In *Proc. of the 5th Conference of the Italian Association for Artificial Intelligence (AI*IA-97)*, Lecture Notes in Artificial Intelligence, ©Springer Verlag, 1997. To appear

A Proof Theory for Tractable Approximations of Propositional Reasoning

Fabio Massacci

Dipartimento di Informatica e Sistemistica
via Salaria 113, I-00198 Roma, Italy
e-mail: `massacci@dis.uniroma1.it`

Abstract. This paper proposes an uniform framework for the proof theory of tractable approximations of propositional reasoning. The key idea is the introduction of *approximate proofs*. This makes possible the development of an approximating sequent calculus for propositional deduction where proofs can be sound, complete or multi-directional approximations of classical logic. We show how this calculus subsumes existing approaches to approximation such as the $BCP-k$ family of anytime reasoners by Dalal and $S-1, S-3$ entailments by Cadoli and Schaerf.

1 Introduction

The use of logic for modeling and reasoning is widespread in Computer Science and Artificial Intelligence. It ranges from logics for knowledge and belief to action, time, processes, classical or multi-valued logics (e.g. [9–11]).

Yet the practical use of logical deduction is hindered by its computational complexity: propositional deduction is co-NP complete and more expressive logics range from PSPACE to EXPTIME [13,11,18]. A computational barrier that one can smooth with proof tactics, heuristics or even passing from theorem proving to model checking [12] but never really overcome.

To overcome this barrier, approximating algorithms for satisfiability have been proposed from the very beginning [13] and are now an established field (see [2] for an overview). On the contrary, much less research has been carried out for its dual, deduction. As noted by Wang in [26]:

> If we compare calculating with proving... we do not have a clear conception of approximate methods in theorem proving. We possess efficient calculating procedures while with proofs... even in a decidable theory, the decision method is not practically feasible. [...] The concept of approximate proofs, though undeniably of a kind other than approximations in numerical calculations, is not incapable of more exact formulation in term of ... gradual improvements towards a correct proof.

Notice that we should distinguish between logics or deduction for approximate reasoning, such as fuzzy or multi-valued logics [1,10,27] from *approximating deduction*. Those logics aim at modeling uncertainty and imprecise concepts, and are NP-complete too [18]. The idea of approximating deduction stems from an engineering perspective: if finding an exact proof of a classical theorem is too hard, why not looking for an approximate one?

To fill this gap, a number of approximation methods for logical consequence have been developed in AI: knowledge compilation [22], semantical approximations [4,15,16,21] and syntactic anytime reasoners [7,6,8].

Knowledge compilation or "vivification" [7,22] is a preprocessing step to transform a theory into another one for which deduction is faster: Horn theories in [22] and boolean constraints propagation in [7]. So it doesn't exactly match the intuition of approximation as an incremental process. Syntactic approximation [7,6,8] is incremental but its essential use of cut (for the incremental steps of the construction) forces to use knowledge compilation for a constructive formulation. These techniques change the formulae of the theory, and lead to an exponential blow up of the approximating theories [4,7,22].

Logical approximation, developed by [4,21] from the initial work of [15,16], is based on semantical approximation of classical logic by multiple valued logic. It is incremental and can be used for both sound or complete approximations. Although algorithms for incremental entailment exist [21] they are based on model enumeration and do not yield a notion of proof (besides making difficult to re-use past computations).

Beyond the restriction to clausal normal form of these approaches, we still miss the notion of *approximate proof* advocated by Wang.

To provide an answer in this direction a comparison with approximating satisfiability is useful. Satisfiability requires a *witness* which satisfies all constraints. An approximate witness only satisfies part of them. Deduction has the dual notion of *proof* e.g. a proof that a counterexample witness does not exists (with refutational theorem proving).

This paper advocates the introduction of approximate proofs which satisfy only some of the constraints a classical proof must have, thus generating the proof space of Fig. 1. This requires an *approximating calculus* where proofs converge to classical proofs, with increasing precision. Since proof are expected to be sound and complete, this also implies the development of a corresponding *superficial semantics*. The idea of approximate proofs has also been useful for proving properties of Frege systems [3] by using unsound steps in a Frege proof.

