Um Provador de Teoremas Multi-Estratégia
A Multi-Strategy Theorem Prover

Adolfo Gustavo Serra Seca Neto
DAINF - UTFPR
`http://www.dainf.ct.utfpr.edu.br/~adolfo`

Data desta versão: 14 de novembro de 2008
Date of this version: November 14th, 2008

Sobre

Este é um capítulo da minha tese de Doutorado intitulada "Um Provador de Teoremas Multi-Estratégia". Esta tese, na área de Ciência da Computação, foi defendida em 30 de janeiro de 2007 no Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP). Meu orientador foi o Prof. Dr. Marcelo Finger. O texto completo desta tese está disponível em
`http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04052007-175943/`

About

This is a chapter of my Ph.D. thesis entitled "A Multi-Strategy Theorem Prover". This Computer Science thesis was defended on January 30th, 2007 at the Institute of Mathematics and Statistics (IME) of the University of São Paulo (USP). My advisor was Prof. Dr. Marcelo Finger. Thesis full text is available at
`http://www.teses.usp.br/teses/disponiveis/45/45134/tde-04052007-175943/`.
Only the first chapter was written in Portuguese. All the following appendices were written in English.

Apêndice B

Tableaux for Classical and Paraconsistent Logics

The method of tableaux is a formal proof procedure existing in many varieties and for several logics [52]. It is a refutation procedure — that is, in order to prove that a formula X is valid we try to invalidate its negation. Besides that, tableau systems are *expansion systems*, i.e., they are systems that contain a finite set of expansion rules [29]. A *tableau* is a tree[1] [26] with one or more branches[2], whose nodes are lists of formulas. The proof search procedure of each specific tableau method describes how to construct a tableau proof (for a list of formulas that one wants to refute) using the expansion rules.

In this appendix we briefly present two tableau systems for classical propositional logic: analytic tableaux [106] and the **KE** inference system [29]. After that, we show **KE** systems we have developed for **mbC** and **mCi**, two paraconsistent logics, and prove some properties of these systems. We finish by discussing the complexity of some of the logical systems and proof methods presented.

[1]In fact, it is better to represent a tableau as a sequence of trees, as it is done in [17], to describe the history of the derivation.

[2]Some of the terms used in this appendix will become clear only in Section C.2.1 when we will describe the **KE** Proof Search Procedure.

B.1 Logical Systems

In this section we present some notions[3] about logical systems which will be used in the rest of this thesis. We assume familiarity with the syntax and semantics of *classical propositional logic*, from now on denoted by **CPL**.

The language of every logic **L** is defined over a propositional *signature* $\Sigma = \{\Sigma_n\}_{n\in\omega}$ such that Σ_n is the set of connectives of arity n. As can be seen in [19], 'ω' is the smallest infinite ordinal, which is the order type of the natural numbers; it can be identified with the set of natural numbers. Therefore, each Σ_n is at most enumerable. The cardinality of each Σ_n is less than or equal to $\aleph_0$ (the cardinal of the set of natural numbers $\mathbb{N}$ [79]); in fact, each Σ_n can be written as a family with indices in ω [74].

The set $\mathcal{P} = \{p_n : n \in \omega\}$ is the set of *propositional variables* (or *atomic formulas*) from which we freely generate the algebra *For* of formulas using Σ. From here on, Σ will denote the signature containing the binary connectives '$\wedge$', '$\vee$', '$\rightarrow$', and the unary connective '$\neg$'. By *For* we will denote the set of formulas freely generated by $\mathcal{P}$ over Σ.

In the same spirit, Σ° ($\Sigma^{\bullet}$) will denote the signature obtained by the addition of a new unary connective '$\circ$' ('$\bullet$') to the signature Σ, and For° ($For^{\bullet}$) will denote the algebra of formulas for the signature Σ° ($\Sigma^{\bullet}$). '$\circ$' and '$\bullet$' are called, respectively, the 'consistency' and 'inconsistency' connectives.

The other connectives' names are the following: '$\wedge$' is the 'and' connective, also called 'conjunction'; '$\vee$' is the 'or' connective, also called 'disjunction'; '$\rightarrow$' is the 'implies' connective, also called 'implication'; and '$\neg$' is the 'not' connective, also called 'negation'.

Given a formula A, the set of its *subformulas*, *sf*(A), is defined recursively in the following way:

- If p is a propositional atom, then $sf(p) = \{p\}$;
- If A is $\odot A_1$, where $\odot$ is a unary connective, then $sf(A) = \{A\} \cup sf(A_1)$;
- If A is $A_1 \oslash A_2$, where $\oslash$ is a binary connective, then $sf(A) = \{A\} \cup sf(A_1) \cup sf(A_2)$.

[3]Most of the notions presented here were taken from [18].

Let $\wp(X)$ be the powerset of a set X. As usual, given a set *For* of formulas, we say that $\vdash$ defines a *(Tarskian) consequence relation* on *For*, where $\vdash \subseteq \wp(For) \times For$, if the following clauses hold, for any formulas A and B, and subsets Γ and Δ of *For* (formulas and commas at the left-hand side of $\vdash$ denote, as usual, sets and unions of sets of formulas) [18]:

$A \in \Gamma$ implies $\Gamma \vdash A$ (reflexivity)

$(\Delta \vdash A$ and $\Delta \subseteq \Gamma)$ implies $\Gamma \vdash A$ (monotonicity)

$(\Delta \vdash A$ and $\Gamma, A \vdash B$ implies $\Gamma, \Delta \vdash B$ (cut)

A *(Tarskian) logic* **L** is a structure of the form $\langle For, \vdash \rangle$, containing a set of formulas and a consequence relation defined on this set [18]. A logical system is a pair $(\vdash, \mathrm{S}_{\vdash})$, where $\vdash$ is a consequence relation (a set of pairs) and $\mathrm{S}_{\vdash}$ is an algorithmic system for generating all the pairs in that relation [54]. For a given consequence relation, there can be many algorithmic systems. For **CPL**, for instance, we have axiomatic systems (see Section B.1.1), the Davis-Putnam procedure [33, 32], the Resolution method [100] and many others.

B.1.1 Classical Propositional Logic

According to [17], the *axiomatic method* is the oldest known proof method. An *axiomatic system* [78] is composed of a set of axioms and a set of inference rules. An *axiom* is a starting point for deducing logically valid propositions, a formula which is taken for granted as valid. And an *inference rule* is a relation between formulas. As stated in [78], for each inference rule $\mathcal{R}$, there is a unique positive integer j such that for every set of j formulas and each formula C, one can effectively decide whether the given j formulas and each C are related to C in $\mathcal{R}$, and, if so, C is said to *follow from* or to be a *direct consequence of* the given formulas by virtue of $\mathcal{R}$. In an inference rule, the j formulas are called *premises* and C is called the *conclusion* of the inference rule.

An *axiom schema* is a formula such that any formula obtained by one or more substitutions in it is taken to be an axiom [67]. Similarly, an *inference rule schema* is an inference rule where one or more substitutions can be performed on the formulas in the inference rule relation (see [17]).

The following definitions were taken from [78]: a *proof* in an axiomatic system AS is a sequence $F_1, \ldots, F_n$ of formulas such that, for each i, either F_i is an axiom of AS or F_i is a direct consequence of some of the preceding formulas in the sequence by virtue of one of the inference rules of AS. A *theorem* of AS is a formula F such that F is the last formula of some proof in AS. A formula C is said to be a *consequence* in AS of a set Γ of formulas if and only if there is a sequence $F_1, \ldots, F_k$ of formulas such that C is F_k and, for each i, either F_i is an axiom or F_i is in Γ, or F_i is a direct consequence by some inference rule of some of the preceding formulas in the sequence. Such a sequence is called a *proof* (or *deduction*) *of* C *from* Γ. The members of Γ are called the *hypotheses* or *premises* of the proof.

Given an axiomatic system for a logic **L,** we write $\Gamma \vdash_{\mathbf{L}} A$ to say that there is proof in **L** of A from the premises in Γ [18]. Axiomatization is the process of defining an axiomatic system for a given logical system. Classical propositional logic can be axiomatized in several ways. The following is an axiomatization for **CPL** [18] that contains axiom and inference rule schemas in the signature Σ:

Axiom schemas:

(Ax1) $A \to (B \to A)$

(Ax2) $(A \to B) \to ((A \to (B \to C)) \to (A \to C))$

(Ax3) $A \to (B \to (A \wedge B))$

(Ax4) $(A \wedge B) \to A$

(Ax5) $(A \wedge B) \to B$

(Ax6) $A \to (A \vee B)$

(Ax7) $B \to (A \vee B)$

(Ax8) $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

(Ax9) $A \vee (A \to B)$

(Ax10) $A \vee \neg A$

(exp) $A \to (\neg A \to B)$

Inference rule schema:

(MP) $\dfrac{A, A \to B}{B}$

Let $\mathbf{2} \overset{\text{def}}{=} \{0,1\}$ be the set of truth-values, where 1 denotes the 'true' value and 0 denotes the 'false' value. By defining a valuation, we can inductively define the truth-value of a formula from the truth-value of its propositional variables. Below we present a definition of a valuation for **CPL**:

Definition B.1.1. *[18] A* **CPL**-valuation *is any function* $v : \text{For} \longrightarrow \mathbf{2}$ *subject to the following clauses:*

(v1) $v(A \wedge B) = 1$ *iff* $v(A) = 1$ *and* $v(B) = 1$*;*

(v2) $v(A \vee B) = 1$ *iff* $v(A) = 1$ *or* $v(B) = 1$*;*

(v3) $v(A \rightarrow B) = 1$ *iff* $v(A) = 0$ *or* $v(B) = 1$*;*

(v4) $v(\neg A) = 1$ *iff* $v(A) = 0$*.*

A formula X is said to be *satisfiable* if truth-values can be assigned to its propositional variables in a way that makes the formula true, i.e. if there is at least one valuation such that $v(X) = 1$. A formula is a *tautology* if all possible valuations make the formula true. In **CPL**, for instance, $A \vee B$ is satisfiable, but it is not a tautology, while $A \vee \neg A$ is a tautology.

