Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 355 (2006) 153–166
Theoretical Computer Science
www.elsevier.com/locate/tcs

The universe of propositional approximations

Marcelo Finger*,[1], Renata Wassermann[2]

Departamento de Ciência da Computação, Universidade de São Paulo, São Paulo, Brazil

Abstract

The idea of *approximate entailment* has been proposed by Schaerf and Cadoli [Tractable reasoning via approximation, Artif. Intell. 74(2) (1995) 249–310] as a way of modelling the reasoning of an agent with limited resources. In that framework, a family of logics, parameterised by a set of propositional letters, approximates classical logic as the size of the set increases.

The original proposal dealt only with formulas in clausal form, but in Finger and Wassermann [Approximate and limited reasoning: semantics, proof theory, expressivity and control, J. Logic Comput. 14(2) (2004) 179–204], one of the approximate systems was extended to deal with full propositional logic, giving the new system semantics, an axiomatisation, and a sound and complete proof method based on tableaux. In this paper, we extend another approximate system by Schaerf and Cadoli, presented in a subsequent work [M. Cadoli, M. Schaerf, The complexity of entailment in propositional multivalued logics, Ann. Math. Artif. Intell. 18(1) (1996) 29–50] and then take the idea further, presenting a more general approximation framework of which the previous ones are particular cases, and show how it can be used to formalise heuristics used in theorem proving.

Keywords: Automated reasoning; Deductive systems; Approximate reasoning; Non-classical logics; Knowledge representation

1. Introduction

Logic has been used in several areas of Artificial Intelligence as a tool for representing knowledge as well as a tool for modelling agents' reasoning. Ideal agents know all the logical consequences of their beliefs. However, real agents are *limited* in their capabilities. Due to these limitations, a real rational agent must devise some strategy to make good use of the available resources.

In this work, we propose a general framework for modelling limited reasoning, and show two systems which are special cases of the more general framework. Several approximations of propositional classical logics have been proposed in the literature with distinct properties. Our system is based on Cadoli and Schaerf's *approximate entailment* [25]. Their method consists in defining different logics for which satisfiability is easier to compute than classical logic and treat these logics as upper and lower bounds for the classical problem. In [25], these approximate logics are defined by means of valuation semantics and algorithms for testing satisfiability. The language they use is restricted to that of clauses, i.e., negation appears only in the scope of atoms and there is no implication.

* Corresponding author. Tel.: +55 11 3091 6135; fax: +55 11 3091 6134.
E-mail addresses: mfinger@ime.usp.br (M. Finger), renata@ime.usp.br (R. Wassermann).
[1] Partly supported by the Brazilian Research Council (CNPq), Grant PQ 300597/95-5.
[2] Partly supported by CNPq Grant PQ 300196/01-6.

0304-3975/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.01.007

Approximate entailment has been used to formalise approximate diagnosis [29] and belief revision [9]. However, the knowledge had to be encoded in clausal form. It happens that each approximation step is characterised by a formal logic. The final step of the approximations is classical logic, in which every formula is equivalent to one in clausal form. However, in *none* of the intermediate systems such equivalence holds.

The original system has been extended to deal with full propositional logic in [21]. In this paper, we extend another system, S_3^*, which was introduced in [7]. We provide a proof method based on tableaux for extended S_3^* and then show that both S_3 and S_3^* are particular cases of a system that we call S_e. We provide semantics and a tableaux method for S_e. We then show how this general system can be used to formalise different heuristics used in theorem proving.

The rest of the paper develops as follows: in the next section, we introduce the notion of classical logic approximation and comment on some existing systems. We then present in more detail the two systems S_3 and S_3^* as proposed by Cadoli and Schaerf and their extensions. In Section 5, we present the general system, of which the previous two are particular cases and show how it can be used to simulate several heuristics of theorem proving.

Notation. Let $\mathcal{P}$ be a countable set of propositional letters. We concentrate on the classical propositional language $\mathcal{L}_C$ formed by the usual boolean connectives $\rightarrow$ (implication), $\wedge$ (conjunction), $\vee$ (disjunction) and $\neg$ (negation).

Throughout the paper, we use lowercase Latin letters to denote propositional letters, lowercase Greek letters to denote formulas, and uppercase letters (Greek or Latin) to denote sets of formulas.

Let $S \subset \mathcal{P}$ be a finite set of propositional letters. We abuse notation and write that, for any formula $\alpha \in \mathcal{L}_C$, $\alpha \in S$ if all its propositional letters are in S. A *propositional valuation* v is a function $v : \mathcal{P} \rightarrow \{0, 1\}$.

If Γ and Δ are any sets, by $\|\Gamma - \Delta\|$ we mean the *symmetric difference* of two sets, namely $\|\Gamma - \Delta\| = (\Gamma - \Delta) \cup (\Delta - \Gamma)$.

2. The notion of logical approximation

The general notion of approximation is based on the following. If L is a logic, we say that the sequence of logics $\mathsf{L}_1, \mathsf{L}_2, \ldots$ approximate L if

$$\|\mathsf{L} - \mathsf{L}_1\| \supseteq \|\mathsf{L} - \mathsf{L}_2\| \supseteq \cdots \supseteq \emptyset.$$

By a logic L we mean either:
- the *set of theorems* (or valid formulas) of L, $Th(\mathsf{L})$; or
- the *entailment relation* $\models_{\mathsf{L}} \subseteq 2^{\mathcal{L}_{\mathsf{L}}} \times \mathcal{L}_{\mathsf{L}}$, where $\mathcal{L}_{\mathsf{L}}$ is the set of formulas of L.

Furthermore, the approximation is *finite* if there exists a L_n in the sequence such that $\mathsf{L}_n = \mathsf{L}$.

There are two special cases of approximation worth of noting. We say that the sequence $\mathsf{L}_1, \mathsf{L}_2, \ldots$ is an *approximation from below* if

$$\mathsf{L}_1 \subset \mathsf{L}_2 \subset \cdots \subset \mathsf{L}.$$

Approximations from below are useful for theorem proving. Each logic L_i proves more theorems of the goal logic L. On the other hand, we may have an *approximation from above* when

$$\mathsf{L}_1 \supset \mathsf{L}_2 \supset \cdots \supset \mathsf{L}.$$

Approximations from above are useful for disproving theorems, which in the case of propositional classical logic is equivalent to the well-known SAT problem. The SAT problem consists in, given a set of propositional formulas Γ, deciding whether Γ is satisfiable, i.e., whether a truth assignment can be found that makes each formula of Γ true.

