\mathcal{D}_{\mathcal{M}}$	$\{\langle t,f,t \rangle, \langle t,t,t \rangle\}$

Remark 3.8. Obviously, $HCL^+ \in \mathbf{H}$ and $\Theta_{HCL^+} = \{\epsilon\}$. Now, $G_{HCL^+} = LK^+$, and the corresponding PNmatrix $\mathcal{M}_{LK^+}$ has two truth values $u_1, u_2 \in \{\epsilon\} \to \{t, f\}$, where $u_1(\epsilon) = t$ and $u_2(\epsilon) = f$. Identifying u_1, u_2 with t and f respectively leads to the classical PNmatrix $\mathcal{M}_{CL^+}$ of Example 3.2.

The next lemma asserts that $\mathcal{M}_G$ valuations are always consistent.

LEMMA 3.9. *Let G be a Θ-simple sequent calculus for a propositional language $\mathcal{L}$, and v be an $\mathcal{M}_G$-valuation for $\mathcal{L}$. Then $v(\bar{\triangleright}\varphi)^{\bar{\star}} = v(\varphi)^{\bar{\star}\bar{\triangleright}}$ for every formula φ and $\bar{\triangleright}, \bar{\star} \in \mathcal{U}^*_{\mathcal{L}}$ such that $\bar{\star}\bar{\triangleright} \in \Theta$.*

PROOF. We prove the claim by induction on the length of $\bar{\triangleright}$. The claim is trivial when $\bar{\triangleright} = \epsilon$. Suppose it holds when $\bar{\triangleright}$ is of length n, and let $\star\bar{\triangleright} \in \mathcal{U}^*_{\mathcal{L}}$ (for $\star \in \mathcal{U}_{\mathcal{L}}$). Since v is an $\mathcal{M}_G$-valuation, $v(\star\bar{\triangleright}\varphi) \in \star_{\mathcal{M}_G}(v(\bar{\triangleright}\varphi))$. By Definition 3.6, this implies that $v(\star\bar{\triangleright}\varphi)^{\bar{\star}} = v(\bar{\triangleright}\varphi)^{\bar{\star}\star}$ (note that $\bar{\star}\star \in \Theta$ since Θ is closed under prefixes and $\bar{\star}\star\bar{\triangleright} \in \Theta$). By the induction hypothesis, $v(\bar{\triangleright}\varphi)^{\bar{\star}\star} = v(\varphi)^{\bar{\star}\star\bar{\triangleright}}$. $\square$

We now come to the main result of this section, namely soundness and completeness for G with respect to $\mathcal{M}_G$.

THEOREM 3.10 (SOUNDNESS AND COMPLETENESS). *Let G be a Θ-simple sequent calculus for $\mathcal{L}$ and s_0 be an $\mathcal{L}$-sequent. Then, $\vdash_G s_0$ iff $\vdash_{\mathcal{M}_G} s_0$.*

PROOF. $\Rightarrow$: It suffices to show that the applications of the rules of G are "sound". Consider an application of a rule $r = Q/s$ of G inferring $\sigma(s) \cup c$ from $\{\sigma(q) \cup c \mid q \in Q\}$, where σ is an $\mathcal{L}$-substitution and c is an $\mathcal{L}$-sequent. Let v be an $\mathcal{M}_G$-valuation for $\mathcal{L}$. Suppose that v satisfies $\sigma(q) \cup c$ for every $q \in Q$. We show that v satisfies $\sigma(s) \cup c$. If v satisfies c, we are done. Suppose otherwise; then v satisfies $\sigma(q)$ for all $q \in Q$. We show that v satisfies $\sigma(s)$. We only consider the case when r is a Θ-unary rule (the proofs for Θ-binary and the rules of LK^+ rules are similar). Hence, $s = \{b_0 : \bar{\triangleright}\, p_1\}$ for $\bar{\triangleright} \in \Theta \setminus \{\epsilon\}$ and $b_0 \in \{t, f\}$. Let $\psi_1 = \sigma(p_1)$. We show that $v(\psi_1)$ satisfies every $q \in Q$. Let $q = \{b : \bar{\star}\, p_1\}$ be a premise in Q. Then, $\bar{\star} \in \Theta$. The fact that v satisfies $\sigma(q) = \{b : \bar{\star}\psi_1\}$

implies that $v(\bar{\star}\psi_1)^\epsilon = b$. By Lemma 3.9, $v(\psi_1)^{\bar{\star}} = b$. Hence $v(\psi_1)$ satisfies q. Since $v(\psi_1) \in \mathcal{V}_{\mathcal{M}_G}$, it respects r, and so $v(\psi_1)^{\bar{\triangleright}} = b_0$. By Lemma 3.9, $v(\bar{\triangleright}\psi_1)^\epsilon = b_0$. Thus, v satisfies $\sigma(s)$.

$\Leftarrow$: Suppose that $\nvdash_G s_0$. We construct an $\mathcal{M}_G$-valuation for $\mathcal{L}$ that does not satisfy s_0. It is a standard matter to construct a "maximal" (infinite) set Ω of labelled $\mathcal{L}$-formulas, extending s_0, that satisfies the following conditions:

(1) $\nvdash_G s$ for every $\mathcal{L}$-sequent $s \subseteq \Omega$.
(2) For every labelled $\mathcal{L}$-formula $b : \psi \notin \Omega$, we have $\vdash_G s \cup \{b : \psi\}$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.

Note that the availability of the rules (cut) and (id) implies the following two facts:

(1) For every $\mathcal{L}$-formula ψ, either $f : \psi \in \Omega$ or $t : \psi \in \Omega$. Otherwise $\vdash_G s_1 \cup \{f : \psi\}$ and $\vdash_G s_2 \cup \{t : \psi\}$ with $s_1, s_2 \subseteq \Omega$. By applying (cut) (and possibly weakenings), we obtain $\vdash_G s_1 \cup s_2$. Since $s_1 \cup s_2 \subseteq \Omega$, this contradicts the properties of Ω.
(2) For every $\mathcal{L}$-formula ψ, either $f : \psi \notin \Omega$ or $t : \psi \notin \Omega$. Otherwise $\{f : \psi, t : \psi\} \subseteq \Omega$, but $\vdash_G \{f : \psi, t : \psi\}$ using (id).