We sketch some (desirable) requirements for an approximating calculus:

- **multi-directional:** sound, complete or multi-directional approximations (partly sound and partly complete) must be possible as in Fig 1;
- **incremental:** old approximate proofs can be re-used to get better results and eventually converge to a classical proof;
- **flexible:** the user can set the degree of approximation;
- **direct:** the (sub)formulae of the original theory should be used;
- **almost classical:** each proof should "look-like" a classical proof with the exception of approximating steps;
- **semantically grounded:** there should be a superficial semantics so that each step of approximation is sound and complete with respect to such semantics and the semantics itself should converge to a classical one;

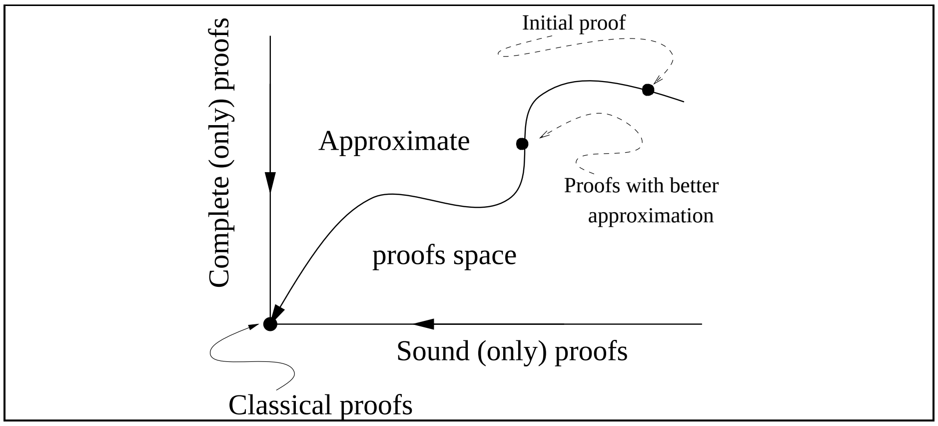

Fig. 1. A Multi-directional Approximate Proofs Space

computationally appealing: approximation should be *more tractable* and it must be possible to use it for an anytime algorithm [20].

The approximating sequent calculus proposed here makes it possible to define multi-directional approximations and can be used to generate a sequence of approximate proof (classical-like) which converge to a classical proof with the performance guarantee required by anytime algorithms[20]. The main idea is to progressively increase the "interesting" (non approximable) elements of a proof.

In what follows, we explain some of the underlying intuitions (§2), introduce some preliminaries (§3) and a sequent calculus for approximate proofs (§4), with a superficial semantics (§5). We show the construction of an anytime deduction procedure (§6) and how other approaches are subsumed (§7).

2 Intuitions

The intuition underlying the sequent calculus we propose is that *not all propositional variables are equal at a first approximation*, developing the ideas and intuition of [4,21] in a proof theoretic setting.

Some atoms describe *interesting* properties and deserve a classical interpretation. They appear in the formulae of an "interesting" (yet approximate) proof or in the truth conditions of a model. Other variables represent *trivial* properties and so are the formula composed with them. They cannot be used for "interesting proofs". Dually other atoms corresponds to *weird* or meaningless properties. They cannot give "interesting models".

For instance suppose that we have a knowledge base representing a circuit [5] where propositional variables of level i are equivalent to a certain formula composed by variables of level $i-1$. We may want to prove a property (a formula) of level i without getting bogged down in the implementation details of level $i-2$, $i-3$, $\ldots 0$. Thus, at a first approximation, we may consider the propositional variables of level lower than $i-1$ as expressing some trivial properties (and so would be the formulae composed only by them) and disregarding them in the search for an approximate proof.

Dually, suppose that we have coded a planning problem into propositional logic and we are looking for a satifiable assignment representing a plan [14]. Again, at a first approximation, we may consider that some properties (and the formulae composed by them) are useless for an interesting plan and therefore regard them as weird: we disregard any model which may require to consider them (i.e. require to give them a truth value).

3 Preliminaries

Propositional formulae, denoted by φ and ψ, are constructed from a set of propositional variables $p \in \mathcal{P}$ with the connectives $\neg$ and $\wedge$. Other connectives are abbreviations e.g. $\varphi \supset \psi \doteq \neg(\varphi \wedge \neg\psi)$. Γ and Δ are sets of formulae. In the sequel by $\mathtt{vars}(\varphi)$ we indicate the set of propositional variables occuring in φ.

Classical logical consequence [23] is denoted by $\Gamma \models \varphi$ and derivability by $\Gamma \vdash \varphi$, while indices represent approximating versions.

Definition 1. A relation $\Gamma \vdash_a \varphi$ is a *sound (complete) approximation* of $\Gamma \models \varphi$ iff for every Γ and φ it is $\Gamma \vdash_a \varphi \;\Rightarrow\; \Gamma \models \varphi$ (resp. $\Gamma \not\vdash_a \varphi \;\Rightarrow\; \Gamma \not\models \varphi$). It is a *multidirectional* one if the above properties depends[1] on Γ and φ.