We say that an approximation is *parameterised* if there exists a universe set U and parameter sets $S \subseteq U$ such that $\mathsf{L}(S)$ is defined for each set $S \subseteq U$. In this case, an approximation from below parameterised by sets of propositional letters would be a sequence such that whenever $\emptyset \subset S^1 \subset S^2 \subset \cdots \subset \mathcal{P}$ we have

$$\mathsf{L}(\emptyset) \subset \mathsf{L}(S^1) \subset \mathsf{L}(S^2) \subset \cdots \subset \mathsf{L}(\mathcal{P}) = \mathsf{L}.$$

We can similarly define a parameterised approximation from above, or a generic parameterised approximation. Given a parameterised approximation, an *anytime algorithm* is a procedure that allows us to expand the parameter set S in

such a way that anytime the algorithm is stopped, it produces a partial answer and the longer the algorithm runs, the better the quality of the partial answer. In case of approximated theorem proving from below, say of formula α, the anytime algorithm would keep expanding S until a decision is made. Such decision would be that $\alpha \in Th(\mathsf{L}(S))$. If the approximation is interrupted just after verifying that $\alpha \notin Th(\mathsf{L}(S^i))$ we have a *partial answer* that up to logic $\mathsf{L}(S^i)$, α is not a theorem; maybe, if not all of the propositional letters of α were in S and the algorithm proceeded further with the approximations it would verify that α is a theorem.

We say that this process of approximation is *incremental* if the algorithm for testing whether $\alpha \in Th(\mathsf{L}(S^{i+1}))$ can be started from the state where the proof of $\alpha \notin Th(\mathsf{L}(S^i))$ stopped. A non-incremental algorithm would try to decide whether $\alpha \in Th(\mathsf{L}(S^{i+1}))$ as if it had never tried to decide if $\alpha \in Th(\mathsf{L}(S^i))$. Analogously, anytime and incremental behaviours can be defined for approximations from above.

In the literature we find many logics that approximate propositional classical logics from below, but a much smaller number of approximations from above (see [20] for a discussion on why it is harder to approximate from above). In this paper, we will focus on parameterised methods of approximating propositional classical logic from below with incremental anytime algorithms.

We now examine some of the relevant proposals of logical approximations existing in the literature.

2.1. Selman and Kautz's knowledge compilation

Perhaps the first proposal of approximation in the literature that fits the notion of approximation above is the work of Selman and Kautz on *knowledge compilation* [26,27].

The idea of knowledge compilation can be applied to a large class of logic theories [27], but the instance that interests us here is the compilation of Horn-clause approximations for a given propositional theory in clausal form [26]. In this approach, a clausal-form theory Σ is compiled into two approximated theories, namely the *greatest Horn lower-bound*, Σ_{glb} and the *least Horn upper-bound*, Σ_{lub}, such that $\Sigma_{\text{glb}} \models \Sigma \models \Sigma_{\text{lub}}$. While the theory Σ_{lub} is unique, there may be several greatest Horn lower-bounds.

The computation of Σ_{lub} and Σ_{glb} proceeds in steps, where each step computes a greater lower bound or a smaller upper bound. The computation of Σ_{lub} and Σ_{glb} is NP-hard with the exact complexity studied in [4,5]; the computation of Σ_{lub} and Σ_{glb} can be interrupted anytime, providing an anytime algorithm that generates theories Σ_{ub} and Σ_{lb} that satisfy $\Sigma_{\text{lb}} \models \Sigma \models \Sigma_{\text{ub}}$. The chain of upper bounds thus computed can be thought of as an approximation from below, for it can be used in theorem proving. Similarly, the chain of lower bound Horn theories can be thought of as an approximation from above, and can be used to refuting inferences, that is, for showing that $\Gamma \not\models \alpha$ which, in the case where $\alpha = \bot$ is the problem of satisfying the set of formulas Γ, i.e., the SAT problem.

This method does not fit exactly to the notion of approximation above, for the approximation chains to not converge exactly to the initial clausal theory Σ. That is, there may be a formula α such that $\Sigma \models \alpha$ but $\Sigma_{\text{lub}} \not\models \alpha$. Similarly, there may be a β such that $\Sigma \not\models \beta$ but $\Sigma_{\text{glb}} \models \beta$ for all Σ_{glb}. The solution to the case where neither $\Sigma_{\text{lub}} \models \gamma$ nor $\Sigma_{\text{glb}} \not\models \gamma$ is to fall back to the original theory Σ. Selman and Kautz performed practical experiments such that, on the average, the knowledge compilation method compares positively against other propositional theorem proving methods in the sense that the time of compilation into Horn LUBs and GLBs plus the time of answering a query using these approximations was, on average, smaller than the time spent answering the query directly from the original theory.

Note that this approach is not parameterised. The semantics of Horn approximations Σ_{lub} and Σ_{glb} were studied in [4,5], but the approximation steps Σ_{ub} and Σ_{lb} have no clear semantic status. So, if the anytime algorithm is interrupted before reaching the upper or lower bound, the significance of this intermediary theory is unclear.

2.2. Cadoli and Schaerf's approximate entailment

The notion of *approximate entailment* was initially proposed by Cadoli and Schaerf in [6] and was later extended and detailed in [25]. Dealing with the clausal fragment of propositional logic (later extended to negation normal form formulas [7]), they proposed two parameterised families of logics.

The family S_3 takes a parameter set $S \subseteq \mathcal{P}$ and consists of a family of subclassical logics that are sound but incomplete with respect to classical logic. The family S_1 also takes a parameter set $S \subseteq \mathcal{P}$ and consists of a family of unsound but complete logics with respect to classical logic. In the limit, when $S = \mathcal{P}$, we have that the two families converge to propositional classical logic $\mathcal{L}_C$, $S_3(S) = S_1(S) = \mathcal{L}_C$.

The family $S_3(S)$ will be discussed in details in Section 3, and it does consist of an approximation of classical logic from below, useful in theorem proving. The family of logics $S_1(S)$ is not strictly an approximation from above, as defined in the beginning of this section, but it can also be used to approximately disprove theorems and SAT.

In this way, two approximated parameterised entailment relations were defined, $\models_S^3$ and $\models_S^1$. Their definition was done in semantic terms, so that an anytime algorithm was created and, if the approximations are interrupted before normal termination, there is a clear semantics of the logics at which the search for proving or refuting a theorem failed. However, no incremental proof-theoretical presentation was given for those logics. For a fixed parameter set S, the families possess a polynomial-time decision procedure, but for the general case the decision problem is intractable.

The work of Cadoli and Schaerf really created a paradigm in approximated logics and will be further discussed in Section 3.

2.3. Dalal's BCP approximations

In [12] Dalal proposed a family of logics that approximate classical logic. Dalal's proposal was based on clausal Boolean constraint propagation (BCP) [24], which is a variant of unit resolution [8]. It also deals only with the clausal fragment of propositional logic, and has the important property that all approximations are polynomial with respect to the number of atoms, but for every approximation step the degree of the polynomial is incremented.