Let v be the function from $\mathcal{L}$ to F_Θ defined by $v(\psi)^{\bar{\star}} = b$ iff $b : \bar{\star}\psi \notin \Omega$ for every $\psi \in \mathcal{L}$ and $\bar{\star} \in \Theta$. By the two facts above, v is well defined. In order to show that v is an $\mathcal{M}_G$-valuation, we use the following properties:

(*). Let σ be an $\mathcal{L}$-substitution. If $v(\sigma(p_1))$ satisfies an $\mathcal{L}$-sequent q of the form $\{b : \bar{\star}p_1\}$ where $\bar{\star} \in \Theta$ then $\vdash_G s \cup \sigma(q)$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
Proof: Suppose that $v(\sigma(p_1))$ satisfies q. Thus $v(\sigma(p_1))^{\bar{\star}} = b$. It follows that $b : \bar{\star}\sigma(p_1) \notin \Omega$. Hence there is some $\mathcal{L}$-sequent $s \subseteq \Omega$ such that $\vdash_G s \cup \{b : \bar{\star}\sigma(p_1)\}$.

(**). Let σ be an $\mathcal{L}$-substitution. If $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies an $\mathcal{L}$-sequent q of the form $\{b : \bar{\star}p_i\}$ where $\bar{\star} \in \Theta$ and $i \in \{1, 2\}$ then $\vdash_G s \cup \sigma(q)$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
Proof: Suppose that $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies q. Thus $v(\sigma(p_i))^{\bar{\star}} = b$. It follows that $b : \bar{\star}\sigma(p_i) \notin \Omega$. Hence there is some $\mathcal{L}$-sequent $s \subseteq \Omega$ such that $\vdash_G s \cup \{b : \bar{\star}\sigma(p_i)\}$.

We first prove that for every $\mathcal{L}$-formula ψ, $v(\psi)$ respects all the Θ-unary rules of G, and so $v(\psi) \in \mathcal{V}_{\mathcal{M}_G}$. Let ψ be an $\mathcal{L}$-formula, and $r = Q/\{b : \bar{\star}p_1\}$ be a Θ-unary rule of G. Suppose that $v(\psi)$ satisfies every $q \in Q$. Consider an $\mathcal{L}$-substitution σ for which $\sigma(p_1) = \psi$. By (*), for every $q \in Q$ there is some sequent $s_q \subseteq \Omega$ such that $\vdash_G s_q \cup \sigma(q)$. By applying (weakenings and) r we obtain $\vdash_G \bigcup_{q\in Q} s_q \cup \{b : \bar{\star}\psi\}$. Thus, $\{b : \bar{\star}\psi\} \notin \Omega$ and so $v(\psi)^{\bar{\star}} = b$.

Next, we show that v respects the truth tables of $\mathcal{M}_G$:

(1) Let $\star \in \mathcal{U}_\mathcal{L}$ and $\psi \in \mathcal{L}$. We show that $v(\star\psi) \in \star_{\mathcal{M}_G}(v(\psi))$. By the definition of $\mathcal{M}_G$, it suffices to show that $v(\star\psi)^{\bar{\star}} = v(\psi)^{\bar{\star}\star}$ whenever $\bar{\star}\star \in \Theta$. This follows directly from the definition of v.
(2) Let $\diamond \in \{\wedge, \vee, \supset\}$ and $\psi_1, \psi_2 \in \mathcal{L}$. We show that $v(\psi_1 \diamond \psi_2) \in \diamond_{\mathcal{M}_G}(v(\psi_1), v(\psi_2))$ by showing (i) $v(\psi_1 \diamond \psi_2)^\epsilon \in \diamond_{\mathcal{M}_{CL^+}}(v(\psi_1)^\epsilon, v(\psi_2)^\epsilon)$ and (ii) $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = b$ for every Θ-binary rule $r = Q/\{b : \bar{\star}(p_1 \diamond p_2)\}$ of G for which $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$.
   (i) We prove (i) for the specific case when $\diamond = \wedge$ and $v(\psi_1)^\epsilon = v(\psi_2)^\epsilon = t$. All other cases are handled similarly. We show that $v(\psi_1 \wedge \psi_2)^\epsilon = t$. The definition of v ensures that both $t : \psi_1$ and $t : \psi_2$ do not occur in Ω. Thus $\vdash_G s_1 \cup \{t : \psi_1\}$ and $\vdash_G s_2 \cup \{t : \psi_2\}$ for some $s_1, s_2 \subseteq \Omega$. By applying (weakenings and) the

rule $(\Rightarrow \wedge)$ of LK^+, we obtain that $\vdash_G s_1 \cup s_2 \cup \{t : \psi_1 \wedge \psi_2\}$. Hence we have $t : \psi_1 \wedge \psi_2 \notin \Omega$. It follows that $v(\psi_1 \wedge \psi_2)^\epsilon = t$.

(ii) Let $r = Q/\{b : \bar{\star}(p_1 \diamond p_2)\}$ be a Θ-binary rule of G, such that $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. Consider an $\mathcal{L}$-substitution σ for which $\sigma(p_1) = \psi_1$ and $\sigma(p_2) = \psi_2$. By (**), for every $q \in Q$ there is some sequent $s_q \subseteq \Omega$ such that $\vdash_G s_q \cup \sigma(q)$. By applying (weakenings and) r we obtain that $\vdash_G \bigcup_{q\in Q} s_q \cup \{b : \bar{\star}(\psi_1 \diamond \psi_2)\}$. Thus, $\{b : \bar{\star}(\psi_1 \diamond \psi_2)\} \notin \Omega$ and so $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = b$.

Finally, note that v does not satisfy s_0. Indeed, every $b : \psi \in s_0$ is also an element of Ω, and hence $v(\psi)^\epsilon \neq b$. Thus we have $v \not\models_{\mathcal{M}_G} \psi$ for every $t : \psi \in s$, and $v \models_{\mathcal{M}_G} \psi$ for every $f : \psi \in s$. □

Combining the previous theorem with Theorem 2.22, we get:

COROLLARY 3.11. *Let $H \in \mathbf{H}$ be a Hilbert calculus for $\mathcal{L}$. For every finite set $\Gamma \cup \{\varphi\}$ of $\mathcal{L}$-formulas, $\Gamma \vdash_H \varphi$ iff $\Gamma \vdash_{\mathcal{M}_{G_H}} \varphi$.*

PROOF. Suppose that $\Gamma \vdash_H \varphi$. By Theorem 2.22, we have $\vdash_{G_H} \Gamma \Rightarrow \varphi$. Theorem 3.10 implies that $\vdash_{\mathcal{M}_{G_H}} \Gamma \Rightarrow \varphi$. By definition, it follows that $\Gamma \vdash_{\mathcal{M}_{G_H}} \varphi$. The converse is similar. □

Using Fact 4, we also obtain a general decidability result:

COROLLARY 3.12 (DECIDABILITY). *Given a Hilbert system $H \in \mathbf{H}$ and a finite set $\Gamma \cup \{\varphi\}$ of formulas, it is decidable whether $\Gamma \vdash_H \varphi$ or not.*

PROOF. Follows by Corollary 3.11 and Fact 4. □

4. ANALYTICITY

In Section 2, each Hilbert axiom **H** is transformed into a set of sequent rules that introduce logical connectives. However, determining whether the obtained calculus as a whole is well-behaved (or analytic), depends on the interplay between all the rules and requires a view of the calculus as a whole. We show below that such view is provided by the semantic framework introduced in Section 3.