Let Γ be a set of formulas in *For*, and A a formula in *For*. We say that A is a semantical consequence of Γ (denoted by $\Gamma \models A$) if for any valuation v we have the following [17]:

$$\text{if } v(B) = 1 \text{ for all } B \text{ in } \Gamma, \text{ then } v(A) = 1.$$

The **CPL** axiomatization we have presented above is sound and complete with respect to the semantical consequence relation presented above. That is, for any Γ and A, $\Gamma \vdash_{\mathbf{CPL}} A$ implies $\Gamma \models_{\mathbf{CPL}} A$ (soundness [17]). And $\Gamma \models_{\mathbf{CPL}} A$ implies $\Gamma \vdash_{\mathbf{CPL}} A$ (strong completeness [17]).

B.1.2 Logics of Formal Inconsistency

Logics of Formal Inconsistency (**LFI**s) are a class of paraconsistent logics. Below we reproduce a short definition of paraconsistent logics taken from [42]:

> A theory T is said to be inconsistent (contradictory) if it has as theorems a formula and its negation; otherwise, T is consistent (non-contradictory). A theory T is said to be trivial if every formula of its language is a theorem; otherwise T is non-trivial.
>
> If a theory T has as its underlying logic classical logic, the deduction of a contradiction leads to its trivialization.
>
> A logic is paraconsistent if it can be used as the underlying logic to inconsistent but non-trivial theories, which we call paraconsistent theories.

Logics of Formal Inconsistency are paraconsistent logics that internalize the notions of consistency and inconsistency at the object-language level (read more about the foundations of **LFI**s in [14]). Below we present some definitions (taken from [18]) which are necessary to give a formal characterization of **LFI**s.

The *Principle of Explosion* states that a logic **L** is explosive when $\forall\Gamma\forall A\forall B(\Gamma, A, \neg A \vdash B)$. It is well known that **CPL** is explosive. To define a Gentle Principle of Explosion, we first have to define *Gently Explosive Theories*. Consider a (possibly empty) set $\bigcirc(A)$ of formulas which depends only on the formula A. This is the set of formulas that, along with A and $\neg A$, makes a given theory Γ explode. We will call a theory Γ *gently explosive* (*with respect to* $\bigcirc(A)$) if there are formulas A and B such that the following hold:

1. $\bigcirc(A), A \nvdash B$;
2. $\bigcirc(A), \neg A \nvdash B$; and
3. $\forall A\forall B(\Gamma, \bigcirc(A), A, \neg A \vdash B)$.

The *Gentle Principle of Explosion* states that a logic **L** is said to be *gently explosive* when there is a set $\bigcirc(A)$ such that all of the theories of **L** are gently explosive (with respect to $\bigcirc(A)$). Finally, a Logic of Formal Inconsistency is defined as any logic in which the Principle of Explosion does not hold, but the Principle of Gentle Explosion does.

B.1.3 mbC, A Fundamental LFI

The logic **mbC** is the weakest **LFI** based on classical logic [18]. Any **LFI** based on classical logic can be axiomatized starting from positive classical logic ($\mathbf{CPL}^+$), whose axiomatization is that of **CPL** without the **(exp)** axiom schema. **mbC** is the weakest of such logics because all other **LFI**s based on classical logic presented in [18] prove more theorems. **mbC** axiomatization is obtained from $\mathbf{CPL}^+$'s axiomatization, over the signature Σ°, by adding the following axiom schema:

(bc1) $\circ A \to (A \to (\neg A \to B))$

The following valuation for **mbC** was presented in [18]:

Definition B.1.2. *An* **mbC***-valuation is any function* $v : \mathrm{For}^\circ \longrightarrow \mathbf{2}$ *subject to* **(v1)-(v3)** *from Definition B.1.1 and the following clauses:*

(v4') $v(\neg A) = 0$ *implies* $v(A) = 1$*;*

(v5) $v(\circ A) = 1$ *implies* $v(A) = 0$ *or* $v(\neg A) = 0$.

The definition of satisfiability in Section B.1.1 also holds for **mbC**. For instance, $A \vee \neg A$ is a tautology in **CPL**, but not in **mbC**. The formula $\neg(A \wedge \neg A \wedge \circ A)$ is an **mbC**-tautology. The intended reading of $\circ A$ is 'A is consistent'. In **mbC**, $\circ A$ is logically independent from $\neg(A \wedge \neg A)$, that is, $\circ$ is a primitive unary connective, not an abbreviation depending on conjunction and negation, as it happens in da Costa's C_n hierarchy of paraconsistent logics [27].

If $\vdash_{\mathbf{mbC}}$ denotes the consequence relation of **mbC**, then we obtain, by **(MP)**:

$$\circ A, A, \neg A \vdash_{\mathbf{mbC}} B \tag{B.1}$$

The *Finite Gentle Principle of Explosion* says that a logic **L** will be said to be *finitely gently explosive* when there is a *finite* set $\bigcirc(A)$ such that all of the theories of **L** are *finitely* gently explosive (with respect to $\bigcirc(A)$). According to [18], Rule (B.1) can be read as saying that 'if A is consistent and contradictory, then it explodes', and amounts to a realization of the Finite Gentle Principle of Explosion.

B.1.4 The mCi Logic

The **mCi** logic is another **LFI** presented in [18]. The motivation for its development was to enrich **mbC** so as to be able to define an inconsistency connective by the direct use of the paraconsistent negation, that is, by setting $\bullet A \stackrel{\text{def}}{=} \neg \circ A$. In **mCi**, $\bullet A$ and $\neg \bullet A$ are logically indistinguishable from $\neg \circ A$ and $\circ A$, respectively.

The logic **mCi** is obtained from **mbC** by the addition of the following axiom schemas:

(ci) $\neg \circ A \rightarrow (A \wedge \neg A)$;

(cc)$_n$ $\circ \neg^n \circ A \quad (n \geq 0)$.

To the above axiomatization is added the definition of an inconsistency connective $\bullet$ by setting $\bullet A \stackrel{\text{def}}{=} \neg \circ A$.

It is easy to verify (and it was shown in [18]) that $A \wedge \neg A \vdash_{\text{mbC}} \neg \circ A$. The converse property does not hold in **mbC** and it was the first additional axiom **(ci)** added to obtain **mCi**. So $\neg \circ A$ and $(A \wedge \neg A)$ are equivalent in **mCi**. To make formulas of the form $\neg \circ A$ 'behave classically', and to obtain a logic that is controllably explosive in contact with formulas of the form $\neg^n \circ A$, where $\neg^0 A \stackrel{\text{def}}{=} A$ and $\neg^{n+1} A \stackrel{\text{def}}{=} \neg\neg^n A$, axioms **(cc)**$_n$ were added to obtain **mCi**. With these axioms added, any formula of the form $\neg^n \circ A$ 'behaves classically' and $\{\neg^n \circ A, \neg^{n+1} \circ A\}$ in an explosive theory in **mCi**. Much more about **mCi** can be found in [18].

The following valuation for **mCi** was presented in [18]:

Definition B.1.3. *An* **mCi**-*valuation is an* **mbC**-*valuation* $v : \text{For}^\circ \longrightarrow \mathbf{2}$ *(see Definition B.1.2) satisfying, additionally, the following clauses:*

(v6) $v(\neg \circ A) = 1$ *implies* $v(A) = 1$ *and* $v(\neg A) = 1$*;*

(v7.n) $v(\circ \neg^n \circ A) = 1$ *for* $n \geq 0$*.*

The definition of satisfiability in Section B.1.1 also holds for **mCi**. In **mCi**, $(\bullet A) \rightarrow (A \wedge \neg A)$, which is by definition equal to $(\neg \circ A) \rightarrow (A \wedge \neg A)$, is a tautology. In **mbC**, the second formula is not a tautology.

B.2 Tableau Systems

In this section we will present some tableau systems for the logical systems presented in Section B.1. We will discuss their origin and motivation, present their rules, as well as prove some properties. We will not discuss here some aspects of the tableau systems which are relevant only for implementation. These aspects will be discussed in Appendix C.

B.2.1 Analytic Tableaux for CPL

The *analytic tableau* (**AT**) method, also known as *semantic tableaux*, is surely the most studied tableau method. It was presented in [106] as "an extremely elegant and efficient proof procedure for propositional logic". According to [52], this method is a variant of Beth's *"semantic tableaux"* [5], and of Hintikka methods [60].

The **AT** for **CPL** is a sound and complete proof system for **CPL** [106]. Its expansion rules (for the connectives in Σ) are presented in Figure B.1. This is the signed formula version of the **AT** for **CPL** (an unsigned version is also presented in [106]). A *signed formula* is an expression $\mathcal{S}\,X$ where $\mathcal{S}$ is called the *sign* and X is a propositional *formula*. The symbols $\mathtt{T}$ and $\mathtt{F}$, respectively representing the 'true' and 'false' truth-values, can be used as signs. The *conjugate* of a signed formula $\mathtt{T}\,A$ ($\mathtt{F}\,A$) is $\mathtt{F}\,A$ ($\mathtt{T}\,A$). Following [29], we shall call *subformulas* of a signed formula $\mathcal{S}A$ (where $\mathcal{S} \in \{\mathtt{T}\,, \mathtt{F}\,\}$) all the formulas of the form $\mathtt{T}\,B$ or $\mathtt{F}\,B$ where B is a subformula of A.

B.2.2 A KE System for CPL

The **KE** inference system is a more recent tableau method [31]. It was developed by Marco Mondadori and discussed in detail in several works authored or co-authored by Marcello D'Agostino [28, 8, 29]. The **KE** system was presented as an improvement, in the computational efficiency sense, over Analytic Tableaux. It is a refutation system that, though close to the analytic tableau method, is not affected by the anomalies of cut-free systems [29].