In this approach, a theory is a set of clauses and $\mathbf{f}$ is the empty clause. An equivalence relation is algebraically imposed on terms:

$$\{\mathbf{f}\} \cup \Gamma =_{BCP} \Gamma,$$

$$\{l, \sim l \vee \varphi\} \cup \Gamma =_{BCP} \{l, \varphi\} \cup \Gamma,$$

where l is a literal (an atom or its negation) and $\sim l$ represents the literal with opposite polarity. It is then defined that $\Gamma \vdash_{BCP} \varphi$ iff $\Gamma \cup \{\varphi\} =_{BCP} \{\mathbf{f}\}$. Such an inference is correct but incomplete with respect to classical logic. In particular, it is possible that $\Gamma \vdash_{BCP} l$, $\Gamma \cup \{l\} \vdash_{BCP} \beta$ but $\Gamma \not\vdash_{BCP} \beta$. So Dalal defines the parametric family of anytime approximations, $\vdash_k^{BCP}$, respecting two rules:

$$1.\ \frac{\Gamma \vdash_{BCP} \varphi}{\Gamma \vdash_k^{BCP} \varphi} \qquad 2.\ \frac{\Gamma \vdash_k^{BCP} \psi; \qquad \Gamma, \psi \vdash_k^{BCP} \varphi}{\Gamma \vdash_k^{BCP} \varphi} \quad \text{for } |\psi| \leqslant k.$$

This is clearly a parametric family of logics that approximates classical logic from below. No similar logical approximation was provided for refuting a formula. A many-valued semantics for those systems was provided in [13].

This approach has been recently extended to the full fragment of propositional logic [15,16]. This extension maintains the property that all approximations are polynomial with respect to the number of atoms, but for every approximation step the degree of the polynomial is incremented. A new, lattice-based semantics has been provided for this family of approximations. Furthermore, an incremental proof-theory has been proposed based on KE-tableaux, with a heuristics that guides the incremental approximations [15].

2.4. Massacci's system

Massacci's approach [23] was the first one to contemplate extending the notion of approximation to the full fragment of propositional logics. In fact, Massacci actually extended Cadoli and Schaerf's approach to the set of *signed formulas*, where propositional formulas are signed with either a T or an F symbol, in the same way that signed formulas are treated in Smullyan's analytic tableaux [28].

As in Cadoli and Schaerf families, the logics are parameterised with two disjoint context sets, namely the set of *unknown atoms* and the set of *incoherent atoms*. If an atom p is incoherent, then both signed formulas $T\ p$ and $F\ p$ are true. Conversely, if an atom q is unknown, then both signed formulas $T\ q$ and $F\ q$ are false.

When both sets of incoherent and unknown atoms are empty, we are in classical logic so the approximation process starts by making one context set empty and the other containing all atoms, and proceeds by removing atoms from this set, thus generating a family that converges to classical logic.

Massacci provides a semantical extension of this notion to all signed formulas. A tableau system is given that is correct and complete, for each pair of parameter sets, with respect to its semantics.

2.5. Finger and Wassermann's system

In [20] and [21], Finger and Wassermann have examined Cadoli and Schaerf's systems S_1 and S_3 and based on them, proposed systems to approximate the full propositional fragment from above and from below. This approach was able to deal with pure, unsigned formulas. While Massacci's approach was based on giving non-classical interpretation to propositional formulas, this approach gives a non-classical interpretation to the negation symbol.

In Cadoli and Schaerf's system S_1, some theorems of classical logic do not hold, which prevents it from being called an approximation from above according to the definitions in the beginning of this section. This problem is discussed in [20], where a new system s_1 is proposed.

The system which approximates from below will be described in the next section, and is a proper extension of Cadoli and Schaerf's S_3, in the sense that they coincide when restricted to the clausal fragment. It is interesting to see that these systems differ from classical logic only in the way they deal with negation. All the other connectives are handled classically. In Section 5, we will see a generalisation of this idea which changes the way in which all connectives are handled.

3. The family S_3

In this section, we first present Cadoli and Schaerf's system S_3 and then Finger and Wassermann's extended version that deals with full propositional logic.

Schaerf and Cadoli [25] define two approximations of classical entailment: $\models^1_S$ which is complete but not sound, and $\models^3_S$ which is classically sound but incomplete. These approximations are carried out over a set of atoms $S \subseteq \mathcal{P}$ which determines their closeness to classical entailment. Here we will concentrate only in the latter, namely the S_3 family of logics.

In the trivial extreme of S_3 approximate entailment, i.e., when $S = \mathcal{P}$, classical entailment is obtained. At the other extreme, $\models^3_\emptyset$ corresponds to Levesque's logic for explicit beliefs [22], which bears a connection to Relevance Logics such as those of Anderson and Belnap [1].

In an S_3 assignment, if $p \in S$, then p and $\neg p$ get opposite truth values, while if $p \notin S$, p and $\neg p$ do not both get 0, but may both get 1. The name S_3 comes from the possible truth assignments for literals outside S. If $p \notin S$, there are three possible S_3 assignments, the two classical ones, assigning p and $\neg p$ opposite truth values, and an extra one, making them both true. The set of formulas for which we are testing entailments is assumed to be in clausal form.

Formally, the semantics of the logic $S_3(S)$ over clauses is constructed by defining an S_3-valuation v^3_S of literals into $\{0, 1\}$ such that

- $v^3_S(\neg p) = 1$ iff $v^3_S(p) = 0$, if $p \in S$.
- if $p \notin S$, we can have one of 3 possibilities:
  - $v^3_S(\neg p) = 1$ and $v^3_S(p) = 0$,
  - $v^3_S(\neg p) = 0$ and $v^3_S(p) = 1$,
  - $v^3_S(\neg p) = v^3_S(p) = 1$.

This valuation can be extended immediately to clauses. By varying S, we generate a family of logics. Also, satisfiability, validity and entailment are defined in the usual way.

Although in classical logic any formula is equivalent to one in clausal form, the usual transformation does not preserve truth-values under the non-standard S_3 semantics. The S_3 family of logics has been extended to propositional formulas in [18], where a sound and complete incremental proof system for it was also provided.

The generalized semantics for S_3 is the following:

Definition 1. An S_3-*valuation* v^3_S is a function, $v^3_S : \mathcal{L}_C \to \{0, 1\}$, that extends a propositional valuation v (i.e., $v^3_S(p) = v(p)$), satisfying the following restrictions:

$$\text{(i)}\quad v^3_S(\alpha \wedge \beta) = 1 \;\Leftrightarrow\; v^3_S(\alpha) = v^3_S(\beta) = 1,$$
$$\text{(ii)}\quad v^3_S(\alpha \vee \beta) = 0 \;\Leftrightarrow\; v^3_S(\alpha) = v^3_S(\beta) = 0,$$

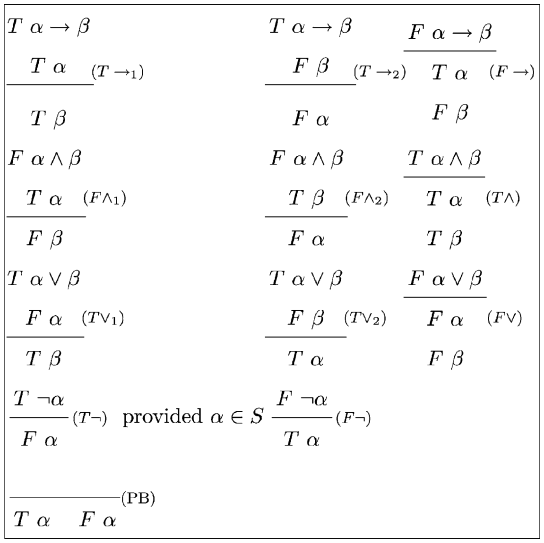

Fig. 1. KE-rules for S_3.