Roughly speaking, a sequent calculus is analytic if, whenever a sequent s is provable in it, it can also be proven by using only the "syntactic material available within s". Usually, this "material" consists of all subformulas occurring in s (in this case 'analyticity' is just another name for the global subformula property), denoted henceforth by $sub[s]$. However, weaker variants have also been considered in the literature, especially in paraconsistent and modal logics, e.g., [Avron 2007; Lellmann and Pattinson 2011; Bezhanishvili and Ghilardi 2013]. In this paper we use the following notion of analyticity, which takes as "syntactic material" not only subformulas but also their sequences with unary connectives.

NOTATION 4. *For $\Theta \subseteq \mathcal{U}^*_{\mathcal{L}}$ and set $\mathcal{W}$ of $\mathcal{L}$-formulas, we denote the set $\{\bar{\star}\psi \mid \bar{\star} \in \Theta, \psi \in \mathcal{W}\}$ by $\Theta(\mathcal{W})$.*

Definition 4.1. Let G be a sequent calculus (for a propositional language $\mathcal{L}$).

(1) Given an $\mathcal{L}$-sequent s and a set $\mathcal{W}$ of $\mathcal{L}$-formulas, we write $\vdash_G^{\mathcal{W}} s$ if there exists a proof of s in G consisting only of ($\mathcal{L}$-sequents that consist of) formulas from $\mathcal{W}$.
(2) Given a set $\Theta \subseteq \mathcal{U}^*_{\mathcal{L}}$, G is called Θ-*analytic* if $\vdash_G s$ implies $\vdash_G^{\Theta(sub[s])} s$ for every sequent s.

Note that our notion of analyticity is closely related to the notion of *bounded proof property* (defined in [Bezhanishvili and Ghilardi 2013] in the context of modal logic).

According to this property the complexity of formulas appearing in s determines the bound on the complexity of the subformulas of s that are allowed to appear in the proof.

Let us take stock of what we have achieved so far. Given a Hilbert calculus $H \in \mathbf{H}$ we introduced an equivalent Θ_H-simple sequent calculus G_H and extracted a suitable finite-valued semantics out of it (the PNmatrix $\mathcal{M}_{G_H}$). We show below that we can use $\mathcal{M}_{G_H}$ to obtain a useful decidable sufficient condition for the Θ_H-analyticity of G_H. All we need is a simple check whether $\mathcal{M}_{G_H}$ is *proper* (i.e., it is an Nmatrix):

Definition 4.2 (*[Baaz et al. 2013]*). A PNmatrix $\mathcal{M}$ for $\mathcal{L}$ is called *proper* if its set of truth values $\mathcal{V}_\mathcal{M}$ is non-empty and $\diamond_\mathcal{M}(x_1, \ldots, x_n) \neq \emptyset$ for every n-ary connective $\diamond$ of $\mathcal{L}$ and $x_1, \ldots, x_n \in \mathcal{V}_\mathcal{M}$.

THEOREM 4.3. *Let G be a Θ-simple sequent calculus. If $\mathcal{M}_G$ is proper then G is Θ-analytic.*

PROOF. Suppose that $\mathcal{M}_G$ is proper and $\nvdash_G^{\Theta(sub[s_0])} s_0$ for some $\mathcal{L}$-sequent s_0. We show that $\nvdash_G s_0$. By Theorem 3.10, it suffices to show that there exists an $\mathcal{M}_G$-valuation for $\mathcal{L}$ that does not satisfy s_0. Let $\mathcal{W} = \Theta(sub[s_0])$. It is a standard matter to extend s_0 into a "maximal" $\mathcal{L}$-sequent s^* that satisfies the following conditions:

(1) s^* consists of labelled $\mathcal{L}$-formulas of the form $b : \psi$ and $\psi \in \mathcal{W}$.
(2) $\nvdash_G^\mathcal{W} s^*$.
(3) For every labelled $\mathcal{L}$-formula $b : \psi$ with $\psi \in \mathcal{W}$, if $b : \psi \notin s^*$ then $\vdash_G^\mathcal{W} s^* \cup \{b : \psi\}$.

As in the proof of Theorem 3.10, the availability of the rules (cut) and (id) implies that for every $\psi \in \mathcal{W}$ there is a unique $b \in \{t, f\}$ such that $b : \psi \in s^*$. Next, we define a function $v : \mathcal{L} \to \mathsf{F}_\Theta$ by induction on the structure of formulas. Suppose that $v(\varphi)$ is defined for every proper subformula φ of an $\mathcal{L}$-formula ψ. We define $v(\psi)$ as follows. First, if $\psi \in sub[s_0]$ then for every $\bar{\star} \in \Theta$: $v(\psi)^{\bar{\star}} = b$ iff $b : \bar{\star}\psi \notin s^*$. Otherwise, if ψ is an atomic formula, $v(\psi)$ is arbitrarily chosen to be one of the truth values in $\mathcal{V}_{\mathcal{M}_G}$. Otherwise, $\psi = \diamond(\psi_1, \ldots, \psi_n)$ is a compound formula, and in this case $v(\psi)$ is arbitrarily chosen to be one of the truth values in $\diamond_{\mathcal{M}_G}(v(\psi_1), \ldots, v(\psi_n))$. Note that the fact that $\mathcal{M}_G$ is proper guarantees that these arbitrary choices are always possible. In order to show that v is an $\mathcal{M}_G$-valuation, we use the following properties:

(*). Let σ be an $\mathcal{L}$-substitution such that $\sigma(p_1) \in sub[s_0]$. If $v(\sigma(p_1))$ satisfies an $\mathcal{L}$-sequent of the form $\{b : \bar{\star}p_1\}$ where $\bar{\star} \in \Theta$ then $\vdash_G^\mathcal{W} s^* \cup \{b : \bar{\star}\sigma(p_1)\}$.
Proof: Suppose that $v(\sigma(p_1))$ satisfies $\{b : \bar{\star}p_1\}$. Thus $v(\sigma(p_1))^{\bar{\star}} = b$. Since $\sigma(p_1) \in sub[s_0]$, we have that $b : \bar{\star}\sigma(p_1) \notin s^*$. The maximality of s^* ensures that $\vdash_G^\mathcal{W} s^* \cup \{b : \bar{\star}\sigma(p_1)\}$.