The intuition is that sound approximations are reliable for yes answers (theorems) whereas complete approximations for no answers (non-theorems).

We are not interested in simply once-off approximation but rather in an *approximating sequence* (of increasing precision). For instance for sound approximations we expect a sequence of $\vdash_a^k$ such that for some n:

$$\Gamma \vdash_a^0 \varphi \;\Rightarrow\; \ldots \Gamma \vdash_a^k \varphi \;\Rightarrow\; \Gamma \vdash_a^{k+1} \varphi \ldots \;\Rightarrow\; \ldots \Gamma \vdash_a^n \varphi \Leftrightarrow \Gamma \models \varphi$$

Approximating sequences should provide performance guarantee so that anytime deduction algorithms can be constructed [20]: each $\vdash_a^k$ should require at most time $t_k = O(2^{ck})$ for some c. So, if an answer (possibly approximate) is desired in time t, then one could choose a k such that $t_k \leq t < t_{k+1}$.

4 An Approximating Sequent Calculus

We define an *approximating partition* (AP for short) of the propositional variables $\mathcal{P}$ into three disjoint sets I, T and W with the intuitive meaning described in §2. By changing the partition we can change the nature and the precision of our approximation. In the sequel a formula φ_T (resp. φ_W) is *completely trivial (weird)* with respect to an approximating partition iff it is composed only by trivial (weird) variables i.e. $\mathtt{vars}(\varphi_T) \subseteq T$ (resp. $\mathtt{vars}(\varphi_W) \subseteq W$).

[1] For multidirectional approximation it is desirable a closure w.r.t. composition with connectives [6] e.g. if it is sound for φ and φ' it should be sound for $\varphi \wedge \varphi'$ or $\neg\varphi$ etc.

Axioms for interesting formulae $\texttt{vars}(\varphi_I) \subseteq I$

$$A_L : \Gamma, \varphi_I, \neg\varphi_I \Longrightarrow \Delta \qquad A_R : \Gamma \Longrightarrow \Delta, \varphi_I, \neg\varphi_I \qquad A_M : \Gamma, \varphi_I \Longrightarrow \Delta, \varphi_I$$

Connectives elimination rules

$$\neg_L : \frac{\Gamma, \varphi \Longrightarrow \Delta}{\Gamma, \neg\neg\varphi \Longrightarrow \Delta} \qquad \neg_R : \frac{\Gamma \Longrightarrow \Delta, \varphi}{\Gamma \Longrightarrow \Delta, \neg\neg\varphi}$$

$$\wedge_L : \frac{\Gamma, \varphi, \psi \Longrightarrow \Delta}{\Gamma, \varphi \wedge \psi \Longrightarrow \Delta} \qquad \wedge_R : \frac{\Gamma \Longrightarrow \Delta, \varphi \qquad \Gamma \Longrightarrow \Delta, \psi}{\Gamma \Longrightarrow \Delta, \varphi \wedge \psi}$$

$$\vee_L : \frac{\Gamma, \neg\varphi \Longrightarrow \Delta \qquad \Gamma, \neg\psi \Longrightarrow \Delta}{\Gamma, \neg(\varphi \wedge \psi) \Longrightarrow \Delta} \qquad \vee_R : \frac{\Gamma \Longrightarrow \Delta, \neg\varphi, \neg\psi}{\Gamma \Longrightarrow \Delta, \neg(\varphi \wedge \psi)}$$

Axiom and deletion for trivial formulae $\texttt{vars}(\varphi_T) \subseteq T$

$$T_L : \frac{\Gamma \Longrightarrow \Delta}{\Gamma, \varphi_T \Longrightarrow \Delta} \qquad T_R : \Gamma \Longrightarrow \Delta, \varphi_T$$

Axiom and deletion for weird formulae $\texttt{vars}(\varphi_W) \subseteq W$

$$W_L : \Gamma, \varphi_W \Longrightarrow \Delta \qquad W_R : \frac{\Gamma \Longrightarrow \Delta}{\Gamma \Longrightarrow \Delta, \varphi_W}$$

Fig. 2. Approximating Sequence Rules

The sequent calculus for approximating deduction is close, except for the approximation steps, to an *analytic symmetric Gentzen system* [23,24].

Thus, given an AP, a *sequent* is a pair $\Gamma \Longrightarrow_{AP} \Delta$ where Γ and Δ are sets of propositional formulae (we may drop AP for simplicity), called respectively the antecedent and the succedent. As usual we abbreviate $\Gamma \cup \{\varphi\}$ with Γ, φ.