(iii) $v_S^3(\alpha \to \beta) = 0 \quad \Leftrightarrow \quad v_S^3(\alpha) = 1$ and $v_S^3(\beta) = 0$,
(iv) $v_S^3(\neg\alpha) = 0 \quad \Rightarrow \quad v_S^3(\alpha) = 1$,
(v) $v_S^3(\neg\alpha) = 1, \alpha \in S \ \Rightarrow \ v_S^3(\alpha) = 0$.

Rules (i)–(iii) are exactly those of classical logic. Rules (iv) and (v) restrict the semantics of negation: rule (iv) states that if $v_S^3(\neg\alpha) = 0$, then negation behaves classically and forces $v_S^3(\alpha) = 1$; rule (v) states that if $v_S^3(\neg\alpha) = 1$, negation must behave classically if $\alpha \in S$. Formulas outside S may behave classically or *paraconsistently*, i.e., both the formula and its negation may be assigned the truth value 1.

Note that an S_3-valuation is not uniquely defined by the propositional valuation it extends. This is due to the fact that if $\alpha \notin S$ and $v_S^3(\alpha) = 1$, the value of $v_S^3(\neg\alpha)$ can be either 0, in which case α has a classical behaviour, or 1, in which case α behaves paraconsistently. A comparison between S_3 semantics and axiomatisation and da Costa's Paraconsistent Logic C_1 was done in [17].

A formula α is S_3-*valid* if $v_S^3(\alpha) = 1$ for any S_3-valuation. A formula is S_3-*satisfiable* if there is at least one v_S^3 such that $v_S^3(\alpha) = 1$. The S_3-*entailment* relationship between a set of formulas Γ and a formula α is represented as

$$\Gamma \models_S^3 \alpha$$

and holds if every valuation v_S^3 that simultaneously satisfies all formulas in Γ also satisfies α. A formula is S_3-valid if and only if it is entailed by $\emptyset$, represented as $\models_S^3 \alpha$.

An inference system for the full logic S_3 based on the KE-tableau methodology was developed in [18] and further developed in [19,21]. KE-tableaux were introduced by D'Agostino [10] as a principled computational improvement over Smullyan's Semantic Tableaux [28], and have since been successfully applied to a variety of logics [11,2,3].

KE-tableaux deal with T- and F-signed formulas. An expansion of a tableau is allowed when the premises of an expansion rule are present in a branch; the expansion consists of adding the conclusions of the rule to the end of all branches passing through the set of all premises of that rule.

For each connective, there are at least one T- and one F-linear expansion rules. Linear expansion rules always have a main premise, and may also have an auxiliary premise. They may have one or two consequences. The only branching rule is the *Principle of Bivalence*, stating that a formula has to be either true of false. Fig. 1 shows KE-tableau expansion rules for the family S_3.

The only way in which such a tableau system differs from a classical one is in the $(T\ \neg)$ rule, which comes with a proviso

$$\frac{T\ \neg\alpha}{F\ \alpha} \quad \text{provided that } \alpha \in S$$

(recall that, as mentioned in Section 1, $\alpha \in S$ means that all atoms in α are in S).

The meaning of this rule is that the expansion of a branch is only allowed if it contains the rule's antecedent *and the proviso is satisfied*, that is, all the atoms of the formula in question belong to S. This rule is actually a restriction of the classical rule, stating that if $\alpha \notin S$ the $(T\ \neg)$-rule cannot be applied. Let us call the system thus obtained KES_3.

This makes the system immediately subclassical, for any tableau that closes for KES_3 also closes for classical logic. So any theorem we prove in KES_3 are also classical theorems.

So KES_3 is correct and incomplete with respect to classical logic. In fact, KES_3 is complete and correct with respect to the semantics above.

Theorem 2 (*Finger and Wassermann [18]*). $\alpha_1, \ldots, \alpha_n \models^3_S \beta$ *iff any possible* KES_3 *tableau for* $\alpha_1, \ldots, \alpha_n \vdash \beta$ *closes. Furthermore, if one* S_3 *tableau for* $\alpha_1, \ldots, \alpha_n \vdash \beta$ *closes, any such tableau closes.*

4. The dual family S_3^*

Cadoli and Schaerf in a subsequent work [7] have proposed a dual family of logics which they called S_3^*. An S_3^*-valuation of literals into $\{0, 1\}$ such that:

- $v_S^{3*}(\neg p) = 1$ iff $v_S^{3*}(p) = 0$, if $p \in S$.
- if $p \notin S$, we can have one of 3 possibilities:
  - $v_S^{3*}(\neg p) = 1$ and $v_S^{3*}(p) = 0$,
  - $v_S^{3*}(\neg p) = 0$ and $v_S^{3*}(p) = 1$,
  - $v_S^{3*}(\neg p) = v_S^{3*}(p) = 0$.

Only this last line differs from the previous S_3 family, in that for an atom $p \notin S$, both p and $\neg p$ may be false. As a result, in such a logic, the formula $p \vee \neg p$ is not valid for $p \notin S$, which characterises such logics as *paracomplete*. This logic was presented in [7] with the same setting as S_3 was presented: formulas in clausal form only (in fact, negation normal form was also accepted); no extension to full logic; no proof theory.

In an analogous way to the extension of S_3, we extend here S_3^* to full propositional logic.

Definition 3. An S_3^*-*valuation* v_S^{3*} is a function, $v_S^{3*} : \mathcal{L}_C \to \{0, 1\}$, that extends a propositional valuation v (i.e., $v_S^{3*}(p) = v(p)$), satisfying the following restrictions:

$$\begin{array}{lll}
\text{(i)} & v_S^{3*}(\alpha \wedge \beta) = 1 & \Leftrightarrow\ v_S^{3*}(\alpha) = v_S^{3*}(\beta) = 1, \\
\text{(ii)} & v_S^{3*}(\alpha \vee \beta) = 0 & \Leftrightarrow\ v_S^{3*}(\alpha) = v_S^{3*}(\beta) = 0, \\
\text{(iii)} & v_S^{3*}(\alpha \to \beta) = 0 & \Leftrightarrow\ v_S^{3*}(\alpha) = 1 \text{ and } v_S^{3*}(\beta) = 0, \\
\text{(iv)} & v_S^{3*}(\neg\alpha) = 0, \alpha \in S & \Rightarrow\ v_S^{3*}(\alpha) = 1, \\
\text{(v)} & v_S^{3*}(\neg\alpha) = 1 & \Rightarrow\ v_S^{3*}(\alpha) = 0.
\end{array}$$

The definition of S_3^*-logical consequence, $\models^{3*}_S$ is totally analogous to that of $\models^3_S$.