(**). Let σ be an $\mathcal{L}$-substitution such that $\{\sigma(p_1), \sigma(p_2)\} \subseteq sub[s_0]$. If $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies an $\mathcal{L}$-sequent q of the form $\{b : \bar{\star}p_i\}$ where $\bar{\star} \in \Theta$ and $i \in \{1, 2\}$ then $\vdash_G^\mathcal{W} s^* \cup \sigma(q)$.
Proof: Suppose that $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies q. Thus $v(\sigma(p_i))^{\bar{\star}} = b$. Since $\sigma(p_i) \in sub[s_0]$, we have that $b : \bar{\star}\sigma(p_i) \notin s^*$. The maximality of s^* ensures that $\vdash_G^\mathcal{W} s^* \cup \sigma(q)$.

We first prove that for every $\mathcal{L}$-formula ψ, we have $v(\psi) \in \mathcal{V}_{\mathcal{M}_G}$. If $\psi \notin sub[s_0]$, this holds by definition. Suppose that $\psi \in sub[s_0]$. We show that $v(\psi)$ respects all the Θ-unary rules of G. Let $r = Q/\{b : \bar{\star}p_1\}$ be such a Θ-unary rule of G. Suppose that $v(\psi)$ satisfies every $q \in Q$. Let σ be any $\mathcal{L}$-substitution assigning ψ to p_1. By (*), $\vdash_G^\mathcal{W} s^* \cup \sigma(q)$ for every $q \in Q$. By applying r we obtain $\vdash_G^\mathcal{W} s^* \cup \{b : \bar{\star}\psi\}$. Hence $v(\psi)$ satisfies $\{b : \bar{\star}p_1\}$.

Thus, $\{b : \bar{\star}\psi\} \notin s^*$ and so $v(\psi)^{\bar{\star}} = b$. Next, we show that v respects the truth tables of $\mathcal{M}_G$.

(1) Let $\star \in \mathcal{U}_{\mathcal{L}}$ and $\psi \in \mathcal{L}$. We show that $v(\star\psi) \in \star_{\mathcal{M}_G}(v(\psi))$. This holds by definition when $\star\psi \notin sub[s_0]$. Suppose now that $\star\psi \in sub[s_0]$ (and so $\psi \in sub[s_0]$ as well). By the definition of $\mathcal{M}_G$, it suffices to show that $v(\star\psi)^{\bar{\star}} = v(\psi)^{\bar{\star}\star}$ whenever $\bar{\star}\star \in \Theta$. This follows directly from the definition of v.

(2) Let $\diamond \in \{\wedge, \vee, \supset\}$ and $\psi_1, \psi_2 \in \mathcal{L}$. We show that $v(\psi_1 \diamond \psi_2) \in \diamond_{\mathcal{M}_G}(v(\psi_1), v(\psi_2))$. This holds by definition when $\psi_1 \diamond \psi_2 \notin sub[s_0]$. Suppose now that $\psi_1 \diamond \psi_2 \in sub[s_0]$ (and so ψ_1 and ψ_2 are in $sub[s_0]$ as well). We prove (i) $v(\psi_1 \diamond \psi_2)^{\epsilon} \in \diamond_{\mathcal{M}_{CL^+}}(v(\psi_1)^{\epsilon}, v(\psi_2)^{\epsilon})$ and (ii) $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = b$ for every Θ-binary rule $r = Q/\{b : \bar{\star}(p_1 \diamond p_2)\}$ of G for which $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. (i) is straightforward. For (ii), let $r = Q/\{b : \bar{\star}(p_1 \diamond p_2)\}$ be a Θ-binary rule of G such that $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. Consider an $\mathcal{L}$-substitution σ for which $\sigma(p_1) = \psi_1$ and $\sigma(p_2) = \psi_2$. By (**), for every $q \in Q$ we have $\vdash_G^{\mathcal{W}} s^* \cup \sigma(q)$. By applying r we obtain $\vdash_G^{\mathcal{W}} s^* \cup \{b : \bar{\star}(\psi_1 \diamond \psi_2)\}$. Thus, $\{b : \bar{\star}(\psi_1 \diamond \psi_2)\} \notin s^*$ and so $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = b$.

Finally, it is immediate to see that v does not satisfy s_0. Indeed, every $b : \psi \in s_0$ occurs also in s^*, and thus $v(\psi)^{\epsilon} \neq b$. $\square$

4.1. Recovering analyticity

Not all $H \in \mathbf{H}$ lead to Θ_H-simple calculi with proper PNmatrices.

Example 4.4. Let H_1 be the calculus obtained by extending HCL^+ by the axioms $(\mathbf{n_1})$, $(\mathbf{b})$, $(\mathbf{k})$, $(\mathbf{o}^{\mathbf{r}}_{\wedge_1})$, and $(\mathbf{o}^{\mathbf{1}}_{\wedge})$ (cf. Figure 1). The PNmatrix $\mathcal{M}_{G_{H_1}}$ associated with H_1 is not proper (see G_{H_1} and $\mathcal{M}_{G_{H_1}}$ in Example 5.1 below).