The **KE** system for **CPL** is a refutation system that is sound and complete. The

$$\frac{\mathtt{T}\,A \to B}{\mathtt{F}\,A \quad | \quad \mathtt{T}\,B}\ (\mathtt{T} \to) \qquad \frac{\mathtt{F}\,A \to B}{\begin{array}{c}\mathtt{T}\,A \\ \mathtt{F}\,B\end{array}}\ (\mathtt{F}\to)$$

$$\frac{\mathtt{F}\,A \wedge B}{\mathtt{F}\,A \quad | \quad \mathtt{F}\,B}\ (\mathtt{F}\wedge) \qquad \frac{\mathtt{T}\,A \wedge B}{\begin{array}{c}\mathtt{T}\,A \\ \mathtt{T}\,B\end{array}}\ (\mathtt{T}\wedge)$$

$$\frac{\mathtt{T}\,A \vee B}{\mathtt{T}\,A \quad | \quad \mathtt{T}\,B}\ (\mathtt{T}\vee) \qquad \frac{\mathtt{F}\,A \vee B}{\begin{array}{c}\mathtt{F}\,A \\ \mathtt{F}\,B\end{array}}\ (\mathtt{F}\vee)$$

$$\frac{\mathtt{T}\,\neg A}{\mathtt{F}\,A}\ (\mathtt{T}\neg) \qquad \frac{\mathtt{F}\,\neg A}{\mathtt{T}\,A}\ (\mathtt{F}\neg)$$

Figure B.1: **CPL AT** rules.

first motivation for its development was to obtain a tableau method inline with classical principles. In [29], D'Agostino argues that analytic tableau rules for **CPL** are not really classical since one of the two principles that form the basis of the classical notion of truth, the Principle of Bivalence, is not clearly present in that tableau system. The Principle of Bivalence (also known as the Principle of the Excluded Middle) states that every proposition is either true or false, and there are no other possibilities. The other principle, the Principle of Non-contradiction, which asserts that no proposition can be true and false at the same time, is embodied in the definition of closed branches in analytic tableaus together with the rules for negation.

The second motivation for **KE** development was to design a computationally more efficient system. Analytic tableau refutations are intrinsically redundant because, depending on the problem being tackled, it is necessary to prove again the same subproblem one or more times [29]. This difficulty is not related to any intrinsic difficulty of the problem considered but only to the redundant behavior of analytic tableau rules. Analytic tableaux correspond to cut-free sequent calculus [56] while **KE** corresponds to sequent calculus with *Cut* rule. Several families of problems have only exponential size proofs in cut-free sequent calculus, but can have polynomial size proofs in sequent calculus with Cut. Therefore, the **KE** system solves this limitation of analytic tableaux by presenting a set of rules that does not have **AT** rules' redundant behavior.

So what exactly is the main difference between the **KE** system and **AT**? A **KE** system is a tableau system with *only one* branching rule, the Principle of Bivalence (PB) rule. And although this rule resembles Gentzen's sequent calculus [56] *Cut* rule, which is not essential for the sequent calculus proof system, (PB) is not eliminable from **KE**.

The **KE** expansion rules for **CPL** are presented in Figure B.2. Notice that some rules have two premises, some have one premise and the (PB) rule is a zero premise rule. We say that the rules with two premises have a *main premise* (the more complex) and an *auxiliary premise*. For rules with only one premise, this premise is also referred as main premise. A linear rule is a rule which does not force branching. All **KE** rules are linear, except the (PB) rule.

Some rules in Figure B.2 can be derived from others. For instance, if we have $(\mathtt{T}\vee_1)$ and (PB) we can derive $(\mathtt{T}\vee_2)$. It is clear that the opposite is true: if we have $(\mathtt{T}\vee_2)$ and (PB) we can derive $(\mathtt{T}\vee_1)$. This relationship also happens between $(\mathtt{F}\wedge_1)$ and $(\mathtt{F}\wedge_2)$, and between $(\mathtt{T}{\to}_1)$ and $(\mathtt{T}{\to}_2)$.

$$\frac{\begin{array}{c}\mathtt{T}\,A \to B\\ \mathtt{T}\,A\end{array}}{\mathtt{T}\,B}\ (\mathtt{T}{\to}_1) \qquad \frac{\begin{array}{c}\mathtt{T}\,A \to B\\ \mathtt{F}\,B\end{array}}{\mathtt{F}\,A}\ (\mathtt{T}{\to}_2) \qquad \frac{\mathtt{F}\,A \to B}{\begin{array}{c}\mathtt{T}\,A\\ \mathtt{F}\,B\end{array}}\ (\mathtt{F}{\to})$$

$$\frac{\begin{array}{c}\mathtt{F}\,A \wedge B\\ \mathtt{T}\,A\end{array}}{\mathtt{F}\,B}\ (\mathtt{F}\wedge_1) \qquad \frac{\begin{array}{c}\mathtt{F}\,A \wedge B\\ \mathtt{T}\,B\end{array}}{\mathtt{F}\,A}\ (\mathtt{F}\wedge_2) \qquad \frac{\mathtt{T}\,A \wedge B}{\begin{array}{c}\mathtt{T}\,A\\ \mathtt{T}\,B\end{array}}\ (\mathtt{T}\wedge)$$

$$\frac{\begin{array}{c}\mathtt{T}\,A \vee B\\ \mathtt{F}\,A\end{array}}{\mathtt{T}\,B}\ (\mathtt{T}\vee_1) \qquad \frac{\begin{array}{c}\mathtt{T}\,A \vee B\\ \mathtt{F}\,B\end{array}}{\mathtt{T}\,A}\ (\mathtt{T}\vee_2) \qquad \frac{\mathtt{F}\,A \vee B}{\begin{array}{c}\mathtt{F}\,A\\ \mathtt{F}\,B\end{array}}\ (\mathtt{F}\vee)$$

$$\frac{\mathtt{T}\,\neg A}{\mathtt{F}\,A}\ (\mathtt{T}\neg) \qquad \frac{\mathtt{F}\,\neg A}{\mathtt{T}\,A}\ (\mathtt{F}\neg)$$

$$\frac{}{\mathtt{T}\,A \quad | \quad \mathtt{F}\,A}\ (\mathrm{PB})$$

Figure B.2: **CPL KE** rules.

B.2.3 A KE System for mbC

In [11], Caleiro et alli exhibit a way of effectively constructing the two-valued semantics of any logic that has a truth-functional finite-valued semantics and a sufficiently expressive language. From there, one can provide those logics with adequate canonical systems of sequents or tableaux. The method permits one to obtain a complete tableau system for any propositional logic which has a complete semantics given through the so-called 'dyadic valuations'. In [18], sound and complete tableau systems for **mbC** and **mCi** obtained by using this general method are presented[4]. Let us call these systems **C^{3}M** tableau systems.

The **C^{3}M** tableau system rules for **mbC** are shown in Figure B.3. It is easy to notice that the rules for the binary connectives in Σ are the same as that from **AT** (see Figure B.1). It also has **AT** (F$\neg$) rule but does not have **AT** (T$\neg$) rule. To compensate for this, it has two additional rules: a branching rule similar to **KE** (PB) rule, and a (T$\circ$) rule. In total, this tableau system has 5 branching rules.

$$\frac{\mathrm{T}\,A \to B}{\mathrm{F}\,A \quad | \quad \mathrm{T}\,B}\ (\mathrm{T} \to) \qquad \frac{\mathrm{F}\,A \to B}{\begin{array}{c}\mathrm{T}\,A\\ \mathrm{F}\,B\end{array}}\ (\mathrm{F}\to)$$

$$\frac{\mathrm{F}\,A \wedge B}{\mathrm{F}\,A \quad | \quad \mathrm{F}\,B}\ (\mathrm{F}\wedge) \qquad \frac{\mathrm{T}\,A \wedge B}{\begin{array}{c}\mathrm{T}\,A\\ \mathrm{T}\,B\end{array}}\ (\mathrm{T}\wedge)$$

$$\frac{\mathrm{T}\,A \vee B}{\mathrm{T}\,A \quad | \quad \mathrm{T}\,B}\ (\mathrm{T}\vee) \qquad \frac{\mathrm{F}\,A \vee B}{\begin{array}{c}\mathrm{F}\,A\\ \mathrm{F}\,B\end{array}}\ (\mathrm{F}\vee)$$

$$\boxed{\frac{\mathrm{T}\ \circ A}{\mathrm{F}\,A \quad | \quad \mathrm{F}\,\neg A}\ (\mathrm{T}\circ)} \qquad \frac{\mathrm{F}\,\neg A}{\mathrm{T}\,A}\ (\mathrm{F}\neg)$$

$$\boxed{\frac{}{\mathrm{T}\,A \quad | \quad \mathrm{F}\,A}\ (\mathrm{PB})}$$

Figure B.3: **mbC** **C^{3}M** tableau rules.

As explained in [29], branching rules lead to inefficiency. To obtain a more efficient proof system, we used the **C^{3}M** tableau system for **mbC** as a basis to devise an original

[4]Some tableau systems for logics of formal inconsistency had already been presented in [15].

mbC KE system. The rules are presented in Figure B.4. The only difference between this system and the **KE** system for **CPL** is the replacement of the **CPL KE** ($\mathtt{T}\neg$) rule by the **KE** ($\mathtt{T}\neg'$) rule. Notice that the **KE** ($\mathtt{T}\neg'$) rule is a **LFI** version of **CPL KE** ($\mathtt{T}\neg$). It states clearly that besides $\mathtt{T}\,\neg A$, we need to have $\mathtt{T}\ \circ A$ to obtain $\mathtt{F}\,A$.

$$
\begin{array}{ccc}
\begin{array}{c} \mathtt{T}\,A \to B \\ \mathtt{T}\,A \\ \hline \mathtt{T}\,B \end{array}\ (\mathtt{T}{\to}_1) &
\begin{array}{c} \mathtt{T}\,A \to B \\ \mathtt{F}\,B \\ \hline \mathtt{F}\,A \end{array}\ (\mathtt{T}{\to}_2) &
\begin{array}{c} \mathtt{F}\,A \to B \\ \hline \mathtt{T}\,A \\ \mathtt{F}\,B \end{array}\ (\mathtt{F}{\to}) \\
\\
\begin{array}{c} \mathtt{F}\,A \wedge B \\ \mathtt{T}\,A \\ \hline \mathtt{F}\,B \end{array}\ (\mathtt{F}\wedge_1) &
\begin{array}{c} \mathtt{F}\,A \wedge B \\ \mathtt{T}\,B \\ \hline \mathtt{F}\,A \end{array}\ (\mathtt{F}\wedge_2) &
\begin{array}{c} \mathtt{T}\,A \wedge B \\ \hline \mathtt{T}\,A \\ \mathtt{T}\,B \end{array}\ (\mathtt{T}\wedge) \\
\\
\begin{array}{c} \mathtt{T}\,A \vee B \\ \mathtt{F}\,A \\ \hline \mathtt{T}\,B \end{array}\ (\mathtt{T}\vee_1) &
\begin{array}{c} \mathtt{T}\,A \vee B \\ \mathtt{F}\,B \\ \hline \mathtt{T}\,A \end{array}\ (\mathtt{T}\vee_2) &
\begin{array}{c} \mathtt{F}\,A \vee B \\ \hline \mathtt{F}\,A \\ \mathtt{F}\,B \end{array}\ (\mathtt{F}\vee) \\
\\
\boxed{\begin{array}{c} \mathtt{T}\,\neg A \\ \mathtt{T}\ \circ A \\ \hline \mathtt{F}\,A \end{array}\ (\mathtt{T}\neg')} &
\begin{array}{c} \mathtt{F}\,\neg A \\ \hline \mathtt{T}\,A \end{array}\ (\mathtt{F}\neg) & \\
\\
\begin{array}{c} \hline \mathtt{T}\,A \quad | \quad \mathtt{F}\,A \end{array}\ (\mathrm{PB}) & &
\end{array}
$$

Figure B.4: **mbC KE** rules.