Also, in an analogous way, we define a KE-tableau proof system for the S-parameterised family of logics S_3^*. The rules for the connectives $\to$, $\wedge$ and $\vee$ are the same as in S_3 (which are the same as the classical rules). The rules

for negation are now

$$\frac{T\ \neg\alpha}{F\ \alpha} \quad \text{and} \quad \frac{F\ \neg\alpha}{T\ \alpha} \quad \text{provided that } \alpha \in S.$$

The rule $(T\ \neg)$ is the classical one, while $(F\ \neg)$ comes with a proviso. This is dual to the situation in S_3.

As an example we illustrate two tableaux. The first one is the principle of contradiction, $p, \neg p \vdash q$, which was not a theorem in paraconsistent S_3 but is now a theorem in S_3^*. The second example is the principle of excluded middle $\vdash p \vee \neg p$, which was a theorem in S_3, but which is not a theorem in paracomplete S_3^*.

$T\ p$	$F\ p \vee \neg p$
$T\ \neg p$	$F\ p$
$F\ q$	$F\ \neg p$
$F\ p$	–
$\times$	

In the first tableau, we simply apply $(T\ \neg)$ to the second line to close the tableau. In the second tableau, we would want to apply $(F\ \neg)$, but when we consider $S = \emptyset$, the proviso precludes us from doing that, and the tableau remains open. The approximation process consists in adding some propositional letters to S and trying to close the tableau. In this example, if we add p to S, we can apply the $F\ \neg\alpha$ rule and the tableau closes. This shows that $p \vee \neg p$ is not a theorem of $S_3^*(\emptyset)$, but in the logic $S_3^*(\{p\})$ the formula can be proved.

We have the following soundness and completeness result for KES_3^* tableaux.

Theorem 4. *$\alpha_1, \ldots, \alpha_n \models_S^{3*} \beta$ iff any possible* KES_3^* *tableau for $\alpha_1, \ldots, \alpha_n \vdash \beta$ closes. Furthermore, if one S_3^* tableau for $\alpha_1, \ldots, \alpha_n \vdash \beta$ closes, any such tableau closes.*

The proof of this theorem is similar to the one for KES_3 in [21] and will appear in a more general form is Section 5.

From the way KES_3^* was built, it is clear that it is also an approximation of classical logic from below. Also, it appears that S_3 and S_3^* are incompatible families, due to the following properties, that come straight from the definitions of v_S^3 and v_S^{3*}, for every $\alpha \notin S$:

(a) In S_3, if $v_S^3(\alpha) = 1$, then $v_S^3(\neg\alpha)$ may be either 0 or 1.

(b) In S_3^*, if $v_S^{3*}(\alpha) = 1$, then $v_S^3(\neg\alpha) = 0$.

(c) In S_3^*, if $v_S^3(\alpha) = 0$, then $v_S^3(\neg\alpha)$ may be either 0 or 1.

(d) In S_3, if $v_S^{3*}(\alpha) = 0$, then $v_S^3(\neg\alpha) = 1$.

However, as we are going to see, this does not consist in any kind of incompatibility, and we may have systems that obey both rules.

If we concentrate on both tableau methods that approximate classical logic from below, we see that both consist of a restriction of one rule in a classical tableaux.

With this view, it turns out that constructing a proof theoretical approximation of classical logic from below is a trivial task!

Creating a family that approximates classical logic from below. It suffices to restrict the use of one (or a set of) rules of one's favourite proof method to a limited set a formulas, in a way that eventually the rule will be applicable to all classical formulas.

It is interesting to see if a semantics can be provided for a limited inference system. In the following, we show that if this limitation is done is a systematic way by restricting the connective rules in the KE-tableau method, it is possible to provide a generic semantics for a very large class of approximated logics.

5. A generalised approximation inference

We have seen how to restrict $(T\neg)$ in KES_3 and how to restrict the use of $(F\neg)$ in KES_3^*. We assumed that the parameter set S governing both approximations was the same, but since there were no interactions between the S_3-rule

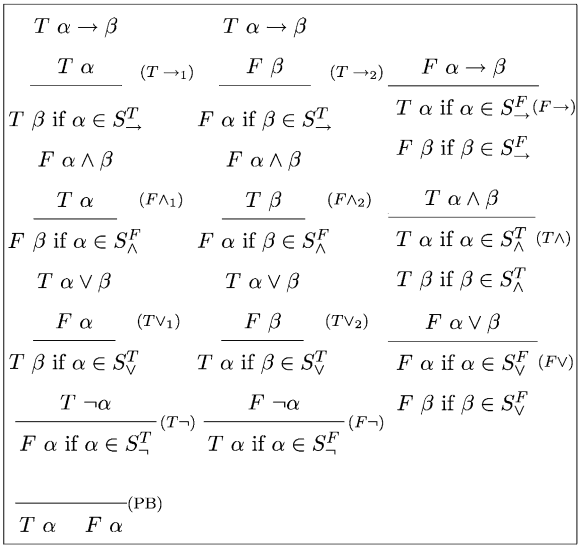

Fig. 2. KE-rules for the generalised system.

and the S_3^* rule, this assumption had no consequences. Now, we are going for greater generality, and we assume different context sets for each rule. This gives us a system where every connective can be assured to behave classically only for formulas in which all propositional letters belong to the corresponding context set: $S^T_\wedge$, $S^F_\wedge$, $S^T_\vee$, $S^F_\vee$, $S^T_\to$, $S^F_\to$, $S^T_\neg$ and $S^F_\neg$.

An initial step towards this generalisation was given in [19], with a system that dealt with $S^T_\neg$ and $S^T_\to$.

We present a generalisation of the KE-tableaux, which we call KES_e, that deals with context sets for all the tableau rules. The system is obtained by adding restrictions to each expansion rule, as illustrated in Fig. 2.

Lemma 5. KES_e *can simulate the dynamic evolution of both* KES_3 *and* KES_3^*.

Proof. To see that KES_e can simulate the dynamic evolution of KES_3, it suffices to set $S = S^T_\neg$ and all other S parameters to the full set of propositional letters. In practice, this amounts to lifting the proviso of all rules except for the $(T\neg)$ rule. Similarly for KES_3^*. □

As usual, we want our system to be based on a sound and complete subclassical semantics.

