In Mark G. Radle, editor, *Proceedings of the 8th International Conference on Tools with Artificial Intelligence (ICTAI-96)*, pages 308–315.© IEEE Computer Society Press, 1996.

Approximate Reasoning for Contextual Databases*

Fabio Massacci†
Computer Laboratory, University of Cambridge
Pembroke Street, Cambridge CB2 3QG, England
e-mail: Fabio.Massacci@cl.cam.ac.uk

Abstract

Contextual reasoning has been proposed as a tool for solving the problem of generality in AI and for effectively handling huge knowledge bases, while approximate reasoning has been developed to overcome the computational barrier of classical deduction.

This paper combines these approaches to provide an intuitive representation of knowledge and an effective deduction. Its semantics and a tableau calculus are presented. The key computational features are discussed.

1. Introduction

In the last years there has been a renewed interest in the use of contexts within AI. Indeed, the discussion about contextual reasoning in AI can be traced back to McCarthy's Turing Award Lecture in 1971, and has been recently tackled by Shoham [28] and McCarthy [24]. The need to put theories in their contexts is connected to meta-reasoning [7, 29], and [8] discuss the modelling of reasoning as non interacting contexts.

The explicit representation of the context, where a property of a common sense knowledge base holds, or an English sentence is uttered, etc. makes it possible to manage huge knowledge bases by localising deduction [13, 14, 16, 15, 18], and can be a possible solution to the problem of generality in AI [3, 5, 24]. This research has also been underpinned by a practical goal [12]:

> One of our main interest is, in fact, to *provide foundation to the implementation of "intelligent" reasoning systems*.

Thus Multi Language hierarchical logics (ML for short) from Giunchiglia et al. [13, 12, 14] may be seen as the foundation of `FOL` and `GETFOL` systems by [29, 11, 14], whereas the logic of context of Guha [15] and Buvač et al. [5, 4] could play the same role for `Cyc`[1] micro-theories [16, 15].

Yet, the widespread use of contextual reasoning for huge knowledge bases has been hindered by the general computational complexity of reasoning. For instance, ML logics [12] are PSPACE-complete as modal logics [17] and propositional logic of context [4] is NP-complete [23]. Although logic of context is more tractable than general ML systems (unless NP=PSPACE) it is still intractable.

To turn around this general computational barrier, approximation techniques have been developed for many problems in NP (e.g. see [1, 2, 20, 26, 27]). This has led to a theoretical classification of approximability [2] and an experimental analysis of satisfiability which identifies easy and hard "average" cases [27]. Most of these approaches [2, 27] focus on partial satisfiability (e.g. find an assignment for at least 90% of the clauses) and [20, 26] base their analysis of entailment on the underlying satisfiability problem.

In the quest for intelligent *and* effective applications a question is important:

Question 1 *Can approximation be applied to contextual reasoning to make it more effective in practice?*

The basic intuitions behind our proposal of approximate contextual databases can be explained with an example of (human) reasoning due to [6]:

> We are consulting an encyclopaedia about animals, looking for a particular information, let's say whether cows have molar teeth or not. It is natural to start our research from the paragraph about cows, then to broaden the scope more and more, taking into account the section on herbivore, the chapter on mammals the volume on vertebrates, ...During this process it may happen that

*Thanks to prof. L. Carlucci Aiello for her encouragement and support; to the AI group in Rome, the MRG group at IRST, the participants at the AAAI Fall symposium and the anonymous ICTAI referees for useful comments who helped to improve this work.

†On leave from Dip. Informatica e Sistemistica, Univ. di Roma "La Sapienza".

[1] The research program on `Cyc` has lead to a spin-off corporation, Cycorp. See the web page `http://www.cyc.com`.

> we find a term that we do not understand, because it is defined in part we did not read. Anyway we continue until we are able to answer the question or we run out of resources (time, motivation, ...) We are performing two different form of local reasoning: 1) only a part of the knowledge base is taken into account and 2) only a part of the language is taken into account.

Traditional contextual reasoning [4, 13, 14, 15, 24] tackles local reasoning of type (1): structure the knowledge base and represent explicitly the context in which each fact or whatsoever is going to be used. At run-time we extract the part we need [3, 14]. Local reasoning of type (2) is captured by logical approximation as proposed by [6, 20, 26]: restrict the language so that the part of the knowledge base which is not "meaningful" can be ignored during the query.

Intuitively we can say that type (1) reasoning corresponds to *contextualize at compile time* whereas type (2) corresponds to *contextualize at run time*.