Theorem 4.3 does not apply to Θ-simple calculi G whose associated matrix $\mathcal{M}_G$ is not proper. As shown below, however, analyticity can be recovered by a transformation of G into a *family* of analytic calculi, equivalent to G in the sense defined below. This is done by transforming $\mathcal{M}_G$ into a *finite family* of *proper* PNmatrices, which satisfy the following property:

Definition 4.5. ([Baaz et al. 2013]) Let $\mathcal{M}$ and $\mathcal{N}$ be PNmatrices for $\mathcal{L}$. We say that $\mathcal{N}$ is a *simple refinement* of $\mathcal{M}$ if $\mathcal{V}_{\mathcal{N}} \subseteq \mathcal{V}_{\mathcal{M}}$, $\mathcal{D}_{\mathcal{N}} = \mathcal{D}_{\mathcal{M}} \cap \mathcal{V}_{\mathcal{N}}$, and $\diamond_{\mathcal{N}}(x_1, \ldots, x_n) \subseteq \diamond_{\mathcal{M}}(x_1, \ldots, x_n)$ for every n-ary connective $\diamond$ of $\mathcal{L}$ and $x_1, \ldots, x_n \in \mathcal{V}_{\mathcal{N}}$.

Note that a simple refinement of a non-proper PNmatrix may be proper, since $\mathcal{V}_{\mathcal{N}}$ may be strictly contained in $\mathcal{V}_{\mathcal{M}}$.

THEOREM 4.6. *For every finite PNmatrix $\mathcal{M}$ for $\mathcal{L}$, there is an algorithm for constructing $\mathcal{M}_1, \ldots, \mathcal{M}_n$, such that: (i) $\mathcal{M}_1, \ldots, \mathcal{M}_n$ are finite proper simple refinements of $\mathcal{M}$, and (ii) $\vdash_{\mathcal{M}} = \bigcap_{i=1,\ldots,n} \vdash_{\mathcal{M}_i}$.*

PROOF. Let $\mathcal{M}$ be a PNmatrix for $\mathcal{L}$. Choose $\mathcal{M}_1, \ldots, \mathcal{M}_n$ to be all simple refinements of $\mathcal{M}$ which are proper PNmatrices. Based on the results in [Baaz et al. 2013], we show that $\vdash_{\mathcal{M}} = \bigcap_{i=1,\ldots,n} \vdash_{\mathcal{M}_i}$. $(\Rightarrow)$ By Proposition 1 in [Baaz et al. 2013], $\vdash_{\mathcal{M}} \subseteq \vdash_{\mathcal{N}}$ for every simple refinement $\mathcal{N}$ of $\mathcal{M}$. Therefore, $\vdash_{\mathcal{M}} \subseteq \bigcap_{i=1,\ldots,n} \vdash_{\mathcal{M}_i}$.
$(\Leftarrow)$ Suppose that $\not\vdash_{\mathcal{M}} s$. Thus $v \not\models_{\mathcal{M}} s$ for some $\mathcal{M}$-valuation v for $\mathcal{L}$. Theorem 1 in [Baaz et al. 2013] ensures that there exists some $\mathcal{M}_i$, such that v is an $\mathcal{M}_i$-valuation. The fact that $v \not\models_{\mathcal{M}} s$ easily entails that $v \not\models_{\mathcal{M}_i} s$, and so $\not\vdash_{\mathcal{M}_i} s$. $\square$

Now, once we have a finite family of proper PNmatrices, we can apply the method of [Avron et al. 2006]. This method produces a cut-free sequent calculus for any proper PNmatrix $\mathcal{M}$, whose set of designated truth values ($\mathcal{D}_{\mathcal{M}}$) is a non-empty proper subset

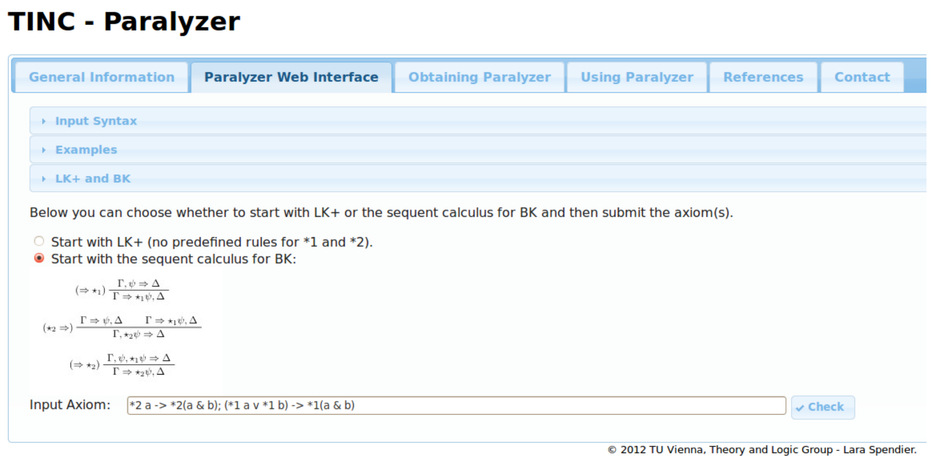

Fig. 3. Main screen of *Paralyzer* (a,b stand for atomic formulas and *1,*2 for unary connectives)

of the set of its truth values ($\mathcal{V}_\mathcal{M}$), provided that its language satisfies the following (slightly reformulated) condition:

Definition 4.7. Let $\mathcal{M}$ be a proper PNmatrix for $\mathcal{L}$. We say that $\mathcal{L}$ is *sufficiently expressive* for $\mathcal{M}$ if for any $x \in \mathcal{V}_\mathcal{M}$, there exists a set $\mathcal{S}_x$ of $\mathcal{L}$-sequents, each of which has the form $\{b : \psi\}$ for some $b \in \{f, t\}$ and $\psi \in \mathcal{L}$ in which p_1 is the only atomic variable, such that the following condition holds:

> For any $\mathcal{M}$-valuation v for $\mathcal{L}$ and $\mathcal{L}$-substitution σ, $v(\sigma(p_1)) = x$ iff v satisfies every $\mathcal{L}$-sequent in $\sigma(\mathcal{S}_x)$ with respect to $\mathcal{M}$.