5.1. Semantics for generalised approximate inference

Definition 6. An S_e*-valuation* v^e_S is a function, $v^e_S : \mathcal{L}_C \to \{0, 1\}$, that extends a propositional valuation v (i.e., $v^e_S(p) = v(p)$), satisfying the following restrictions:

$$
\begin{aligned}
&(\wedge_1)\ v^e_S(\alpha \wedge \beta) = 1, \alpha \in S^T_\wedge && \Rightarrow v^e_S(\alpha) = 1,\\
&(\wedge_2)\ v^e_S(\alpha \wedge \beta) = 1, \beta \in S^T_\wedge && \Rightarrow v^e_S(\beta) = 1,\\
&(\wedge_3)\ v^e_S(\alpha \wedge \beta) = 0, v^e_S(\alpha) = 1, \alpha \in S^F_\wedge && \Rightarrow v^e_S(\beta) = 0,\\
&(\wedge_4)\ v^e_S(\alpha \wedge \beta) = 0, v^e_S(\beta) = 1, \beta \in S^F_\wedge && \Rightarrow v^e_S(\alpha) = 0,\\
&(\vee_1)\ v^e_S(\alpha \vee \beta) = 0, \alpha \in S^F_\vee && \Rightarrow v^e_S(\alpha) = 0,
\end{aligned}
$$

$$
\begin{array}{lll}
(\vee_2) & v^e_S(\alpha \vee \beta) = 0, \beta \in S^F_\vee & \Rightarrow v^e_S(\beta) = 0, \\
(\vee_3) & v^e_S(\alpha \vee \beta) = 1, v^e_S(\alpha) = 0, \alpha \in S^T_\vee & \Rightarrow v^e_S(\beta) = 1, \\
(\vee_4) & v^e_S(\alpha \vee \beta) = 1, v^e_S(\beta) = 0, \beta \in S^T_\vee & \Rightarrow v^e_S(\alpha) = 1, \\
(\rightarrow_1) & v^e_S(\alpha \rightarrow \beta) = 0, \alpha \in S^F_\rightarrow & \Rightarrow v^e_S(\alpha) = 1, \\
(\rightarrow_2) & v^e_S(\alpha \rightarrow \beta) = 0, \beta \in S^F_\rightarrow & \Rightarrow v^e_S(\beta) = 0, \\
(\rightarrow_3) & v^e_S(\alpha \rightarrow \beta) = 1, v^e_S(\alpha) = 1, \alpha \in S^T_\rightarrow & \Rightarrow v^e_S(\beta) = 1, \\
(\rightarrow_4) & v^e_S(\alpha \rightarrow \beta) = 1, v^e_S(\beta) = 0, \beta \in S^T_\rightarrow & \Rightarrow v^e_S(\alpha) = 0, \\
(\neg_1) & v^e_S(\neg\alpha) = 0, \alpha \in S^F_\neg & \Rightarrow v^e_S(\alpha) = 1, \\
(\neg_2) & v^e_S(\neg\alpha) = 1, \alpha \in S^T_\neg & \Rightarrow v^e_S(\alpha) = 0.
\end{array}
$$

It is easy to see that the semantics of S_3 is a particular case of the system above, where the sets $S^T_\wedge$, $S^F_\wedge$, $S^T_\vee$, $S^F_\vee$, $S^T_\rightarrow$, $S^F_\rightarrow$, and $S^F_\neg$ contain all the propositional letters of the language and $S = S^T_\neg$. Similarly, the semantics of S^*_3 corresponds to $S = S^F_\neg$ and $S^T_\wedge = S^F_\wedge = S^T_\vee = S^F_\vee = S^T_\rightarrow = S^F_\rightarrow = S^T_\neg = S^F_\neg = \mathcal{P}$.

The notions of S_e-validity, S_e-satisfiability and S_e-entailment are defined in the usual way.

5.2. Soundness and completeness

We say that KES_e is *sound with respect to the S_e semantics* if whenever a tableau closes for an input sequent, then the sequent's antecedent formulas entail its consequent in S_e. Conversely, the KES_e-tableau method is *complete with respect to the S_e semantics* if for all sequents such that the antecedent entails the consequent in S_e, all KES_e-tableaux close.

We extend an S_e-valuation to signed formulas making $v^e_S(T\alpha) = 1$ iff $v^e_S(\alpha) = 1$ and $v^e_S(F\alpha) = 1$ iff $v^e_S(\alpha) = 0$. A valuation satisfies a branch in a tableau if it simultaneously satisfies all the signed formulas in the branch.

To prove soundness, we first show the correctness of all linear expansion rules of KES_e.

Lemma 7. *If the antecedents of the* KES_e *linear expansion rules are S-satisfied in S_e by v^e_S so are its conclusions.*

Proof. A simple inspection of the rules in Fig. 2 shows the result. □

We now show that the branching rule PB also preserves satisfiability.

Lemma 8. *If a branch is satisfied by a valuation v^e_S prior to the application of PB, then at least one of the two branches generated is satisfied by a valuation v^e_S after the application of PB.*

Proof. Suppose the branching occurs over the formula α. Because v^e_S is a function onto $\{0, 1\}$, we have that $v^e_S(T\,\alpha) = 1$ or $v^e_S(F\,\alpha) = 1$, so v^e_S satisfies one of the two branches generated by the application of PB. □

Theorem 9 (*Soundness*). *Suppose a tableau for $\alpha_1, \ldots, \alpha_n \vdash \beta$ closes. Then $\alpha_1, \ldots, \alpha_n \models^e_S \beta$.*

Proof. We show the contrapositive. Suppose $\alpha_1, \ldots, \alpha_n \not\models^e_S \beta$, so there is a valuation v^e_S such that $v^e_S(\alpha_1) = \cdots = v^e_S(\alpha_1) = 1$ and $v^e_S(\beta) = 0$. In this case, the initial tableau for $\alpha_1, \ldots, \alpha_n \vdash \beta$ is such that all formulas $T\,\alpha_1, \ldots, T\,\alpha_n, F\,\beta$ are satisfied by v^e_S.

By Lemmas 7 and 8, we see that each application of an expansion rule preserves at least one satisfiable branch. As closed branches are not satisfiable, at least one branch remains open and the tableau cannot close. □

We say that a branch of a tableau is complete if there are no more applicable expansion rules.

Lemma 10. *An open complete branch in a* KES_e*-tableau is S-satisfiable in S_e.*

Proof. Let B be the set of formulas that occur in the open complete branch. We have to build an S_e valuation that satisfies it.

We start the construction of v by setting $v(\alpha) = 1$ for every α such that $T\,\alpha \in B$ and $v(\alpha) = 0$ for every α such that $F\,\alpha \in B$. Since the branch is open and complete, there is no formula α for which both $T\,\alpha$ and $F\,\alpha$, hence v is a partial function that satisfies the branch. We have to (i) extend v to a total function and (ii) show that v is an S_e-valuation.

(i) We can extend v by combining it with any classical propositional valuation, by randomly attributing values to the undefined propositions and then extend it classically. So, if $\gamma = \alpha \wedge \beta$ and $v(\gamma)$ undefined. We set $v(\gamma)$ to 1 iff $v(\alpha) = v(\beta) = 1$. Analogously for the other connectives.

(ii) To show that v is an S_e-valuation, we must show that it satisfies the properties in Definition 6. For example, take property $(\wedge_1)$:

$$v^e_S(\alpha \wedge \beta) = 1, \quad \beta \in S^T_\wedge \Rightarrow v^e_S(\alpha) = 1.$$

If $T\,\alpha \wedge \beta$ is in B, then since $\alpha \in S^T_\wedge$, by the rule $(T_\wedge)$, $T\,\alpha$ is also in B and hence, $v(\alpha) = 1$. If $T\,\alpha \wedge \beta$ is not in B, then $v(\alpha \wedge \beta) = 1$ iff $v(\alpha) = 1$ and $v(\beta) = 1$. Consider rule $(\neg_1)$:

$$v^e_S(\neg\alpha) = 0, \quad \alpha \in S^F_\neg \Rightarrow v^e_S(\alpha) = 1.$$

If $F\,\neg\alpha$ is in B and $\alpha \in S^F_\neg$, then by the rule $(F_\neg)$, $T\,\alpha$ is in B and hence, $v(\alpha) = 1$. If $F\,\neg\alpha$ is not in B, then $v(\neg\alpha) = 0$ iff $v(\alpha) = 1$.