In this paper we show how we can merge these two approaches simply by developing the concept of a *context-dependent vocabulary* and propose a tableaux calculus based on labelled deduction as in [21, 23] to boost deduction to a greater extent. In this way one can use contextual databases to represent intuitively commonsense knowledge and still use reasonably effective deduction mechanism. Indeed we show that using approximate reasoning may lead to an exponential speed-up of the deduction process.

In the next section we present the intuitions behind approximate contextual reasoning and a flavour of its features. In Sect. 3 we introduce syntax and semantics and give a tableaux calculus in Sect. 4. How to speed-up its proofs and the corresponding computational pay-off are discussed in Sect. 5. The setting of a vocabulary is discussed in Sect. 6 and conclusion are drawn in Sect. 7.

2. Intuitions

To represent properties of contexts we use the propositional logic of contexts introduced by [15] (see also [4, 24] for a discussion and the motivations underlying the logic of context). We build formulae from propositional letters with traditional propositional connectives enhanced by the contextual modality $\mathtt{ist}\,(c, \varphi)$, where c is a context and ϕ is a formula. Intuitively $\mathtt{ist}\,(c, \varphi)$ means that property φ **is t**rue in context c. Contexts can be themselves context dependent and we allows for arbitrary levels of nesting [4, 15, 24].

For instance we can represent the fact that in US politics gen. Powell (once a potential candidate to US presidency) is a republican (and hence regarded a conservative), yet republicans do not think he is enough not leftist, as follows:

$$\begin{array}{l}
\mathtt{ist}\,(US, politician(P))\\
\mathtt{ist}\,(US, \mathtt{ist}\,(politics, gen(P) \wedge \neg candid(P)))\\
\mathtt{ist}\,(US, \mathtt{ist}\,(politics, \neg cons(P) \vee \neg left(P)))\\
\mathtt{ist}\,(US, \mathtt{ist}\,(politics, cons(P)))\\
\mathtt{ist}\,(US, \mathtt{ist}\,(politics, \neg\mathtt{ist}\,(republican, \neg left(P))))
\end{array}$$

Deduction method for this logic can be Hilbert style as in [5, 4], natural deduction with tactics as in [15] or the tableaux method as in [22, 23].

The basic idea of the tableaux calculus is that a context is a set of interpretations and thus we simply need (i) to give a name to each interpretation [23] and (ii) use labelled deduction as in [9, 21, 23]. The intuition behind labelled deduction is that we label formulae with model theoretic information (the context, the names of interpretations) and use rules which take into account both formulae and labels.

For instance, suppose that we want to prove that P is a leftist within the context of $republican$ which is a sub-context of $politics$, itself a sub-context of US (we represent nesting of contexts by simple sequences such as $\langle US, politics\rangle$). To construct the proof, we start assuming that there is interpretation e.g. $\langle US, politics\rangle\,[1]$ where the formula $\neg\mathtt{ist}\,(republican, \neg left(P))$ holds:

$$\langle US, politics\rangle\,[1] : \neg\mathtt{ist}\,(republican, \neg left(P))$$

and by breaking down connectives we can derive that

$$\langle US, politics, republican\rangle\,[2] : left(P)$$

Tableaux rules reason about the model: suppose a certain property $\vec{c}[n] : \varphi$ holds for the sequence of contexts $\vec{c}$, then another property $\vec{c'}[n'] : \varphi'$ should hold for another (or maybe the same) contexts sequence $\vec{c'}$ etc. If the initial formula is not satisfiable then we always arrive to contradictory requirements (see also [23]).

To represent local reasoning of type (2) we use the concept of *vocabulary* of a user. The intuition is that not all atomic sentences are equally important in every context (at least according a particular user). On the contrary a user may be willing to spend exponential resources on meaningful sentences, which are in his/her vocabulary, and get rid has soon as possible of meaningless ones (outside the vocabulary). At the same time we can have mixed sentences and we may not want to discard a whole formula when because we do not understand a small part of it.

For instance for the context $\langle US, politics\rangle$, we may not be interested whether Powell is a general or not but only whether he is a leftist or not:

$$\mathtt{vocab}(\langle US, politics\rangle) = \{left(P), cons(P)\}$$

If we found out sentences such as

$$\begin{array}{l}
\langle US, politics\rangle\,[1] : gen(P) \vee \neg gen(P)\\
\langle US, politics\rangle\,[1] : gen(P) \wedge \neg gen(P)
\end{array}$$

these are *both completely meaningless sentences* for our vocabulary. Thus we want them to be *both satisfiable* (we assume database designers were making sense) and *both non valid* (after all we do not understand them).