A *sequent tree* Π is a binary tree where each node is labelled with a sequent and whenever a parent node is labelled with a sequent, its children must be labelled with the sequents resulting from the application of one rule from Fig. 2. The root will be labelled with the goal sequent we are trying to prove. We say that rule A is applied before rule B when the former labels a node closer to the goal sequent i.e. the root of the tree[2].

We do not use structural rules (such as exchange, contraction or weakening) since they are incorporated directly into the set-oriented formalism.

Notice how rules have been substantially divided into three categories:

- axioms for interesting formulae which matches the classical symmetric rules in Gentzen systems [23];
- rules for the elimination of connectives which can be applied irrespectively of the composition of the formulae and which are also classical;
- rules for deleting formulae or extra axioms which involve only completely trivial or weird formulae.

[2] In this way the terminology itself mirrors closely the process of proof search where we start from the goal and apply rules until we find a proof or a countermodel.

An *approximate proof* is a sequent tree where each leaf is labelled by an axiom sequent. An *open leaf* is labelled by a non axiom sequent to which no rule can be applied. For instance $\phi_T \Longrightarrow$ is not an open leaf because deletion can still be applied. We refer to open leaves as *approximate counter models* for the root sequent. A *branch* is path from the root to a leaf which we call open if the leaf is open and closed if it is labelled by an axiom.

The extra rules for deleting formulae or the extra axioms are indeed the approximations steps of our proofs. For instance using an "extra" axiom corresponds to an complete-only approximate step: we close a branch even if couldn't close it yet by interesting formulae and thus we may get closer to a "proof".

Notice also that *deletion rules* W_R and T_L are not equivalent to thinning, which is just an unrestricted version of the deletion rules [24,23]. Indeed, if thinning is always applied first during the proof search in classical logic, one may not get a proof (we may delete a formula necessary for the proof) whereas this doesn't happen for W_R and T_L (we state this formally in §6,Thm.6).

The interesting features of the calculus come out with formulae with variables of different nature. As an example consider $\Gamma = \{a, a \supset b, b \supset c, c \supset d\}$ and fix $W = \emptyset$, $I = \{a, b, d\}$ and $T = \{c\}$. Then $\Gamma \Longrightarrow_{AP} b$ and $\Gamma \Longrightarrow_{AP} c \wedge \neg c$, yet $\Gamma \not\Longrightarrow_{AP} d$. Modus ponens does not propagate through trivial formulae: at the stage of approximation defined by the AP, they are not interesting for a proof.

It is important to notice that the choice of a symmetric sequent calculus is not simply a matter of taste. The key observation is that formulae never change side of the sequent during a proof. Many of the following results would not hold without this property and the only way to use rules which allow formulae to change side (e.g. for negation) is to restrict them to interesting formulae.

However, the property of "keeping the sign" is also necessary for intuitionistic or nonmonotonic logics. Thus the system proposed here is robust since it is easily adapted e.g. to circumscription [19].

5 A Superficial Semantics

We modify and extend the various propositional semantics proposed in [7,16,21].

Definition 2. Let $W \cdot I \cdot T$ be an AP, a *superficial interpretation* for $W \cdot I \cdot T$ is any function $\mathcal{I}$ from literals to $\{\top, \bot\}$ such that:

1. $\mathcal{I}(p) \neq \mathcal{I}(\neg p)$ iff $p \in I$;
2. $\mathcal{I}(p) = \mathcal{I}(\neg p) = \top$ iff $p \in T$;
3. $\mathcal{I}(p) = \mathcal{I}(\neg p) = \bot$ iff $p \in W$;

This is *not a multi-valued* logic although it looks like Belnap's 4-valued logic [1] with the extra values "unknown" and "over-constrained", or the multi-valued assignment for non-S variables of the $S-3$ entailment in [21]. In Belnap, and in any other multi-valued logic, every variable $p \in P$ can assume every value of the domain: $\{\top, \bot, u, o\}$. Here variables in I can only assume values $\top$ or $\bot$; all variables in T are bound to over constrained and cannot assume other values while W variables are fixed as undefined.