COROLLARY 4.8. *For any $H \in \mathbf{H}$, there is an algorithm for constructing a family of Θ_H-analytic sequent calculi $\mathbf{F}_H$, such that for every finite set $\Gamma \cup \{\varphi\}$ of formulas: $\Gamma \vdash_H \varphi$ iff $\vdash_G \Gamma \Rightarrow \varphi$ for every $G \in \mathbf{F}_H$.*

PROOF. We start by constructing $\mathcal{M}_{G_H}$. If $\mathcal{D}_{\mathcal{M}_{G_H}} = \emptyset$ or $\mathcal{D}_{\mathcal{M}_{G_H}} = \mathcal{V}_{\mathcal{M}_{G_H}}$, then $\mathcal{M}_{G_H}$ has a trivial corresponding Θ_H-analytic calculus. For the rest of the cases, we can apply Theorem 4.6 to obtain an equivalent family of proper PNmatrices. Next we show that $\mathcal{L}$ is sufficiently expressive for any simple refinement of $\mathcal{M}_{G_H}$. Indeed, for $x \in \mathcal{V}_{\mathcal{M}_{G_H}}$, define $\mathcal{S}_x = \{x^\epsilon : p_1\} \cup \{x^{\bar{\star}} : \bar{\star} p_1 \mid \bar{\star} \in \Theta_H\}$. Let $\mathcal{M}$ be a simple refinement of $\mathcal{M}_{G_H}$ and let v be an $\mathcal{M}$-valuation for $\mathcal{L}$. The required condition is met by the fact that for every $\bar{\star} \in \Theta_H$ and $\psi \in \mathcal{L}$, $v(\bar{\star}\psi)^\epsilon = v(\psi)^{\bar{\star}}$. By the method of [Avron et al. 2006], we obtain a family of corresponding cut-free calculi. The forms of the $\mathcal{S}_x$-s guarantee that the rules of the obtained calculi are Θ_H-simple, which together with their cut-admissibility implies Θ_H-analyticity. □

5. PARALYZER

Our "taming" procedure is implemented in Prolog in the system *Paralyzer*[4] (PARAconsistent (and other) logics anaLYZER), available as part of the **TINC**-project[5] at

www.logic.at/tinc/webparalyzer/

The main page of the tool is illustrated in Figure 3. The user should provide as input:

[4]Paralyzer currently implements the grammar in [Ciabattoni et al. 2013].
[5]TINC stands for "Tools for the Investigation of Non-Classical logics".

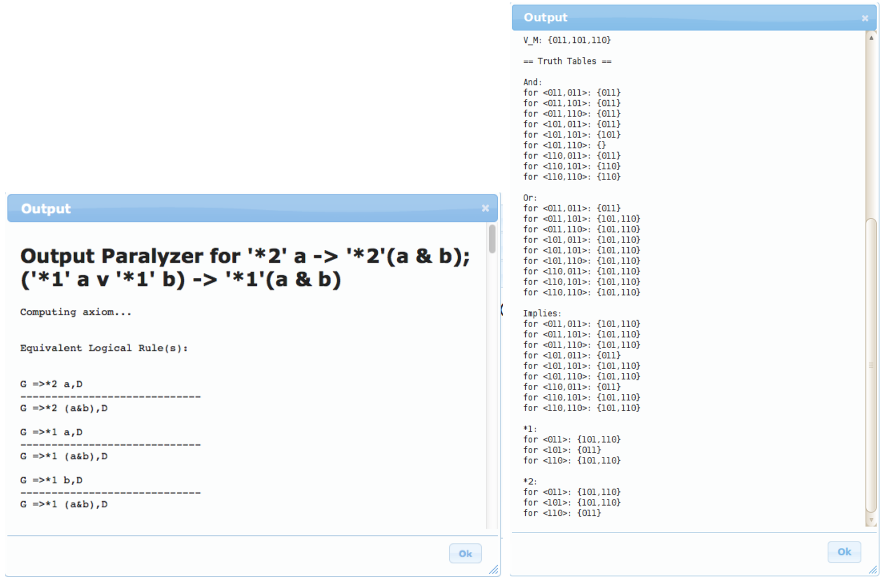

Fig. 4. Computation step (a): set of logical rules; (b): PNmatrix

(i) a set of axioms that falls into the specified grammar satisfying the imposed condition (in Figure 3 the axioms are $(\mathbf{o}^1_\wedge)$ $\circ p_1 \supset \circ(p_1 \wedge p_2)$ and $(\mathbf{o}^r_{\wedge_1})$ $(\neg p_1 \vee \neg p_2) \supset \neg(p_1 \wedge p_2)$ where $\neg$ is denoted as `*1`, $\circ$ as `*2`, p_1 as `a`, and p_2 as `b`), and

(ii) the base calculus, that is the calculus we want to extend with the generated rules.

The default option for the base calculus is LK^+. A second option for (ii) is the sequent calculus in [Avron et al. 2012; 2013] for the basic Logic of Formal Inconsistency BK, which is LK^+ extended with the rule $(\Rightarrow \neg)$ $\{\{f : p_1\}\}/\{t : \neg p_1\}$ for $\neg$ and the two (invertible) rules for the connective $\circ$:

$$(\circ \Rightarrow)\ \{\{t : p_1\}, \{t : \neg p_1\}\}/\{f : \circ p_1\} \qquad (\Rightarrow \circ)\ \{\{f : p_1, f : \neg p_1\}\}/\{t : \circ p_1\}$$

Recall that BK is obtained by extending HCL^+ with the axioms $(\mathbf{n_1})$ $p_1 \vee \neg p_1$, $(\mathbf{b})$ $p_1 \supset (\neg p_1 \supset (\circ p_1 \supset p_2))$ and $(\mathbf{k})$ $\circ p_1 \vee (p_1 \wedge \neg p_1)$ from Figure 1, where the intuitive meaning of $\circ\psi$ is that "ψ is consistent".

After the user has provided (i) and (ii), *Paralyzer* gives as output (a paper written in LaTeX containing)

(a) the set of logical rules equivalent to the input axioms in the base calculus indicated in (ii),
(b) the PNmatrix $\mathcal{M}_{G_H}$ for the newly obtained sequent calculus G_H, and
(c) an encoding of G_H for the generic proof assistant *Isabelle* [Nipkow, Paulson and Wenzel 2002; Wenzel, Paulson and Nipkow 2008].