Example B.2.1. In Figure B.5 we show a proof of $\circ A, \circ C, A \to \circ B, B \to C, (\neg B) \to (D \to \neg A) \vdash \neg(A \wedge \neg C \wedge D)$. Notice that to obtain 17 from 16 we used the optional rule $\mathtt{F}_{\text{for}}$ (a derived rule presented in Section C.2.4).

Analyticity, Correctness and Completeness Proof

A tableau proof enjoys the *subformula property* if every signed formula in the proof tree is a subformula of some formula in the list of signed formulas to be proved. Let us call *analytic* the applications of (PB) which preserve the subformula property. And the *analytic restriction* of a tableau system is the system obtained by restricting (PB) to analytic applications.

$$\begin{array}{ll}
1 & \mathtt{T}\ \circ A \\
2 & \mathtt{T}\ \circ C \\
3 & \mathtt{T}\ A \to \circ B \\
4 & \mathtt{T}\ B \to C \\
5 & \mathtt{T}\ (\neg B) \to (D \to \neg A) \\
6 & \mathtt{F}\ \neg(A \wedge \neg C \wedge D) \\
\hline
7 & \mathtt{T}\ A \wedge \neg C \wedge D \\
8 & \mathtt{T}\ A \\
9 & \mathtt{T}\ \neg C \wedge D \\
10 & \mathtt{T}\ \neg C \\
11 & \mathtt{T}\ D \\
12 & \mathtt{T}\ \circ B
\end{array}$$

$$\begin{array}{ll@{\qquad}ll}
13 & \mathtt{T}\,B & 16 & \mathtt{F}\,B \\
14 & \mathtt{T}\,C & 17 & \mathtt{T}\,\neg B \\
15 & \mathtt{F}\,C & 18 & \mathtt{T}\,D \to \neg A \\
 & \times & 19 & \mathtt{T}\,\neg A \\
 & & 20 & \mathtt{F}\,A \\
 & & & \times
\end{array}$$

Figure B.5: An **mbC KE** proof.

Given a rule R of an expansion system **S,** we say that an application of R to a branch θ is *analytic* when it has the *subformula property,* i.e. if all the new signed formulas appended to the end of θ are subformulas of signed formulas occurring in θ. According to [29], a *rule* R is *analytic* if every application of it is analytic. When the analytic restriction of a tableau system is sound and complete we say that this system is analytic (and that we have proved the system's analyticity).

Our intention here is to prove that the **mbC KE** system is analytic, sound and complete. As we have seen in Section B.2.3, the **mbC KE** system originated from the **CPL KE** system and the **C^{3}M** tableau system. It is easy to notice that all **CPL KE** rules, except (PB), are analytic. And although (PB) is not analytic, the **KE** system for **CPL** was proven to be analytic, sound and complete [29]. On the other hand, the **C^{3}M** tableau system for **mbC** has two non-analytic rules: (PB) and ($\mathtt{T}\circ$). It is sound and complete [18, 11] but there is no proof either that it is analytic or not analytic.

It is easy to show a procedure that transforms any proof in the **C^{3}M** tableau system for **mbC** in an **mbC KE** proof, thus proving that **mbC KE** system is also sound and complete. We will not do this here. Instead, we will prove directly that the **mbC KE** system is sound, complete and also analytic.

Proving that analytic restriction of **mbC KE** is sound and complete is a little bit more difficult than proving that **CPL KE** is analytic, because **mbC KE** has a two-premise rule, the $(\mathtt{T}\neg')$ rule, where neither premise is a subformula of the other premise, a condition satisfied by all **CPL KE** two-premise rules.

Because of this feature of the $(\mathtt{T}\neg')$ rule, it could be necessary to have non-analytic applications of the (PB) rule. But that is not the case: when performing an **mbC KE** proof we can restrict ourselves to analytic applications of (PB), applications which do not violate the subformula property, without affecting completeness. In this way, we demonstrate that even the analytic restriction of **mbC KE** is sound and complete.

The proof will be as follows. First we will define the notion of downward saturatedness for **mbC**. Then we will prove that every downward saturated set is satisfiable. The **mbC KE** proof search procedure for a set of signed formulas S either provides one or more downward saturated sets that give a valuation satisfying S or finishes with no downward saturated set.

Therefore, if an **mbC KE** tableau for a set of formulas S closes, then there is no downward saturated set that includes it, so S is unsatisfiable. However, if the tableau is open and completed, then any of its open branches can be represented as a downward saturated set and be used to provide a valuation that satisfies S. By construction, downward saturated sets for open branches are analytic, i.e. include only subformulas of S. We then conclude that the **mbC KE** system is analytic. As a corollary, it is also sound and complete.

Note: some concepts used in this proof are defined in Section C.2.1.

Definition B.2.1. A set of **mbC** signed formulas DS is *downward saturated* if

1. whenever a signed formula is in DS, its conjugate[5] is *not* in DS;

2. when all premises of any **mbC KE** rule (except (PB)) are in DS, its conclusions are also in DS;

[5]For any formula A The conjugate of $\mathtt{T}\ A$ is $\mathtt{F}\ A$, and vice-versa.

3. when the major premise of a two-premise **mbC KE** rule is in DS, either its auxiliary premise or its conjugate is in DS. For **mbC KE**, this item is valid for every rule except $(\mathtt{T}\neg')$. In this case, if $\mathtt{T}\,\neg X$ (which we define as the major premise in $(\mathtt{T}\neg')$) is in DS, either $\mathtt{T}\,\circ X$ or $\mathtt{F}\,\circ X$ can be in DS, but only if $\circ X$ is a subformula of some other formula in DS. If $\circ X$ is *not* a subformula of some other formula in DS, neither $\mathtt{T}\,\circ X$ nor $\mathtt{F}\,\circ X$ are in DS.

We can extend valuations to signed formulas in an obvious way: $v(\mathtt{T}\,A) = v(A)$ and $v(\mathtt{F}\,A) = 1 - v(A)$. A set of signed formulas L is satisfiable if it is not empty and there is a valuation such that for every formula $\mathcal{S}X \in L$, $v(\mathcal{S}X) = 1$. Otherwise, it is unsatisfiable.

Lemma B.2.2. *(Hintikka's Lemma for* **mbC***) Every* **mbC** *downward saturated set is satisfiable.*

Proof. For any downward saturated set DS, we can easily construct an **mbC** valuation v such that for every signed formula $\mathcal{S}X$ in the set, $v(\mathcal{S}X) = 1$. How can we guarantee this is in fact an **mbC** valuation? First, we know that there is no pair $\mathtt{T}\,X$ and $\mathtt{F}\,X$ in DS. Second, all premised **mbC KE** rules preserve **mbC** valuations. That is, if $v(\mathcal{S}X_i) = 1$ for every premise $\mathcal{S}X_i$, then $v(\mathcal{S}C_j) = 1$ for all conclusions C_j. And if $v(\mathcal{S}X_1) = 1$ and $v(\mathcal{S}X_2) = 0$, where X_1 and X_2 are, respectively, major and minor premises of an **mbC KE** rule, then $v(\mathcal{S}'X_2) = 1$, where $\mathcal{S}'X_2$ is the conjugate of $\mathcal{S}X_2$. For instance, suppose $\mathtt{T}\,A \wedge B \in DS$, then $v(\mathtt{T}\,A \wedge B) = 1$. In accordance with the definition of downward saturated sets, $\{\mathtt{T}\,A, \mathtt{T}\,B\} \subseteq DS$. And by the definition of **mbC** valuation, $v(\mathtt{T}\,A \wedge B) = 1$ implies $v(\mathtt{T}\,A) = v(\mathtt{T}\,B) = 1$. □

Theorem B.2.3. *Let DS' be a set of signed formulas. DS' is satisfiable* if and only if *there exists a downward saturated set DS" such that DS'* $\subseteq$ *DS".*

Proof. $(\Leftarrow)$ First, let us prove that if there exists a downward saturated set DS" such that DS' $\subseteq$ DS", then DS' is satisfiable. This is obvious because from DS" we can obtain a valuation that satisfies all formulas in DS", and DS' $\subseteq$ DS".

$(\Rightarrow)$ Now, let us prove that if DS' is satisfiable, there exists a downward saturated set

DS" such that DS' $\subseteq$ DS".

So, suppose that DS' is satisfiable and that there is no downward saturated set DS" such that DS" $\subseteq$ DS'. Using items (2) and (3) of (B.2.1), we can obtain a family of sets of signed formulas $DS\text{'}_i$ $(i \geq 1)$ that include DS'. If none of them is downward saturated, it is because for all i, $\{\mathtt{T}\,X, \mathtt{F}\,X\} \in DS\text{'}_i$ for some X. But all rules are valuation-preserving, so this can only happen if DS is unsatisfiable, which is a contradiction. □

Corollary B.2.4. *DS' is an unsatisfiable set of formulas if and only if there is no downward saturated set DS" such that DS'* $\subseteq$ *DS".*

Theorem B.2.5. *The* **mbC KE** *system is analytic.*

Proof. The **mbC KE** proof search procedure for a set of signed formulas S either provides one or more downward saturated sets that give a valuation satisfying S or finishes with no downward saturated set. If an **mbC KE** tableau for a set of formulas S closes, then there is no downward saturated set that includes it, so S is unsatisfiable. If the tableau is open and completed, then any of its open branches can be represented as a downward saturated set and be used to provide a valuation that satisfies S. By construction, downward saturated sets for open branches are analytic, i.e. include only subformulas of S. Therefore, the **mbC KE** system is analytic. □

Corollary B.2.6. *The* **mbC KE** *system is sound and complete.*

Proof. The **mbC KE** system is a refutation system, as most tableau systems. The **mbC KE** system is sound because if an **mbC KE** tableau for a set of formulas S closes, then S is unsatisfiable. And if the tableau is open and completed, S is satisfiable. This has been shown in the proof of the theorem above. It is complete because if S is satisfiable, no **mbC KE** tableau for a set of formulas S closes. And if S is unsatisfiable, all completed **mbC KE** tableau for S close. □