We can extend this interpretation to formulae as follows:

$$\begin{array}{ll}
\mathcal{I} \| \!\!- l & iff\ \mathcal{I}(l) = \top \text{ for all literals } l = p \text{ or } l = \neg p \\
\mathcal{I} \| \!\!- \neg\neg\varphi & iff\ \mathcal{I} \| \!\!- \varphi \\
\mathcal{I} \| \!\!- \varphi \wedge \psi & iff\ \mathcal{I} \| \!\!- \varphi \text{ and } \mathcal{I} \| \!\!- \psi \\
\mathcal{I} \| \!\!- \neg(\varphi \wedge \psi) & iff\ \mathcal{I} \| \!\!- \neg\varphi \text{ or } \mathcal{I} \| \!\!- \neg\psi
\end{array}$$

If $W = T = \emptyset$ then $\mathcal{I}$ is a classical interpretation and for any φ_I s.t. $\texttt{vars}(\varphi_I) \subseteq I$ a superficial interpretation is indistinguishable from a classical one. Logical consequence and satisfiability are introduced in the usual way:

Definition 3. Let $W \cdot I \cdot T$ be an AP, a formula φ is *satisfiable* for $W \cdot I \cdot T$ if there is a superficial interpretation $\mathcal{I}$ such that $\mathcal{I} \| \!\!- \varphi$. It is a *logical consequence* of a set of formulae Γ, i.e. $\Gamma \models_{AP} \varphi$, iff for every superficial interpretation $\mathcal{I}$ if $\mathcal{I} \| \!\!- \bigwedge\!\!\!\bigwedge \Gamma$ then $\mathcal{I} \| \!\!- \varphi$.

The model theoretic interpretation of sequents is also standard: a superficial interpretation $\mathcal{I}$ satisfies a sequent $\Gamma \Longrightarrow \Delta$ iff whenever $\mathcal{I} \| \!\!- \bigwedge\!\!\!\bigwedge \Gamma$ then $\mathcal{I} \| \!\!- \bigvee\!\!\!\bigvee \Delta$.

The following properties have a particular role in the definition of the approximate sequent calculus as they are the semantics analogue of the corresponding axioms and deletion properties:

Proposition 4. *Let $W \cdot I \cdot T$ be an AP, then for any interpretation $\mathcal{I}$ and any completely trivial φ_T, weird φ_W, and interesting φ_I it is:*

- $\mathcal{I} \| \!\!- \varphi_T$ *and* $\mathcal{I} \| \!\!- \neg\varphi_T$
- $\mathcal{I} \| \!\!\not- \varphi_W$ *and* $\mathcal{I} \| \!\!\not- \neg\varphi_W$
- $\mathcal{I} \| \!\!- \neg\varphi_I$ *iff* $\mathcal{I} \| \!\!\not- \varphi_I$.

The soundness and completeness of the calculus with respect to the approximate semantics can be easily established by using proposition 4:

Theorem 5. *Let $W \cdot I \cdot T$ be an AP, then $\Gamma \Longrightarrow_{AP} \varphi$ is provable iff $\Gamma \models_{AP} \varphi$.*

A counter model for the root sequent can be built from an open leaf by setting all literals in Γ to $\top$ and those in Δ to $\bot$.

6 Anytime (Approximate) Deduction

It is well known that the time and size complexity of a sequent proof is worst case exponential in the size of the root sequent [25]. So we can gain from this machinery only if we can

1. discard more formulae than classical proof search will do;
2. anticipate the finding of "classical" proof (or countermodel);
3. arrange the above two steps into an anytime algorithm.

The first property, which also shows that deletion is different from thinning is guaranteed by the following theorem:

Theorem 6. *If $\Gamma \Longrightarrow_{AP} \Delta$ has an approximate proof then it has an approximate proof where deletion rules are the only rules used to reduce weird and trivial formulae. other rules.*

In alternative to deletion we may simply postpone the reduction of the main connective of the corresponding weird or trivial formula and consider as open leaves those where the only unreduced formulae are either weird (in the succedent) or trivial (in the antecedent).

To "anticipate" classical proofs we use the following properties:

Proposition 7. *If $\Gamma \Longrightarrow_{AP} \Delta$ has an approximate sequent proof Π where all leaves are labelled by axioms for interesting formulae then Π is a* classical proof.

Proposition 8. *If $\Gamma \Longrightarrow_{AP} \Delta$ has an approximate sequent tree Π with a path from the root to an open leaf without application of deletion rules then the leaf is a* classical counter model.

Hence, if a branch is closed by an axiom for interesting formulae we can discard it. It is valid for classical proofs and it is of no use for classical counter models. So the *active frontier* of the approximate proof (the part we must store and eventually revise for better approximations) is only made by closures due to axioms for non interesting formulae (rules W_L, T_R) or, for open branches, by the application of deletion (rules W_R, T_L).