Example 5.1. Let H_1 be the system in Example 4.4 obtained by extending HCL^+ with the axioms $(\mathbf{n_1})$, $(\mathbf{b})$, $(\mathbf{k})$, $(\mathbf{o}^1_\wedge)$ $\circ p_1 \supset \circ(p_1 \wedge p_2)$ and $(\mathbf{o}^r_{\wedge_1})$ $(\neg p_1 \vee \neg p_2) \supset \neg(p_1 \wedge p_2)$. H_1 is then BK extended with the axioms $(\mathbf{o}^1_\wedge)$ and $(\mathbf{o}^r_{\wedge_1})$. The output of *Paralyzer* computed in steps (a) and (b) is displayed in Figure 4 (`*1` stands for $\neg$, `*2` for $\circ$, `a` for p_1, and `b` for p_2). Note that the generated rules are displayed in the standard sequent notation where `G` and `D` abbreviate the sequent contexts Γ and Δ. In the PNmatrix the truth

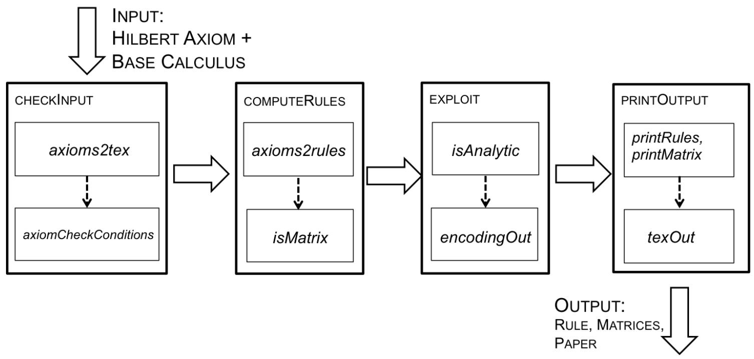

Fig. 5. Implementation details *Paralyzer*

values 0,1 are used instead of f, t, while functions $\langle \epsilon : x, \neg : y, \circ : z\rangle$ are abbreviated as $\langle xyz\rangle$.

Remark 5.2. When selecting BK as base calculus in step (ii), *Paralyzer* also exploits the invertibility of the rules $(\circ \Rightarrow)$ and $(\Rightarrow \circ)$ for defining the calculus rules. Notice that for the C-systems with finite-valued semantics, its output produces the same sequent calculi and associated semantics as those introduced in a semi-automated[6] way in [Avron et al. 2012; 2013].

5.1. Implementation Details

Paralyzer is implemented in Prolog. The general structure of the implementation is depicted in Figure 5.

The input formula is provided as a parameter to the first component CHECKINPUT and syntactically checked by the method *axioms2tex* and the function *axiomCheck-Conditions*. If the formula passes all syntactic checks it can be processed by the second component COMPUTERULES which contains the implementation of our procedure. The method *axioms2rules* transforms the axioms given as input into equivalent sequent rules (Step 1) and the function *isMatrix* extracts a partial non-deterministic matrix out of the newly generated calculus (Step 2).

The third component EXPLOIT uses the PNmatrix to check the analyticity of the calculus (*isAnalytic*). Moreover, the method *encodingOut* constructs a formalization of the calculus in the language of the generic proof assistant *Isabelle* that allows to perform proof search within the encoded logic by applying tactics and automated procedures. Figure 6 shows the application forms of the rules $(\Rightarrow \neg), (\Rightarrow \circ)$ and $(\circ \Rightarrow)$ of BK and their *Isabelle*-encodings.

The last component PRINTOUTPUT contains the methods for creating a textual representation of the calculus and its semantics on the command-line or web interface (*printRules*); moreover, the generated LaTeX-paper contains the resulting calculus (and information, whether it is analytic or not), its PNmatrix and the *Isabelle*-encoding (*texOut*).

6. CONCLUSIONS

In this paper we have handled a large family **H** of Hilbert calculi, obtained by (i) extending the language of CL^+ with finitely many unary connectives, and (ii) adding

[6]The construction of the Nmatrices (proper PNmatrix, in our terminology) out of the Hilbert calculi is done manually in [Avron et al. 2012; 2013], and it requires some ingenuity.

	Rule applications	Encoding
$(\Rightarrow \star_1)$	$\dfrac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma \Rightarrow \star_1\varphi, \Delta}$	`$H,P |- $E,$F ==> $H |- $E,~P,$F`
$(\Rightarrow \star_2)$	$\dfrac{\Gamma, \varphi, \star_1\varphi \Rightarrow \Delta}{\Gamma \Rightarrow \star_2\varphi, \Delta}$	`$H,P,~P |- $E,$F ==> $H |- $E,+P,$F`
$(\star_2 \Rightarrow)$	$\dfrac{\Gamma \Rightarrow \varphi, \Delta \quad \Gamma \Rightarrow \star_1\varphi, \Delta}{\Gamma, \star_2\varphi \Rightarrow \Delta}$	`[| $H,$G |- $E,P; $H,$G |- $E,~P |] ==> $H,+P,$G |- $E`

Fig. 6. Rule applications for $\star_1 = \neg$ and $\star_2 = \circ$ of BK and their encoding in *Isabelle*. In the encoding we use $\sim$ to denote $\star_1$ and + for $\star_2$. Upper-case letters denote single formulas, while upper-case letters preceded by $ denote (possibly empty) sequences of formulas. Rules premises are encoded left of ==> (within brackets `[|, |]` and comma-separated (`;`) in case of multiple premises) while the conclusions are right of ==>.

to any Hilbert axiomatization of CL^+ suitable axioms over the extended language. We introduced an algorithm, which for any system from **H** generates an equivalent sequent calculus (Step 1) and effective semantics based on PNmatrices (Step 2) for it. The semantics was used to show the decidability of the logics and to provide a simple sufficient condition for the analyticity of the corresponding calculi.

The family **H** includes many C-systems [Da Costa 1974; Carnielli and Marcos 2002], for which a semi-automated procedure to define semantics and analytic calculi was introduced in [Avron et al. 2012; 2013]. It should be noted that the method in [Avron et al. 2012; 2013] is quite different from ours: it works by first constructing the corresponding Nmatrix and then extracting an analytic sequent calculus from that. Note also that although this last step is algorithmic, the construction of Nmatrices out of the Hilbert calculi is done manually, and it requires some ingenuity. In this paper we fill this gap for all C-systems having a finite-valued Nmatrix, as well as for many other logics, by *fully* automatizing the generation of effective semantics and analytic calculi for them.

This paper generalizes our previous results in [Ciabattoni et al. 2013] by extending **H** to include (infinitely many) new axioms containing nesting of unary connectives of *any* fixed depth. This allows us to consider new logics (e.g. those in [Kamide 2013]) that could not be dealt with in our previous approach. While Step 1 of our procedure easily extended to the new axioms, Step 2 required major changes that were reflected in the construction of our PNmatrices, in the new notion of analyticity, and in the more involved corresponding proofs of soundness, completeness, and analyticity.

The system *Paralyzer* implements our algorithm and provides an encoding of the introduced calculi for the generic proof assistant *Isabelle*.