B.2.4 A KE System for mCi

A tableau system for **mCi**, the **C^{3}M** system for **mCi**, was presented in [18] as an extension of the **C^{3}M mbC** system. Its rules are shown in Figure B.6. This system has a new rule called (T$\neg\circ$) that corresponds to the axiom **(cc)** (see **mCi** axiomatization in Section B.1.4) and rules (T $\circ\neg^{n}\circ$), for $n \geq 0$, that correspond to axioms **(cc)**$_n$.

$$\frac{\mathrm{T}\, A \to B}{\mathrm{F}\, A \quad \mid \quad \mathrm{T}\, B}\ (\mathrm{T} \to) \qquad \frac{\mathrm{F}\, A \to B}{\begin{array}{c}\mathrm{T}\, A \\ \mathrm{F}\, B\end{array}}\ (\mathrm{F}\to)$$

$$\frac{\mathrm{F}\, A \wedge B}{\mathrm{F}\, A \quad \mid \quad \mathrm{F}\, B}\ (\mathrm{F}\wedge) \qquad \frac{\mathrm{T}\, A \wedge B}{\begin{array}{c}\mathrm{T}\, A \\ \mathrm{T}\, B\end{array}}\ (\mathrm{T}\wedge)$$

$$\frac{\mathrm{T}\, A \vee B}{\mathrm{T}\, A \quad \mid \quad \mathrm{T}\, B}\ (\mathrm{T}\vee) \qquad \frac{\mathrm{F}\, A \vee B}{\begin{array}{c}\mathrm{F}\, A \\ \mathrm{F}\, B\end{array}}\ (\mathrm{F}\vee)$$

$$\frac{\mathrm{T} \circ A}{\mathrm{F}\, A \quad \mid \quad \mathrm{F} \neg A}\ (\mathrm{T}\circ) \qquad \frac{\mathrm{F} \neg A}{\mathrm{T}\, A}\ (\mathrm{F}\neg)$$

$$\boxed{\frac{\mathrm{T} \neg (\circ A)}{\begin{array}{c}\mathrm{T}\, A \\ \mathrm{T} \neg A\end{array}}\ (\mathrm{T}\neg\circ)} \qquad \boxed{\frac{}{\mathrm{T} \circ (\neg^{n}(\circ A))}\ \text{for } {\scriptstyle (n \geq 0)}\ (\mathrm{T} \circ \neg^{n}\circ)}$$

$$\frac{}{\mathrm{T}\, A \quad \mid \quad \mathrm{F}\, A}\ (\mathrm{PB})$$

Figure B.6: **mCi C^{3}M** tableau rules.

As we have done for **mbC**, we use the **C^{3}M** tableau system for **mCi** as a basis to devise an original **mCi KE** system. **mCi KE** rules are presented in Figure B.7. We can see this system as an extension of the **mbC KE** system, where we include the (T$\neg\circ$) rule and the new (F $\circ\neg^{n}\circ$) rules, for $n \geq 0$. These (F $\circ\neg^{n}\circ$) rules were motivated by the same axioms that motivated (T $\circ\neg^{n}\circ$) rules, with the advantage of being analytic. The (T$\neg\circ$) rule, however, is not analytic. So neither **C^{3}M** for **mCi** nor **mCi-KE** are analytic proof systems.

Example B.2.2. In Figure B.8 we show an example of an **mCi KE** non analytic proof:

$$\frac{\begin{array}{c}\mathtt{T}\,A \to B\\ \mathtt{T}\,A\end{array}}{\mathtt{T}\,B}\ (\mathtt{T}{\to}_1) \qquad \frac{\begin{array}{c}\mathtt{T}\,A \to B\\ \mathtt{F}\,B\end{array}}{\mathtt{F}\,A}\ (\mathtt{T}{\to}_2) \qquad \frac{\mathtt{F}\,A \to B}{\begin{array}{c}\mathtt{T}\,A\\ \mathtt{F}\,B\end{array}}\ (\mathtt{F}{\to})$$

$$\frac{\begin{array}{c}\mathtt{F}\,A \wedge B\\ \mathtt{T}\,A\end{array}}{\mathtt{F}\,B}\ (\mathtt{F}\wedge_1) \qquad \frac{\begin{array}{c}\mathtt{F}\,A \wedge B\\ \mathtt{T}\,B\end{array}}{\mathtt{F}\,A}\ (\mathtt{F}\wedge_2) \qquad \frac{\mathtt{T}\,A \wedge B}{\begin{array}{c}\mathtt{T}\,A\\ \mathtt{T}\,B\end{array}}\ (\mathtt{T}\wedge)$$

$$\frac{\begin{array}{c}\mathtt{T}\,A \vee B\\ \mathtt{F}\,A\end{array}}{\mathtt{T}\,B}\ (\mathtt{T}\vee_1) \qquad \frac{\begin{array}{c}\mathtt{T}\,A \vee B\\ \mathtt{F}\,B\end{array}}{\mathtt{T}\,A}\ (\mathtt{T}\vee_2) \qquad \frac{\mathtt{F}\,A \vee B}{\begin{array}{c}\mathtt{F}\,A\\ \mathtt{F}\,B\end{array}}\ (\mathtt{F}\vee)$$

$$\frac{\begin{array}{c}\mathtt{T}\,\neg A\\ \mathtt{T}\,\circ A\end{array}}{\mathtt{F}\,A}\ (\mathtt{T}\neg') \qquad \frac{\mathtt{F}\,\neg A}{\mathtt{T}\,A}\ (\mathtt{F}\neg)$$

$$\boxed{\frac{\mathtt{T}\,\neg(\circ A)}{\begin{array}{c}\mathtt{T}\,A\\ \mathtt{T}\,\neg A\end{array}}\ (\mathtt{T}\neg\circ)} \qquad \boxed{\frac{\mathtt{F}\,\circ(\neg^n(\circ A))}{\times}\ \text{for }_{(n\geq 0)}\ (\mathtt{F}\circ\neg^n\circ)}$$

$$\frac{}{\mathtt{T}\,A \quad | \quad \mathtt{F}\,A}\ (\mathrm{PB})$$

Figure B.7: **mCi KE** rules.

a proof of $\circ A \vdash \neg\neg\circ A$. The formula number 5 is not a subformula of any formula in the sequent being proved.

Example B.2.3. In Figure B.9 we show another example of a non analytic proof of $\neg\neg\circ A \vdash \circ A$. In fact, it is not possible to prove this sequent in **mCi KE** without a non analytic application of the (PB) rule.

$$\begin{array}{cc}
1 & \mathtt{T}\ \circ A \\
2 & \mathtt{F}\ \neg\neg\circ A \\
\hline
3 & \mathtt{T}\ \neg\circ A \\
4 & \mathtt{T}\ A \\
5 & \mathtt{T}\ \neg A \\
6 & \mathtt{F}\ A \\
 & \times
\end{array}$$

Figure B.8: An **mCi KE** proof of $\circ A \vdash \neg\neg\circ A$.

$$\begin{array}{ccc}
 & \mathtt{T}\ \neg\neg\circ A & \\
 & \mathtt{F}\ \circ A & \\
\mathtt{T}\ \circ\neg\circ A & & \mathtt{F}\ \circ\neg\circ A \\
\mathtt{F}\ \neg\circ A & & \times \\
\mathtt{T}\ \circ A & & \\
\times & &
\end{array}$$

Figure B.9: An **mCi KE** proof of $\neg\neg\circ A \vdash \circ A$.

Correctness and Completeness Proof

Our intention here is to prove that the **mCi KE** system is sound and complete. It seems clear to us that the **mCi KE** system is not analytic, because of its ($\mathtt{T}\neg\circ$) rule, which is not analytic. We will follow the same schema used in Section B.2.3.

Definition B.2.7. A set of **mCi** signed formulas DS is *downward saturated*:

1. whenever a signed formula is in DS, its conjugate is *not* in DS;
2. when all premises of any **mCi KE** rule (except (PB) and ($\mathtt{F}\circ\neg^n\circ$), for $n \geq 0$) are in DS, its conclusions are also in DS;

3. when the major premise of a two-premise **mCi KE** rule is in DS, either its auxiliary premise or its conjugate is in DS. And the same condition for the ($\mathtt{T}\neg'$) rule that holds in Definition B.2.1 also holds here: if $\mathtt{T}\,\neg X$ is in DS, either $\mathtt{T}\,\circ X$ or $\mathtt{F}\,\circ X$ can be in DS, but only if $\circ X$ is a subformula of some other formula in DS. If $\circ X$ is *not* a subformula of some other formula in DS, neither $\mathtt{T}\,\circ X$ nor $\mathtt{F}\,\circ X$ are in DS;

4. if a signed formula $\mathcal{S}\ X$ is in DS, then for any sign $\mathcal{S}$, for any formula X, for all subformulas Y of X and for all $n \geq 0$, the signed formula $\mathtt{T}\,\circ \neg^n \circ Y$ is in DS.

The Hintikka's Lemma also holds for **mCi** downward saturated sets:

Lemma B.2.8. *(Hintikka's Lemma for* **mCi***) Every* **mCi** *downward saturated set is satisfiable.*

Proof. For any downward saturated set DS, we can easily construct an **mCi** valuation v such that for every signed formula $\mathcal{S}X$ in the set, $v(\mathcal{S}X) = 1$. How can we guarantee this is in fact a valuation? First, we know that there is no pair $\mathtt{T}\,X$ and $\mathtt{F}\,X$ in DS. Second, all premised **mCi KE** rules (except ($\mathtt{F}\circ\neg^n\circ$) rules) preserve valuations. Note that ($\mathtt{F}\circ\neg^n\circ$) rules are taken into account by the last clause in Definition B.2.7. That is, if we have a set of signed formulas that contains $\mathtt{F}\,\circ\neg^n\circ X$, every downward saturated set that contains this set should also contain $\mathtt{T}\,\circ\neg^n\circ X$. Therefore it is not downward saturated. To be downward saturated a set DS must contain, for all its subformulas[6] X, $\mathtt{T}\,\circ\neg^n\circ X$ (and must not contain any $\mathtt{F}\,\circ\neg^n\circ X$). As we can see in clause **(v7.n)** of the **mCi** valuation definition (see Definition B.1.3), $v(\mathtt{T}\,\circ\neg^n\circ X) = 1$ for all X. Therefore, DS is satisifable. □

Theorem B.2.3 and Corollary B.2.4 also hold for **mCi** downward saturated sets.