In an analogous way, we can show that the valuation v satisfies the other twelve properties in Definition 6. □

Theorem 11 (*Completeness*). *If* $\alpha_1, \ldots, \alpha_n \models^e_S \beta$ *then any possible* KES_e *tableau for* $\alpha_1, \ldots, \alpha_n \vdash \beta$ *closes.*

Proof. Suppose for contradiction that there is a tableau for $\alpha_1, \ldots, \alpha_n \vdash \beta$ with an open complete branch B. Then by Lemma 10 there is an S_e valuation that satisfies B, which includes $T\,\alpha_1, \ldots, T\,\alpha_n, F\,\beta$, contradicting $\alpha_1, \ldots, \alpha_n \models^e_S \beta$. □

5.3. Applications of S_e

We examine here the use of S_e as a formalisation of proof strategies using KE-tableaux. In a tableau expansion, more than one rule may be applicable at a time, and the choice of which rule to use may have dramatic effects, for a short proof may exist but the wrong choice of rule application may lead to an explosion in the number of branches.

Let $X \in \{T, F\}$ and $\bullet \in \{\wedge, \vee, \rightarrow, \neg\}$. The use of an $S^X_\bullet$ context set in KES_e-tableaux may lead to a delay in using the rule $(X\bullet)$. This works as follows: suppose the rule $(X\bullet)$ is classically applicable at one point in the branch expansion, but the corresponding proviso, $\alpha \in S^X_\bullet$, is not met at that point. The use of the rule $(X\bullet)$ is then blocked. All other applicable rules would take precedence, and will be applied. After their application, there are two possibilities: either all branches passing through that $(X\bullet)$-blocked point are closed, in which case there is nothing to be done, or there is at least one open branch. In the latter case, the atoms of the formula α are inserted in $S^X_\bullet$, the logic is changed to one "closer" to classical logic, from $S_e(S^X_\bullet)$ to $S_e(S^X_\bullet \cup \{\alpha\})$, so that the proviso is now met. The expansion of the tableau can then proceed incrementally in the new logic, without having to restart from square 1.

In the systems S_3 and S^*_3 we have seen that most of the sets $S^X_\bullet$ were equal to $\mathcal{P}$. So the choice of which set $S^X_\bullet$ to be chosen to be different from $\mathcal{P}$ has to do with which rule application we want to postpone.

Clearly, we do not want to postpone the application of one-premised linear rules. These rules, which correspond to the α-rules in Smullyan's analytic tableaux, never generate a new branch and are all commutative, for the application of one rule does not invalidate the application of another. This means that for theorem proving purposes we would want to have

$$S^F_\rightarrow = S^T_\wedge = S^F_\vee = S^T_\neg = S^F_\neg = \mathcal{P}.$$

Note that the context sets of S_3 and S^*_3 are included in the sets above that we want to maintain fixed in $\mathcal{P}$. In fact, this is in accordance to some experiments made in [14] with the implementation of KES_3 tableaux, in which a decrease in

performance was noted from the use of KES_3 strategy in respect to classical KE in which the one-premised rules were given application precedence, as above.

The two-premised rules are normally associated with the branching process, which is the important point to concentrate on when trying to reduce the size of a proof. There are two premises in those rules. The main premise is the *main formula*, which in Smullyan's analytic tableaux are associated to β-rules and the branching process. The other premise is the *auxiliary* formula, which is associated with the *KE branching heuristics*. According to such heuristics, given a main formula of a two-premised rule where the auxiliary formula is absent, one should branch using PB so as to generate, in one of the branches, the missing auxiliary formula.

As a result, a strategy for tableau branching based on S_e should keep track of the formulas in the context sets $S^F_\wedge$, $S^T_\vee$ and $S^T_\to$.

The resulting strategy goes in accordance with the general intuition of tableau expansion: first expand the formulas that do not generate new branches, and only then expand the branching rules. Furthermore, our new strategy now places further restrictions on the branching rule, for we are giving preference to branch over a formula with all its atoms already on one of the sets $S^F_\wedge$, $S^T_\vee$ and $S^T_\to$. That is, our strategy gives preference for branching over subformulas of formulas over which there has already occurred a branching operation higher up in the tableau.

Example 12. In this example we will consider $S = S^F_\wedge = S^T_\vee = S^T_\to$. That is, if there is a branching over some formula, subsequent branches over subformulas of it will be privileged. This strategy is good if there are irrelevant formulas, for it helps to avoid using them for the branching heuristics.

To see that, consider the sequent

$$p, (p \to q) \vee (p \to r), ((p \wedge (q \vee r)) \to s) \vee (p \wedge (q \vee r) \to t), (x \wedge y) \to (q \vee r) \vdash s \vee t$$

which generates, with initially $S = \emptyset$, the initial tableau

1. $T\ p$
2. $T (p \to q) \vee (p \to r)$
3. $T (p \wedge (q \vee r) \to s) \vee (p \wedge (q \vee r) \to t)$
4. $T (x \wedge y) \to (q \vee r)$
5. $F\ s \vee t$

Note that line 4 is totally irrelevant to the proof, and we want to avoid using it. After the first expansion of line 5 into

6. $F\ s$ $(F\vee)5$
7. $F\ t$ $(F\vee)5$

we have a choice of lines 2, 3 and 4 over which to apply the branching heuristics. As all those lines are T-marked, we choose a formula that has some atoms in common with the F-marked formulas in tableau; this is justifiable, for the F-marked formulas are those we are trying to prove, and so we choose formulas that are relevant to the goal.

This choice leads to a branch over 3, with branching formulas $F\ p \wedge (q \vee r) \to s$ and $T\ p \wedge (q \vee r) \to s$. The left-hand branch develops as follows:

8a. $F\ p \wedge (q \vee r) \to s$
9a. $T\ p \wedge (q \vee r) \to t$ $(T\vee)$ 3, 8a $S := \{p, q, r, s\}$
10a. $T\ p \wedge (q \vee r)$ $(F \to)$ 8a
11a. $T\ t$ $(T \to)$ 9a, 10a
$\times$

In line 9a the use of $(T\vee)$ using 3 as main premise and 8a as auxiliary premise forces the insertion of all atoms of 8a into S. Since $S = S^F_\wedge = S^T_\vee = S^T_\to$, this new S allows the use of $(T \to)$ to obtain line 12a, which closes the branch with line 7.