We have now the formal machinery to construct an approximating sequence of proofs of increasing precision as in Fig. 1. To decide whether $\Gamma \vdash \varphi$

1. start with an AP choosing W_0, I_0, T_0 so that the set of interesting variables I_0 is small (e.g. logarithmic in the size of Γ and φ);
2. derive an approximate sequent proof Π_k (or counter model) exploiting the trivial/weird axioms and rules to close as many branches as possible or discard as many formulae as possible;
3. if the approximate proof (counter model) is also a classical one then stop;
4. else (if we still have time) change the AP and enlarge I_k to I_{k+1}, visit only the part of Π_k where trivial/weird axioms and rules have been applied since they may no longer be "valid rules", and go to (2) for a new Π_{k+1};

If we time out after Π_k, we have available either an approximate proof (all branches of Π_k are closed) or an approximate counter model (an open leaf of Π_k). On the contrary, with sufficient resources we can always arrive at $I = \mathcal{P}$ and thus derive exactly a classical sequent tree.

Notice that we always re-use older proofs since after (4) we still have an "almost everywhere" correct labeling: the search must continue only where $W_{L,R}$ or $T_{L,R}$ rules have been applied.

From the viewpoint of the computational complexity we can check in polynomial time if a formula is completely weird, trivial or interesting and do *not need branching* for $\wedge, \vee$ of trivial or weird formulae: rather we use deletion rules to delete them (Thm. 6) or the corresponding axioms to close the branch.

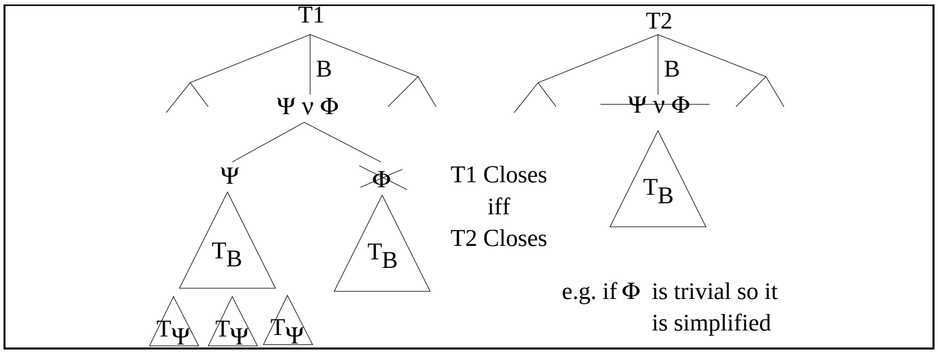

Fig. 3. Cutting the Search Space

A further gain is obtained by *look ahead branch subsumption* as sketched in Fig. 3, where Φ is deleted due to some weird/trivial deletion rule.

Thus one can prove that for an AP *the proof search is (worst case) exponential only in the size*[3] *of interesting formulae.* This means that the cost of finding Π_k is exponential in size the interesting part (according I_k) of $\Gamma \Longrightarrow_{AP_k} \varphi$ while Π_{k+1} is exponential in the size of the I_{k+1} part. If I_0 is logarithmic in the size of the sequent, the first steps will be polynomial and afterwards we have the desired anytime approximating algorithm.

For theoremhood, $\Longrightarrow \varphi$, an approximating sequence of only sound (only complete) proofs is given by setting $T \doteq \emptyset$ (resp. $W \doteq \emptyset$) and progressively increasing I and decreasing W (resp. T).

For logical consequence (usually $|\Gamma| \gg |\varphi|$), an approximating sequence of sound (complete) proofs is built by initializing $\mathtt{vars}(\varphi) \subseteq I$ and $W \doteq \emptyset$ (resp. $T \doteq \emptyset$). Subsequently we can increase I and decrease T (resp. W).

7 Incorporating Other Systems

For the comparison with other systems we restrict our general framework to the clausal normal form (CNF) used in [21,7,6]. Therefore in the sequel we assume that Γ is a set of clauses and $\Delta = \{\varphi\}$ where φ is a clause.

In [21] two approximate entailments are defined, $\models^3_S$ and $\models^1_S$, where S is a set of "interesting" variables. The first corresponds to a sound approximation whereas the second is a complete one. $S-1$ entailment corresponds to our superficial entailment when all variables not in S are weird. To define $S-3$ entailment we need to change the definition of interpretation so that not interesting variables are interpreted with Levesque's 3-valued logic [15] which is close to Belnap's relevance logic [1]. Thus an $S-3$ interpretation $\mathcal{I}^3_S$ maps all interesting literals as a classical interpretation whereas for not interesting literals p, $\neg p$ we

[3] Note that in tree-like analytic sequent calculi as the one designed here we must count each occurrence as a distinct formula [25].

have simply the constraints that at least one of $\mathcal{I}^3_S(p) \neq \perp$ or $\mathcal{I}^3_S(\neg p) \neq \perp$ (see [21] for details). In a nutshell, with $S-3$ entailment not interesting variables can either have a classical interpretation or the same interpretation of trivial variables in a superficial interpretation.