Theorem B.2.9. *The* **mCi KE** *system is sound and complete.*

Proof. The **mCi KE** proof search procedure for a set of signed formulas S either provides

[6]To be precise, by the subformulas of a set of signed formulas $\{\mathcal{S}_iF_i\}$, where $\mathcal{S}_i$ is a sign and F_i is an unsigned formula, we mean the set of subformulas of $\{F_i\}$.

one or more downward saturated sets that give a valuation satisfying S or finishes with no downward saturated set. The **mCi KE** system is a refutation system. The **mCi KE** system is sound because if an **mCi KE** tableau for a set of formulas S closes, then there is no downward saturated set that includes it, so S is unsatisfiable. If the tableau is open and completed, then any of its open branches can be represented as a downward saturated set and be used to provide a valuation that satisfies S (in other words, S is satisfiable).

The **mCi KE** system is complete because if S is satisfiable, no **mCi KE** tableau for a set of formulas S closes. And if S is unsatisfiable, all completed **mCi KE** tableau for S close. □

B.3 Complexity of Logical Systems

In this section, we are going to discuss some issues related to the complexity of logical systems. We begin with the complexity of decision problems.

B.3.1 Complexity of Decision Problems

The **CPL** satisfiability problem (known as 'SAT') is a decision problem studied in complexity theory. A decision problem is a problem that can be answered by 'yes' or 'no'. SAT can be described as "given a propositional formula, decide whether or not it is satisfiable". Many other decision problems, such as graph coloring problems, planning problems, and scheduling problems can be encoded into SAT.

SAT was the first known NP-complete problem. The class of NP-complete problems is a subclass of NP. While P is the class of decision problems that can be solved in polynomial time by a *deterministic* algorithm, NP is the class of decision problems that can be solved in polynomial time by a *nondeterministic* algorithm. Therefore P $\subseteq$ NP. The problems in NP are such that positive solutions can be verified in polynomial time. NP-complete problems are the most difficult problems in NP, the ones most likely not to be in P. If we find a polynomial time algorithm for any NP-complete problem, we can solve all problems in NP in polynomial time, because there is a polynomial time reduction from any NP problem into any NP-complete problem.

The *complement* of a decision problem is the decision problem resulting from reversing the 'yes' and 'no' answers. We can generalize this to the complement of a complexity class, called the complement class, which is the set of complements of every problem in the class. co-NP is the complement of the complexity class NP. It is the class of problems for which a 'no' answer can be verified in polynomial time. And co-NP-complete is the complement of the class of NP-complete problems.

mCi is co-NP-complete

The **CPL** decision problem ("given a propositional formula, decide whether or not it is a *tautology*") is co-NP-complete, because a formula in **CPL** is a tautology if and only if its negation is unsatisfiable. In [18], it was shown that the decision problem for **mbC** is also co-NP-complete.

As the **mbC** decision problem is co-NP-complete and **mCi** extends **mbC**, the **mCi** decision problem is co-NP-hard. To prove that the **mCi** decision problem is also co-NP-complete (as suggested in [18]) we need a NP algorithm for the complement of **mCi** decision: the falsification of a formula. That is, we must show that given a formula A and a **mCi**-valuation v it is possible to verify if $v(A) = 0$ in polynomial time.

Let A be an **mCi** formula. We show below how to construct an **mCi**-valuation v for A. This is here only to show that it is more difficult to build an **mCi**-valuation than a **CPL**-valuation.

Let $\mathrm{SSF}(A)$ be the set of all strict subformulas of A. A *strict subformula* of A is any subformula of A except A itself. Then we construct a new set $\mathrm{ESSF}(A)$, such that for all $X \in \mathrm{SSF}(A)$, X, $\circ X$ and $\neg X$ belong to $\mathrm{ESSF}(A)$.

If n is the size of A, then the size of $\mathrm{ESSF}(A)$ is at most $3(n-1)$. To build a valuation v for A we must, for any $X \in \mathrm{ESSF}(A)$, set $v(X)$ either to 0 or to 1, obbeying the **mCi**-valuation clauses presented in Definition B.1.3.

Up to now, we have only $v(X)$ for all $X \in \mathrm{ESSF}(A)$ (not necessarily a value for $v(A)$). The following algorithm allows us to find a value for $v(A)$:

1. if, for some X, A is $\circ X$, then:
   (a) if $v(\neg X) = 0$, then $v(X) = 1$ and $v(A)$ can be set either to 0 or to 1;
   (b) if $v(X) = 1$ and $v(\neg X) = 1$, then $v(A) = 0$;
   (c) if $v(X) = 0$ and $v(\neg X) = 1$, then $v(A)$ can be set either to 0 or to 1;
2. if, for some X, A is $\neg X$, then:
   (a) if $v(X) = 1$ then:
      i. if $v(\circ X) = 1$ then $v(A) = 0$;
      ii. if $v(\circ X) = 0$ then $v(A)$ can be set either to 0 or to 1;
   (b) if $v(X) = 0$ then $v(A) = 1$;
3. if, for some X, Y, A is $X \wedge Y$, then:
   (a) if $v(X) = 1$ and $v(Y) = 1$, then $v(A) = 1$;
   (b) otherwise, $v(A) = 0$;
4. if, for some X, Y, A is $X \vee Y$, then:
   (a) if $v(X) = 0$ and $v(Y) = 0$, then $v(A) = 0$;
   (b) otherwise, $v(A) = 1$;
5. if, for some X, Y, A is $X \rightarrow Y$, then:
   (a) if $v(X) = 0$ and $v(Y) = 1$, then $v(A) = 0$;
   (b) otherwise, $v(A) = 1$.

Therefore, it is more difficult to build an **mCi**-valuation than a **CPL**-valuation[7]. But, given a formula A, if we have a valuation for all formulas in ESSF(A), it is easy to verify that $v(A)$ can be 0. The algorithm above is clearly polynomial in time (and also in space). As the NP class contains the problems that can be verified in polynomial time [26], the complement of the decision problem (falsification) for **mCi** is in NP. Therefore, the decision problem for **mCi** is co-NP-complete.

[7]A **CPL**-valuation can be built by setting values only to atomic formulas (see Definition B.1.1).

B.3.2 Complexity of Theorem-Proving Procedures

Besides the complexity of decision problems for logics, the complexity of theorem-proving procedures [23] and the length of proofs in **CPL** [22] has also been extensively studied. Given a possible tautology, we are faced with the problem of finding a proof, if one exists [7]. Then we encounter two additional problems: the first is concerned with *the complexity of the proof search* while the second with *the complexity of the smallest possible proof*, which might be exponential in the size of the tautology.

The complexity of proof search algorithms is obviously related to the complexity of the smallest possible proof. For instance, if the smallest proof is exponential in size, the proof search has to be exponential. But sometimes even when the proof system has polynomial size proofs for some classes of problems, current algorithms can be exponential, because it is harder to find the smallest proofs in stronger proof systems.

To study the length of **CPL** proofs, some families of problems that are known to be difficult are used. The pigeon hole principle (PHP) family of problems is probably the most studied of such families. Some works have shown that there are polynomial size proofs of this problem in some propositional proof systems [9, 24].

The resolution method [100] is the most widely studied propositional proof system. It can also be used for first-order classical logic and is implemented by many theorem provers, such as OTTER [77] and Vampire [98]. Resolution has exponential lower bounds for PHP and other classes of formulas [58, 113]. That is, all resolution proofs of PHP are exponential in length.

Another famous and successful decision procedure for **CPL** is the Davis-Logemann-Loveland (DLL) procedure [32]. Its modern variants, such as Chaff [83], are used in the most efficient SAT provers. However, these provers perform poorly on many important families of problems, including PHP. The most competitive SAT solvers show exponential scaling on these simple structured problems. This happens because DLL is based on tree resolution, a variant of resolution. Because of that, the proof search procedure of DLL also has exponential complexity.

Some extensions of DLL (see [40] for a good coverage) make it stronger than tree resolution, but not stronger than resolution. Therefore, one solution to achieve shorter PHP proofs is to use stronger proof systems. However, as we said before, sometimes even when the proof system has polynomial size proofs for some classes of problems, current algorithms can be exponential, because it is harder to find proofs in stronger proof systems. In [40], a DLL style satisfiability solver that uses pseudo-Boolean representation and automates cutting plane [25], an inference system properly stronger than resolution, is presented. This pseudo-Boolean solver allowed exponential speedups over traditional methods on PHP problems.

Complexity of Tableau Methods

The complexity of Analytic Tableaux (**AT**) has been much studied [22, 2, 76]. For instance, Cook and Reckhow established in [22] that the family Σ_n of unsatisfiable formulas gives a lower bound of $2^{\Omega(2^n)}$ on the proof size with **AT**. Later, Massacci [76] exhibited an **AT** proof for Σ_n for whose size he proved an *upper bound* of $O(2^{n^2})$, which, although not polynomial in the size $O(2^n)$ of the input, is exponentially shorter than the claimed lower bound.

The **KE** system was proven to be more efficient than Smullyan's tableaux in [29]. There, a simple refutation procedure for **KE** was defined and called *canonical procedure.* And the *canonical restriction of* **KE** was defined as **KE** used with this canonical proof search procedure. The canonical restriction of **KE** can polinomially simulate the analytic tableau method, but the tableau method *cannot* polinomially simulate the canonical restriction of **KE**. In other words, for each analytic tableau proof of size n, there is a **KE**-Tableau proof with size polynomial in n. But there is at least one proof in **KE**-Tableau of size n whose corresponding proof in Analytic Tableaux has size superpolynomial in n. Besides that, the **KE** system polinomially simulates the truth-table procedure, although the analytic tableau method does not.

The **KE** system is more efficient than **AT** because it is based on Sequent Calculus with Cut, while the analytic tableau method is based on cut-free Sequent Calculus. It is well known that several families of problems have only exponential size proofs in cut-free

Sequent Calculus, but can have polynomial size proofs in Sequent Calculus with Cut. The set of rules for the **KE** system has only one rule of the branching type, the (PB) rule. Even if we add this rule to the analytic tableau method, it is not difficult to construct with this extended system short refutations of the 'hard examples' for Smullyan's tableaux [29]. However, in this system the (PB) rule is dispensable, while in **KE** formulation it is essential.