Our work is a concrete step towards automated construction of analytic sequent calculi and effective semantics for paraconsistent (and related) logics. Many practical and theoretical issues are still to be addressed, e.g., extension of our algorithm to first-order logics, larger classes of axioms, substructural logics, etc. Regarding this, note that while Step 1 of our procedure, which essentially follows the ideas in [Ciabattoni et al. 2008], could be easily adapted to capture a larger class of logics (e.g. the infinite-valued logic L_ω investigated in [Kamide 2009]), the construction of the corresponding PNmatrices would require a deeper investigation. For the time being there is indeed no theory of PNmatrices for first-order logics, intuitionistic logics or substructural logics (that in fact lack even a theory of Nmatrices). Characterizing the logics for which such theory can be developed is a a natural direction for identifying the borders of our methodology.

On a more practical level, the encoding of the calculi computed by *Paralyzer* allows us to find proofs of theorems in the considered logics in a semi-automated way. The definition of automated deduction procedures is currently under investigation; a possible approach is to try extending the reduction of analytic “pure” sequent calculi to SAT recently proposed in [Lahav and Zohar 2014] to cover all the calculi we generate. If successful, this approach would lead to a procedure that transforms **H** systems into automated SAT-based deduction procedures.

REFERENCES

Arnon Avron. 2005. Logical non-determinism as a tool for logical modularity: An introduction. In *We Will Show Them: Essays in Honor of Dov Gabbay*. College Publications, 105–124.

Arnon Avron. 2007. Non-deterministic Semantics for Families of Paraconsistent Logics. In *Handbook of Paraconsistency*, Jean-Yves Beziau, Walter A. Carnielli, and Dov Gabbay (Eds.). Studies in Logic, Vol. 9. College Publications, 285–320.

Arnon Avron, Jonathan Ben-Naim, and Beata Konikowska. 2006. Cut-free ordinary sequent calculi for logics having generalized finite-valued semantics. *Logica Universalis* 1 (2006), 41–69.

Arnon Avron, Beata Konikowska, and Anna Zamansky. 2012. Modular Construction of Cut-free Sequent Calculi for Paraconsistent Logics. In *Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*. 85–94.

Arnon Avron, Beata Konikowska, and Anna Zamansky. 2013. Cut-free sequent calculi for C-systems with generalized finite-valued semantics. *Journal of Logic and Computation*, 21, 3 (2013), 517–540.

Arnon Avron and Iddo Lev. 2005. Non-deterministic Multiple-valued Structures. *Journal of Logic and Computation* 15, 3 (2005), 241–261.

Arnon Avron and Anna Zamansky. 2011. Non-deterministic semantics for logical systems - A survey. In *Handbook of Philosophical Logic*, Dov Gabbay and Franz Guenthner (Eds.). Vol. 16. Springer, 227–304.

Matthias Baaz, Christian G. Fermüller, Gernot Salzer and Richard Zach. 1996. MUltlog 1.0: Towards an expert system for many-valued logics.In *Proceedings of CADE (Lecture Notes in Artificial Intelligence)*, Vol. 1104. Springer, 226–230.

Matthias Baaz, Ori Lahav, and Anna Zamansky. 2013. Finite-valued Semantics for Canonical Labelled Calculi. *Journal of Automated Reasoning* (2013), 1–30.

Nick Bezhanishvili and Silvio Ghilardi. 2013. Bounded proofs and step frames. In *Proceedings of TABLEAUX (Lecture Notes in Computer Science)*, Vol. 8123. Springer, 44–58.

Samuel R. Buss. 1998. An introduction to proof theory. In *Handbook of Proof Theory*, Samuel R. Buss (Ed.). Vol. 137. Elsevier, 1–78.

Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. 2007. Logics of formal inconsistency. In *Handbook of Philosophical Logic*, Dov Gabbay and Franz Guenthner (Eds.). Vol. 14. Springer, 15–107.

Walter A. Carnielli and João Marcos. 2002. A taxonomy of C-systems. In *Paraconsistency: The Logical Way to the Inconsistent*, Walter A. Carnielli, Marcelo E. Coniglio, and Itala D'Ottaviano (Eds.). Number 228 in Lecture Notes in Pure and Applied Mathematics. Marcel Dekker, 1–94.

Agata Ciabattoni, Nikolaos Galatos, and Kazushige Terui. 2008. From Axioms to Analytic Rules in Nonclassical Logics. In *23rd Annual IEEE Symposium on Logic in Computer Science, 2008. LICS '08.* 229–240.

Agata Ciabattoni, Ori Lahav, Lara Spendier, and Anna Zamansky. 2013. Automated Support for the Investigation of Paraconsistent and Other Logics. In *Logical Foundations of Computer Science (Lecture Notes in Computer Science)*, Vol. 7734. Springer, 119–133.

Willem Conradie and Alessandra Palmigiano. 2012. Algorithmic correspondence and canonicity for distributive modal logic. *Annals of Pure and Applied Logic* 163, 3 (2012), 338–376.

Newton C. A. Da Costa. 1974. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic* 15 (1974), 497–510.

Norihiro Kamide. 2009. Proof Systems Combining Classical and Paraconsistent Negations. *Studia Logica* 91, 2 (2009), 217–238.

Norihiro Kamide. 2013. A hierarchy of weak double negations. *Studia Logica* 101, 6 (2013), 1277–1297.

Ori Lahav and Yoni Zohar. 2014. SAT-based Decision Procedure for Analytic Pure Sequent Calculi. In *Proceedings of the 7th International Joint Conference on Automated Reasoning, IJCAR 2014 (Lecture Notes in Computer Science)*. 76–90.

Björn Lellmann and Dirk Pattinson. 2011. Cut Elimination for Shallow Modal Logics. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods 2011 (Lecture Notes in Computer Science)*. 211–225.

Topias Nipkow, Lawrence C. Paulson and Markus Wenzel. 2002. Isabelle/HOL – A Proof Assistant for Higher-Order Logic. *Lecture Notes in Computer Science* . Vol. 2283.

Hans Jürgen Ohlbach. 1994. Computer Support for the Development and Investigation of Logics. *Journal of the IGPL* 4, 1 (1994), 109–127.

Markus Wenzel, Lawrence C. Paulson and Topias Nipkow. 2008. The Isabelle Framework. In *Proceedings of Theorem Proving in Higher Order Logics 2008 (Lecture Notes in Computer Science)*. 33–38.

Received Month Year; revised Month Year; accepted Month Year