The equivalence with our system is described by the following properties:

Theorem 9. *For every Γ, φ in CNF and set $S \subseteq \mathcal{P}$, the AP_1 with $T \doteq \emptyset$ and $I \doteq S$ is such that $\Gamma \models^1_S \varphi$. iff $\Gamma \Longrightarrow_{AP_1} \varphi$ is provable.*

Theorem 10. *For every Γ, φ in CNF and set $S \subseteq \mathcal{P}$, the AP_3 with $W \doteq \emptyset$ and $I \doteq S \cup \mathtt{vars}(\varphi)$ is such that $\Gamma \models^3_S \varphi$ iff $\Gamma \Longrightarrow_{AP_3} \varphi$ is provable.*

To prove this theorem we use a result in [21, Lemma 5.2], i.e. $\Gamma \models^3_S \varphi$ if and only if $\Gamma \models^3_{S'} \varphi$ where $S' = \mathtt{vars}(\varphi) \cup S$, and then prove the equivalence between AP_3 and the new $\models^3_{S'}$.

Notice the asymmetry between $S-1$ and $S-3$ entailment. Indeed Theorem 10 does not hold if φ is an arbitrary formula (but then $S-3$ entailment is not tractable anymore [21]).

The family $\Gamma \vdash^k_{BCP} \varphi$ of sound approximate reasoners [7] applies boolean constraints propagation (BCP), a variant of unit resolution, as a basic tractable inference and subsequently uses cut on clauses of increasing size $k = 1, 2 \ldots$ [6,7]. So if φ_k is a clause with *at most* k literals we have the following rules:

$$BCP : \frac{\Gamma \vdash_{BCP} \varphi}{\Gamma \vdash^k_{BCP} \varphi} \qquad\qquad k-cut : \frac{\Gamma \vdash^k_{BCP} \varphi_k \qquad\qquad \Gamma, \varphi_k \vdash^k_{BCP} \psi}{\Gamma \vdash^k_{BCP} \psi}$$

Since unit resolution is easily translated into an analytic sequent proof, the transformation of a $\vdash^k_{BCP}$ derivation into an approximate proof is simply an application of cut-elimination [24,23]. The AP is then chosen by setting $W \doteq \emptyset$ and I as the set of variables such that the relative literals occurs along the branches corresponding to unit resolution. Dually, for non valid formulae, take for I the literals occurring in the equivalents classes defined by $\vdash^k_{BCP}$ in [7].

Beside this easy equivalence, it is also possible a more interesting construction which (i) is independent from the particular $\vdash^k_{BCP}$ derivation of φ from Γ and, especially, (ii) gives us an example of a possible construction of the I_k sets in the anytime algorithm described in §6.

$$\begin{aligned} I_k^{(0)} &= \mathtt{vars}(\varphi) \\ I_k^{(n+1)} &= I_k^{(n)} \cup \{\mathtt{vars}(\psi) \mid \psi \in \Gamma \text{ and } |\mathtt{vars}(\psi) \setminus I_k^{(n)}| \leq k+1\} \\ I_k &= \cup_n I_k^{(n)} \end{aligned}$$

Once I_k is computed we do not need to restart from scratch and can simply put $I^{(0)}_{k+1} = I_k$. The preliminary computation of I_k is polynomial in $|\Gamma \cup \{\varphi\}|$.

Theorem 11. *For every Γ, φ in CNF and integer k, the AP_k with $W \doteq \emptyset$ and $I \doteq I_k$ is such that $\Gamma \vdash^k_{BCP} \varphi$ implies $\Gamma \Longrightarrow_{AP_k} \varphi$ is provable.*

8 Conclusions

This paper proposes a logical calculus and a semantics for approximating classical deduction by faster and semantically grounded approximations. It based on the idea of developing a sequence of approximate proofs which converge to a classical proof with the performance of anytime algorithms.

It provides sound, complete or multidirectional approximations and subsumes previous families of approximate reasoners [7,21]. It can be also used as an interactive system for computationally sound proofs [17]: the verifier chooses the "interesting variables" and asks the prover for an approximate proof.

This approach can be directly extended to tableaux for circumscription [19] or modal and first order logic (the extension of weird predicates, and their negation, is empty while trivial ones hold for the whole universe). An different approximation is possible where objects can be impossible or interesting.