The complexity of **KE** deserves more study. According to [29], "a detailed study of proof-search in the **KE** system will have to involve more sophisticated criteria for the choice of the (PB)-formulae. A good choice may sometimes be crucial for generating essentially shorter proofs than those generated by the analytic tableau method".

Referências Bibliográficas

[1] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. In *STOC '02: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 448–456, New York, NY, USA, 2002. ACM Press.

[2] Noriko Arai, Toniann Pitassi, and Alasdair Urquhart. The complexity of analytic tableaux. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 356–363. ACM Press, 2001.

[3] Bernhard Beckert, Richard Bubel, Elmar Habermalz, and Andreas Roth. jTAP - a Tableau Prover in Java, February 1999. Universitat Karlsruhe. Available at `http://i12www.ira.uka.de/~aroth/jTAP/`. Last accessed, November 2006.

[4] Bernhard Beckert and Joachim Posegga. leanTAP: Lean tableau-based deduction. *Journal of Automated Reasoning*, 15(3):339–358, 1995.

[5] Evert W. Beth. *The Foundations of Mathematics.* North-Holland Publishing Company, Amsterdam, 1959.

[6] Maria Paola Bonacina and Thierry Boy de la Tour. Fifth Workshop on Strategies in Automated Deduction - Workshop Programme, 2004. `http://tinyurl.com/y8dkbj`. Last accessed, November 2006.

[7] Maria Luisa Bonet and Nicola Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *FOCS '99: Proceedings of the 40th Annual*

Symposium on Foundations of Computer Science, page 422, Washington, DC, USA, 1999. IEEE Computer Society.

[8] Krysia Broda, Marcello D'Agostino, and Marco Mondadori. A Solution to a Problem of Popper. In *The Epistemology of Karl Popper*. Kluwer, 1995. `http://citeseer.nj.nec.com/broda95solution.html`. Last accessed, November 2006.

[9] S. R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[10] Liming Cai. *Nondeterminism and optimization.* PhD thesis, Texas A&M University, USA, 1994.

[11] Carlos Caleiro, Walter Carnielli, Marcelo E. Coniglio, and Joao Marcos. Two's company: "The humbug of many logical values". In *Logica Universalis*, pages 169–189. Birkhäuser Verlag, Basel, Switzerland, 2005. Pre-print available at `http://tinyurl.com/yb5qbz`. Last accessed, November 2006.

[12] Alessandra Carbone and Stephen Semmes. *Graphic Apology for Symmetry and Implicitness.* Oxford University Press, 2000.

[13] W. A. Carnielli. Systematization of the finite many-valued logics through the method of tableaux. *The Journal of Symbolic Logic*, 52:473–493, 1987.

[14] W.A. Carnielli and J. Marcos. Ex Contradictione Non Sequitur Quodlibet. *Bulletin of Advanced Reasoning and Knowledge*, 1:89–109, 2001.

[15] W.A. Carnielli and J. Marcos. Tableau systems for logics of formal inconsistency. *Proceedings of the International Conference on Artificial Intelligence (IC-AI'2001)*, pages 848–852, 2001.

[16] Walter Carnielli. How to build your own paraconsistent logic: an introduction to the Logics of Formal (In)Consistency. *Proceedings of the Workshop on Paraconsistent Logic (WoPaLo)*, 2002.

[17] Walter Carnielli, Marcelo Coniglio, and Ricardo Bianconi. *Logic and Applications: Mathematics, Computer Science and Philosophy (in Portuguese).* Unpublished, 2005. Preliminary version. Chapters 1 to 5.

[18] Walter Carnielli, Marcelo E. Coniglio, and Joao Marcos. Logics of Formal Inconsistency. In *Handbook of Philosophical Logic*, volume 12. Kluwer Academic Publishers, 2005. To appear. Pre-print available at `http://tinyurl.com/ybn4yw`. Last accessed, November 2006.

[19] Walter A. Carnielli and Richard L. Epstein. *Computabilidade – Funções Computáveis, Lógica e os Fundamentos da Matemática.* Editora da Unesp, 2006.

[20] Stephen Cook. The P versus NP problem, 2000. `http://tinyurl.com/n5thm`. Last accessed, May 2005.

[21] Stephen Cook. The importance of the P versus NP question. *J. ACM*, 50(1):27–29, 2003.

[22] Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 135–148, New York, NY, USA, 1974. ACM Press.

[23] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

[24] Stephen A. Cook. A short proof of the pigeon hole principle using extended resolution. *SIGACT News*, 8(4):28–32, 1976.

[25] W. Cook, C. R. Coullard, and G. Turán. On the complexity of cutting-plane proofs. *Discrete Appl. Math.*, 18(1):25–38, 1987.

[26] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms - Second Edition.* MIT Press, 2001.

[27] Newton C. A. da Costa, Décio Krause, and Otávio Bueno. Paraconsistent logics and paraconsistency: Technical and philosophical developments. *CLE e-prints (Section Logic)*, 4(3), 2004. Pre-print available at `http://tinyurl.com/yxhon7`. Last accessed, November 2006.

[28] Marcello D'Agostino. Are Tableaux an Improvement on Truth-Tables? Cut-Free proofs and Bivalence, 1992. Available at `http://citeseer.nj.nec.com/140346.html`. Last accessed, May 2005.

[29] Marcello D'Agostino. Tableau methods for classical propositional logic. In Marcello D'Agostino et al., editor, *Handbook of Tableau Methods*, chapter 1, pages 45–123. Kluwer Academic Press, 1999.

[30] Marcello D'Agostino, Dov Gabbay, and Krysia Broda. Tableau methods for substructural logics. In Marcello D'Agostino et al., editor, *Handbook of Tableau Methods*, chapter 6, pages 397–467. Kluwer Academic Press, 1999.

[31] Marcello D'Agostino and Marco Mondadori. The taming of the cut: Classical refutations with analytic cut. *Journal of Logic and Computation*, pages 285–319, 1994.

[32] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962.

[33] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[34] Sandra de Amo, Walter Carnielli, and João Marcos. A Logical Framework for Integrating Inconsistent Information in Multiple Databases. In Thomas Eiter and Klaus-Dieter Schewe, editors, *Lecture Notes in Computer Science*, volume 2284, pages 67–84. Springer-Verlag, Berlim., 2002.

[35] Eleonora de Moraes. *Lista de discussão gestar bem interior-sp*, 2004. `http://br.groups.yahoo.com/group/gestarbeminterior-sp`. Last accessed, December 2006.

[36] Rina Dechter. *Constraint Processing.* Morgan Kaufmann, 2003.

[37] Luis Fariñas del Cerro, David Fauthoux, Olivier Gasquet, Andreas Herzig, Dominique Longin, and Fabio Massacci. Lotrec: The generic tableau prover for modal and description logics. In *IJCAR '01: Proceedings of the First International Joint Conference on Automated Reasoning*, pages 453–458. Springer-Verlag, 2001.

[38] Wagner Dias. Tableaux implementation for approximate reasoning (in portuguese). Master's thesis, Computer Science Department, Institute of Mathematics and Statistics, University of São Paulo, 2002.

[39] Simone Grilo Diniz and Ana Cristina Duarte. *Parto normal ou cesárea? O que toda mulher deve saber (e todo homem também).* Editora da Unesp, São Paulo, 2004.

[40] Heidi Dixon. *Automating Pseudo-Boolean Inference Within a DPLL Framework.* PhD thesis, University of Oregon, USA, Dec 2004. Available at `http://www.cirl.uoregon.edu/dixon/dixonDissertation.pdf`. Last accessed, August 2005.

[41] Amigas do Parto. *Lista de discussão Parto Nosso*, 2003. `br.groups.yahoo.com/group/partonosso`. Last accessed, December 2006.

[42] Itala M. Loffredo D'Ottaviano and Milton Augustinis de Castro. Analytical Tableaux for da Costa's Hierarchy of Paraconsistent Logics $C_n, 1 \leq n \leq \omega$. *Journal of Applied Non-Classical Logics*, 15(1):69–103, 2005.

[43] Tzilla Elrad, Robert E. Filman, and Atef Bader. Aspect-Oriented Programming. *Communications of the ACM*, 44, 2001.

[44] M. Enkin, M. Skiers, J. Nelson, C. Crowder, L. Duly, and E. Hodnett. *A guide to effective care during pregnancy and childbirth.* Oxford, UK: Oxford University Press, 2000.

[45] Fadynha. *Lista de discussão partonatural*, 1999. `br.groups.yahoo.com/group/partonatural`. Last accessed, December 2006.

[46] Luis Fariñas del Cerro, David Fauthoux, Olivier Gasquet, Andreas Herzig, Dominique Longin, and Fabio Massacci. Lotrec: the generic tableau prover for modal and description logics. In *International Joint Conference on Automated Reasoning*, LNCS, page 6. Springer Verlag, 18-23 juin 2001.

[47] M. Finger and R. Wassermann. Approximate Reasoning and Paraconsistency-Preliminary Report. *Proceedings of the Eigth Workshop on Logic, Language, Information and Comunication (WoLLIC), Braslia, Brazil*, 2001.

[48] M. Finger and R. Wassermann. The universe of approximations. *Electronic Notes in Theoretical Computer Science*, 84:1–14, 2003.

[49] M. Finger and R. Wassermann. Anytime Approximations of Classical Logic from Above. *Journal of Logic and Computation*, 2006.

[50] M. Finger and R. Wassermann. The universe of propositional approximations. *Theoretical Computer Science*, 355(2):153–166, 2006.

[51] Melvin Fitting. *First-order logic and automated theorem proving (2nd ed.)*. Springer-Verlag New York, Inc., 1996.

[52] Melvin Fitting. Introduction. In Marcello D'Agostino et al., editor, *Handbook of Tableau Methods*, chapter 1, pages 1–43. Kluwer Academic Press, 1999.

[53] Zhaohui Fu, Yogesh Mahajan, and Sharad Malik. New Features of the SAT'04 versions of zChaff, 2004. `http://www.princeton.edu/~chaff/zchaff/sat04.pdf`. Last accessed, September 2005.

[54] Dov M. Gabbay. *Labelled Deductive Systems, Volume 1.* Oxford University Press, Oxford, 1996.

[55] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Adisson-Wesley, 1994.

[56] Gerhard Gentzen. Investigations into logical deductions, 1935. In M. E. Szabo, editor, *The Collected Works of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, 1969.