On the right-hand branch, we obtain the following expansion:

8b. $T\ p \wedge (q \vee r) \to s$
9b. $F\ p \wedge (q \vee r)$ $(T \to)$ 8b, 6
10b. $F (q \vee r)$ $(F\wedge)$ 9b, 1
11b. $F\ q$ $(F\vee)$ 10b
12b. $F\ r$ $(F\vee)$ 10b

The fact that $q, r \in S$ licenses the use on $(T \rightarrow)$ in line 9b; similarly, $p \in S$ licenses $(F \wedge)$ in line 10b. At this point we have to branch over lines 2 or 4. But branching over line 4 is blocked, for some of its atoms are outside S, which does not occur with line 2. So the expansion proceeds branching over $p \rightarrow q$.

13ba. $T\ p \rightarrow q$	13bb. $F\ p \rightarrow q$
14ba. $T\ q$ $(T \rightarrow)$ 13ba, 1	14bb. $T\ p \rightarrow r$ $(T \vee)$ 2, 13bb
$\times$	15bb. $T\ r$ $(T \rightarrow)$ 14bb, 1
	$\times$

As all branches are closed, the tableau is proved in S_e and also in classical logic.

6. Conclusions and future work

In this paper, we have extended the system S_3^* [7] to deal with full propositional logic, obtaining a family of paracomplete logics which is dual to the family of paraconsistent logics S_3. Comparing the semantics and proof methods of both systems, we noted that the idea behind those systems, namely restricting the application of a rule, could be further generalised. This generalisation gave us the system S_e, for which we gave a semantic and a sound and complete proof method. We then showed how S_e can be used to formalise different heuristics used for theorem proving.

Future work includes extending the implementation of the theorem prover for KE and KES_3 to KES_e and testing it extensively with different context sets.

References

[1] A. Anderson, N. Belnap, Entailment: The Logic of Relevance and Necessity, Vol. 1, Princeton University Press, NJ, 1975.
[2] K. Broda, M. Finger, KE-tableaux for a fragment of linear logic, in: Proc. Fourth Internat. Workshop on Analytic Tableaux and Related Methods, Koblenz, 1995.
[3] K. Broda, M. Finger, A. Russo, Labelled natural deduction for substructural logics, Logic J. IGPL 7 (3) (1999) 283–318.
[4] M. Cadoli, Semantical and computational aspects of Horn approximations, in: Proc. 13th Internat. Joint Conf. Artificial Intelligence (IJCAI-93), 1993, pp. 39–44.
[5] M. Cadoli, F. Scarcello, Semantical and computational aspects of Horn approximations, Artif. Intell. 119 (1–2) (2000) 1–17.
[6] M. Cadoli, M. Schaerf, Approximation in concept description languages, in: Proc. Third Internat. Conf. on Principles of Knowledge Representation and Reasoning (KR92), 1992, pp. 330–341.
[7] M. Cadoli, M. Schaerf, The complexity of entailment in propositional multivalued logics, Ann. Math. Artif. Intell. 18 (1) (1996) 29–50.
[8] C. Chang, R. Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press, London, 1973.
[9] S. Chopra, R. Parikh, R. Wassermann, Approximate belief revision, Logic J. IGPL 9 (6) (2001) 755–768.
[10] M. D'Agostino, Are tableaux an improvement on truth-tables?—cut-free proofs and bivalence, J. Logic Language Inform. 1 (1992) 235–252.
[11] M. D'Agostino, D. Gabbay, A generalization of analytic deduction via labelled tableaux, part I: basic substructural logics, J. Automated Reason. 13 (1994) 243–281.
[12] M. Dalal, Anytime families of tractable propositional reasoners, in: Internat. Symp. of Artificial Intelligence and Mathematics AI/MATH-96, 1996, pp. 42–45.
[13] M. Dalal, Semantics of an anytime family of reasoners, in: 12th European Conference on Artificial Intelligence, 1996, pp. 360–364.
[14] W. Dias, Tableaux implementation for approximate reasoning (in portuguese), Master's Thesis, Department of Computer Science, University of São Paulo, 2002.
[15] M. Finger, Polynomial approximations of full propositional logic via limited bivalence, in: Ninth European Conference on Logics in Artificial Intelligence (JELIA 2004), Lecture Notes in Artificial Intelligence, Vol. 3229, Springer, Lisbon, Portugal, 2004, pp. 526–538.
[16] M. Finger, Towards polynomial approximations of full propositional logic, in: A.L.C. Bazzan, S. Labidi (Eds.), XVII Brazilian Symposium on Artificial Intelligence (SBIA 2004), Lecture Notes in Artificial Intelligence, Vol. 3171, Springer, Berlin, 2004, pp. 11–20.
[17] M. Finger, R. Wassermann, Approximate reasoning and paraconsistency, in: Eighth Workshop on Logic, Language, Information and Computation (WoLLIC'2001), 2001, pp. 77–86.
[18] M. Finger, R. Wassermann, Tableaux for approximate reasoning, in: L. Bertossi, J. Chomicki, (Eds.), IJCAI-2001 Workshop on Inconsistency in Data and Knowledge, Seattle, 2001, pp. 71–79.
[19] M. Finger, R. Wassermann, Expressivity and control in limited reasoning, in: F. van Harmelen (Ed.), 15th European Conference on Artificial Intelligence (ECAI02), IOS Press, Lyon, France, 2002, pp. 272–276.
[20] M. Finger, R. Wassermann, Logics for approximate reasoning: approximating classical logic "from above", in: XVI Brazilian Symposium on Artificial Intelligence (SBIA'02), Lecture Notes in Artificial Intelligence, Vol. 2507, Springer, Berlin, 2002, pp. 21–30.

[21] M. Finger, R. Wassermann, Approximate and limited reasoning: semantics, proof theory, expressivity and control, J. Logic Comput. 14 (2) (2004) 179–204.
[22] H. Levesque, A logic of implicit and explicit belief, in: Proc. AAAI-84, 1984, pp. 198–202.
[23] F. Massacci, Efficient approximate deduction and an application to computer security, Ph.D. Thesis, Dottorato in Ingegneria Informatica. Università di Roma I "La Sapienza", Dipartimento di Informatica e Sistemistica, 1998. URL⟨http://www.ing.unitn.it/∼massacci/Publications/Papers/mass-98-PHD.ps.gz⟩.
[24] D. McAllester, Truth maintenance, in: Proc. Eighth National Conf. on Artificial Intelligence (AAAI-90), 1990, pp. 1109–1116.
[25] M. Schaerf, M. Cadoli, Tractable reasoning via approximation, Artificial Intelligence 74 (2) (1995) 249–310.
[26] B. Selman, H. Kautz, Knowledge compilation using Horn approximations, in: Proc. AAAI-91, 1991, pp. 904–909.
[27] B. Selman, H. Kautz, Knowledge compilation and theory approximation, J. ACM 43 (2) (1996) 193–224.
[28] R.M. Smullyan, First-Order Logic, Springer, Berlin, 1968.
[29] A. ten Teije, F. van Harmelen, Computing approximate diagnoses by using approximate entailment, in: Proc. of KR'96.