However the key question is whether we should accept the idea of trading accuracy for speed in the case of proofs. This could be the case for deductive planning in reactive systems and also for formal verification, where approximate but quick proof can be useful for the design and prototyping phase.

Acknowledgements

I would like to thank L. Carlucci Aiello, F. Pirri and M. Schaerf and an anonymous referee for many useful suggestions. Part of this work has been done while the author was visiting the Computer Laboratory at the University of Cambridge (UK). This work has been partly supported by ASI, CNR and MURST grants.

References

1. A. Anderson and N. Belnap. *Entailment: the Logic of Relevance and Necessity.* Princeton University Press, Princeton, NJ, 1975.
2. G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solution of NP optimization problems. *Theor. Comp. Sci.*, 150(1):1–55, 1995.
3. S. Bellantoni, T. Pitassi, and A. Urquhart. Approximation and small-depth frege proofs. *SIAM J. of Computing*, 21(6):1161–1179, 1992.
4. M. Cadoli. *Tractable Reasoning in Artificial Intelligence*, volume 941 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1995.
5. L. J. Claesen, editor. *Formal VLSI Correctness Verification: VLSI Design Methods*, volume II. Elsevier Sci. Publishers (North-Holland), Amsterdam, 1990.
6. M. Dalal. Anytime families of tractable propositional reasoners. In *International Symposium on Artificial Intelligence and Mathematics AI/MATH-96*, pages 42–45, 1996. Extended version submitted to Annals of Mathematics and Artificial Intelligence.
7. M. Dalal. Semantics of an anytime family of reasoners. In *Proc. of the 12th Eur. Conf. on Artificial Intelligence (ECAI-96)*, pages 360–364. John Wiley & Sons, 1996.

8. M. Dalal and D. W. Etherington. Tractable approximate deduction using limited vocabularies. In *Proceeding of the 9th Canadian Conference on Artificial Intelligence (AI-92)*, pages 206–221, 1992.
9. N. J. Fisher and R. E. Ladner. Propositional dynamic logic of regular programs. *J. of Comp. and System Sci.*, 18:194–211, 1979.
10. R. Hähnle. *Automated Deduction in Multiple-Valued Logics*, volume 10 of *International Series of Monographs on Computer Science*. Oxford University Press, 1994.
11. J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
12. J. Y. Halpern and M. Y. Vardi. Model checking vs. theorem proving: A manifesto. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-91)*, 1991.
13. D. S. Johnson and M. R. Garey. *Computers and Inctractability: A Guide to the Theory of NP-Completeness.* Freeman, S. Francisco, 1979.
14. H. Kautz and B. Selman. Planning as satisfiability. In *Proc. of the 10th Eur. Conf. on Artificial Intelligence (ECAI-92)*, pages 359–363. John Wiley & Sons, 1992.
15. H. J. Levesque. A logic of implicit and explicit belief. In *Proc. of the 4th Nat. Conf. on Artificial Intelligence (AAAI-84)*, pages 198–202, 1984.
16. H. J. Levesque. Logic and the complexity of reasoning. *J. of Philosophical Logic*, 17:355–389, 1988.
17. S. Micali. CS proofs (extended abstract). In *Proc. of the 35th Annual Symp. on the Found. of Comp. Sci. (FOCS-94)*, pages 436–453. IEEE Comp. Society Press, 1994.
18. D. Mundici. Satisfiability in many-valued sentential logic is NP-complete. *Theor. Comp. Sci.*, 52(1-2):145–153, 1987.
19. N. Olivetti. Tableaux and sequent calculus for minimal entailment. *J. of Automated Reasoning*, 9:99–139, 1992.
20. S. Russel and S. Zilberstein. Composing real-time systems. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91)*, pages 212–217, 1991.
21. M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74:1–62, 1995.
22. B. Selman and H. Kautz. Knowlege compilation and theory approximation. *J. of the ACM*, 43(2):193–224, 1996.
23. R. M. Smullyan. *First Order Logic.* Springer-Verlag, 1968. Republished by Dover, New York, in 1995.
24. M. Szabo, editor. *The collected papers of G. Gentzen.* North-Holland Publ. Co., Amsterdam, Amsterdam, 1969.
25. A. Urquhart. The complexity of propositional proofs. *The Bulletin of Symbolic Logic*, 1(4):425–467, Dec 1995.
26. H. Wang. Towards mechanical mathematics. *IBM Journal for Research and Development*, 4:2–22, 1960.
27. M. Ying. A logic for approximate reasoning. *J. of Symbolic Logic*, 59(3):830–837, 1994.