[57] J. Gosling, B. Joy, and G. Steele. *The Java Programming Language*. Addison-Wesley, Reading, MA, 1996.

[58] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[59] Klaus Havelund and Natarajan Shankar. Experiments in theorem proving and model checking for protocol verification. In *FME '96: Proceedings of the Third International Symposium of Formal Methods Europe on Industrial Benefit and Advances in Formal Methods*, pages 662–681. Springer-Verlag, 1996.

[60] J. Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:7–55, 1955.

[61] Jan Holub and Borivoj Melichar. Implementation of nondeterministic finite automata for approximate pattern matching. In *WIA '98: Revised Papers from the Third International Workshop on Automata Implementation*, pages 92–99. Springer-Verlag, 1999.

[62] Scott E. Hudson, Frank Flannery, C. Scott Anaian, Dan Wang, and Andrew Appel. *CUP Parser Generator for Java*, 1999. `http://www2.cs.tum.edu/projects/cup`. Last accessed, November 2006.

[63] Ullrich Hustadt and Renate A. Schmidt. Simplification and backjumping in modal tableau. In *Lecture Notes in Computer Science*, volume 1397 of *Lecture Notes in Computer Science*, pages 187–201, 1998.

[64] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold. Getting Started with AspectJ. *Communications of the ACM*, 44:59–65, 2001.

[65] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, and William G. Griswold. An overview of AspectJ. *Lecture Notes in Computer Science*, 2072:327–355, 2001.

[66] Gerwin Klein. *JFlex: the fast lexical analyzer generator for Java*, 1998. `http://jflex.de`. Last accessed, November 2006.

[67] S. Kundu and J. Chen. Fuzzy logic or Lukasiewicz logic: A clarification. *Fuzzy Sets and Systems*, 95(3):369–379, 1998.

[68] L. Di Lascio. Analytic fuzzy tableaux. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 5(6):434–439, Dec 2001.

[69] D. W. Loveland. Automated deduction: Some achievements and future directions. Technical report, National Science Foundation, 1997. Available at `http://tinyurl.com/y7mb6p`. Last accessed, May 2005.

[70] D. W. Loveland. Automated deduction: achievements and future directions. *Commun. ACM*, 43(11es):10, 2000.

[71] Wendy MacCaull. Tableau method for residuated logic. *Fuzzy Sets Syst.*, 80(3):327–337, 1996.

[72] Alistair Manning, Andrew Ireland, and Alan Bundy. Increasing the Versatility of Heuristic Based Theorem Provers. In *LPAR'93*, 1993.

[73] Heiko Mantel and Jens Otten. lintap: A tableau prover for linear logic. In *TABLEAUX '99: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, pages 217–231, London, UK, 1999. Springer-Verlag.

[74] João Marcos. Personal communication by email, October 2006.

[75] Fabio Massacci. Simplification: A general constraint propagation technique for propositional and modal tableaux. In *TABLEAUX '98: Proceedings of the Inter-*

national Conference on Automated Reasoning with Analytic Tableaux and Related Methods, pages 217–231. Springer-Verlag, 1998.

[76] Fabio Massacci. The proof complexity of analytic and clausal tableaux. *Theor. Comput. Sci.*, 243(1-2):477–487, 2000.

[77] William McCune and Larry Wos. Otter - The CADE-13 Competition Incarnations. *J. Autom. Reason.*, 18(2):211–220, 1997.

[78] Elliott Mendelson. *Introduction to Mathematical Logic.* Chapman & Hall, London, UK, fourth edition, 1997.

[79] Paulo Blauth Menezes. *Linguagens Formais e Autômatos.* Instituto de Informática da UFRGS : Editora Sagra Luzzatto, Porto Alegre, 232p., 2005.

[80] Sun Microsystems. Java Runtime Environment (JRE) 5.0 Installation Notes, 2006. `http://java.sun.com/j2se/1.5.0/jre/install.html`. Last accessed, November 2006.

[81] S. H. Mirian and M. Mousavi. Nondeterminism in set-theoretic specifications (in persian). In *Proceedings of Iranian Computer Society Annual Conference (CSICC'02)*, feb 2002.

[82] David G. Mitchell, Bart Selman, and Hector J. Levesque. Hard and easy distributions for SAT problems. In Paul Rosenbloom and Peter Szolovits, editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 459–465, Menlo Park, California, 1992. AAAI Press.

[83] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an Efficient SAT Solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, June 2001.

[84] John K. Myers. An introduction to planning and meta-decision-making with uncertain nondeterministic action using 2nd-order probabilities. In *Proceedings of the first*

international conference on Artificial intelligence planning systems, pages 297–298. Morgan Kaufmann Publishers Inc., 1992.

[85] Adolfo Neto. An Object-Oriented Implementation of a KE Tableau Prover, nov 2003. Avaliable at `http://tinyurl.com/y3qmkx`. Last accessed, November 2006.

[86] Adolfo Neto. Modifications on the implementation of a framework for tableau methods, jul 2003. Avaliable at `http://tinyurl.com/yjgjnq`. Last accessed, November 2006.

[87] Adolfo Neto and Marcelo Finger. A Multi-Strategy Tableau Prover. In *I Simpósio de Iniciação Científica e Pós-Graduação do IME-USP*. University of São Paulo, 2005. Available at `http://tinyurl.com/tbdd6`. Last accessed, November 2006.

[88] Adolfo Neto and Marcelo Finger. A Multi-Strategy Tableau Prover. In *SeMe-2005. Workshop "Semantics and Meaning"*, IFIP International Federation for Information Processing. Unicamp. Campinas-SP., 2005. Available at `http://tinyurl.com/yzx8ve`. Last accessed, November 2006.

[89] Adolfo Neto and Marcelo Finger. Implementing a multi-strategy theorem prover. In Ana Cristina Bicharra Garcia and Fernando Santos Osório, editors, *Proceedings of the V ENIA (Encontro Nacional de Inteligência Artificial), held in São Leopoldo-RS, Brazil, July 22-29 2005*, 2005. Available at `http://tinyurl.com/yd6n6n`. Last accessed, November 2006.

[90] Adolfo Neto and Marcelo Finger. Using Aspect-Oriented Programming in the Development of a Multi-Strategy Theorem Prover. In *Anais da II Jornada do Conhecimento e da Tecnologia do Univem*, Marília-SP, 2005. Available at `http://www.ime.usp.br/~adolfo/trabalhos/jornada2005.pdf`. Last accessed, November 2006.

[91] Adolfo Neto and Marcelo Finger. Effective Prover for Minimal Inconsistency Logic. In *Artificial Intelligence in Theory and Practice*, IFIP International Federation for Information Processing, pages 465–474. Springer Verlag, 2006. Available at

http://www.springerlink.com/content/b80728w7m6885765. Last accessed, November 2006.

[92] Adolfo Neto and Marcelo Finger. *KEMS - A* **KE** *Multi-Strategy Tableau Prover*, 2006. http://kems.iv.fapesp.br. Last accessed, November 2006.

[93] Michel Odent. *The Caesarean.* Free Association Books, 2004.

[94] Lawrence C. Paulson. *Handbook of logic in computer science (vol. 2): background: computational structures*, chapter Designing a theorem prover, pages 415–475. Oxford University Press, Inc., 1992.

[95] Francis Jeffry Pelletier. Seventy-five problems for testing automatic theorem provers. *J. Autom. Reason.*, 2(2):191–216, 1986.

[96] J. V. Pitt and R. J. Cunningham. Theorem proving and model building with the calculus ke. *Journal of the IGPL*, 4(1):129–150, 1996.

[97] Awais Rashid and Lynne Blair. Editorial: Aspect-oriented Programming and Separation of Crosscutting Concerns. *The Computer Journal*, 46(5):527–528, 2003.

[98] Alexandre Riazanov and Andrei Voronkov. Vampire 1.1 (system description). In *IJCAR '01: Proceedings of the First International Joint Conference on Automated Reasoning*, pages 376–380. Springer-Verlag, 2001.

[99] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[100] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.

[101] Satisfiability suggested format, 1993. http://www.satlib.org. Last accessed, March 22, 2005.

[102] Satisfiability library, 2003. http://www.satlib.org. Last accessed, March 22, 2005.

[103] N. Scharli, S. Ducasse, O. Nierstrasz, and A.P. Black. Traits: Composable units of behaviour. *Proc. of ECOOP*, 2743:248–274, 2003.

[104] J. Schumann. Tableau-based theorem provers: Systems and implementations. *Journal of Automated Reasoning*, 13(3):409–421, 1994. `http://www.springerlink.com/content/k182u80451306371`. Last accessed, November 4th, 2006.

[105] Bart Selman, Hector J. Levesque, and D. Mitchell. A New Method for Solving Hard Satisfiability Problems. In Paul Rosenbloom and Peter Szolovits, editors, *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 440–446, Menlo Park, California, 1992. AAAI Press.

[106] Raymond M. Smullyan. *First-Order Logic*. Springer-Verlag, 1968.

[107] Sérgio Soares and Paulo Borba. AspectJ - Programação orientada a aspectos em Java. *Tutorial no SBLP 2002, 6o. Simpósio Brasileiro de Linguagens de Programação. 5 a 7 de Junho, PUC-Rio, Rio de Janeiro, Brasil*, pages 39–55, 2002.

[108] R. Statman. Bounds for proof-search and speed-up in the predicate calculus. *Annals of Mathematical Logic*, pages 225–287, 1978.

[109] Friedrich Steimann. The paradoxical success of aspect-oriented programming. *SIGPLAN Not.*, 41(10):481–497, 2006.

[110] Geoff Sutcliffe. An overview of automated theorem proving, 2001. `http://www.cs.miami.edu/~tptp/OverviewOfATP.html`. Last accessed, March 2005.

[111] Geoff Sutcliffe. Thousands of problems for theorem provers, 2001. `http://www.cs.miami.edu/~tptp`. Last accessed, March 2005.

[112] Geoff Sutcliffe and Christian Suttner. The CADE ATP System Competition, 2003. `http://www.cs.miami.edu/~tptp/CASC`. Last accessed, March 2005.

[113] Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987.

[114] Marian Vittek. A compiler for nondeterministic term rewriting systems. In *RTA '96: Proceedings of the 7th International Conference on Rewriting Techniques and Applications*, pages 154–167. Springer-Verlag, 1996.

[115] Wikipedia. *Nondetermnistic algorithm*, 2007. `http://en.wikipedia.org/wiki/Nondeterministic`. Last accessed, February 2007.