This is exactly what happens with common-sense reasoning (see the example in the introduction): when we do not understand something we just ignore it and go ahead, concentrating on meaningful sentences.

To capture this behaviour with a firmer theoretical framework we use superficial valuations, extending [19, 26]. Intuitively, if a propositional letter p is not in the vocabulary of a context c i.e.$p \notin \mathtt{vocab}(c)$ then for every superficial valuation σ it is $\sigma(p) = \sigma(\neg p) = \mathtt{true}$. Only if $p \in \mathtt{vocab}(c)$ we have $\sigma(p) \neq \sigma(\neg p)$ i.e.only meaningful sentences "deserve" a 2-valued interpretation. Thus we still recognise the following pair as contradictory:

$$\langle US, politics\rangle\ [1] : left(P) \wedge \neg left(P)$$

A vocabulary has also been used in [4, 5, 15, 22] with the different objective of constructing partially defined contexts rather than tractability. Still, since they are often based based on multiple valued logic, there is no computational gain (which is the major objective here).

The differences with multi valued logics are worth noticing, since they, as classical logic, "give the same chances" to all propositional variables: the truth values that a particular propositional letter can assume are not restricted a priori; all propositional letter can assume all n (or infinite) truth values. Therefore they are not sensible to the context. For instance Kripke/Kleene 3-valued logic does not have propositional tautologies. In contrast, superficial valuations still has tautologies, although restricted to meaningful sentences: the set of tautologies varies according the context (or better its vocabulary).

In a nutshell, if we only deal with meaningful sentences, we have the properties of the "good old fashioned" classical logic but we can also reason about formulae mixing meaningful with meaningless sentences. We can interpret *meaningless sentences as noise in the communication*. If it is not too much we can still make sensible deductions. We have a *sound approximation* of classical contextual reasoning: if we find that a query (approximately) follows from a KB then it is also derivable with a classical calculus [4, 15, 23] where the vocabulary contains all atoms of the language.

This provide us with a tool to speed up the computation. The major intuition is that as soon as we met a completely meaningless sentence (no matter how big or how complicated) we do *not* start breaking its connectives to reduce it at atomic level as classical tableaux calculi do. We just write it off at once. With efficient data structures we can "get rid" of non relevant contexts and meaningless sentences in linear time and thus spend the bulk of resources (at worst exponential) only on relevant part.

3. Superficial semantics

We assume some basic knowledge of the *propositional logic of context* by [4, 15, 24]. If $\mathcal{P}$ is a set of propositional letters p, $\mathcal{C}$ a set of contexts c (eventually indexed), then our formulae φ, ψ are

$$\varphi, \psi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathtt{ist}\,(c, \varphi)$$

Other connectives can be seen as abbreviations. In the sequel l is a literal and $\vec{c}_n$ is a sequences of contexts $\langle c_1, \ldots, c_n\rangle$. The sequence $\vec{c}_n \circ c_A$ is a shortcut for the concatenation of $\vec{c}_n$ and $\langle c_A\rangle$. A sequence $\vec{c}_n$ *extends* a sequence $\vec{c}_i$ if there is a $\vec{c}_j$ such that $\vec{c}_n = \vec{c}_i \circ \vec{c}_j$.

A user can define a *vocabulary*, i.e.a function $\mathtt{vocab}() : \mathcal{C}^* \to 2^{\mathcal{P}}$ from sequences of contexts to sets of propositional letters. As already noted, only the propositional letters $p \in \mathtt{vocab}(\vec{c})$ will receive a careful evaluation (i.e.a 2-valuation) whereas if p is not in $\mathtt{vocab}(\vec{c})$ then it is meaningless and a superficial evaluation will be fair enough. In the sequel we use the words *"classical" contextual reasoning* to denote the case when $\mathtt{vocab}()$ is just the set of all propositional letter; so that in practice we have not superficial valuations but only classical ones [4, 23, 24].

Definition 1 *Let $V \subseteq \mathcal{P}$ be a set of propositional letters, then a* superficial valuation *w.r.t. V is a function $\sigma_V : \mathcal{P} \cup \neg\mathcal{P} \to \{\mathtt{true}, \mathtt{false}\}$ such that*

- $\sigma_V(p) \neq \sigma_V(\neg p)$ *for any $p \in V$,*
- $\sigma_V(p) = \sigma_V(\neg p) = \mathtt{true}$ *for any $p \in \mathcal{P}/V$.*

In the sequel we indicate with S_V the set of all superficial valuations σ_V w.r.t. V. Intuitively $\mathtt{vocab}(\vec{c})$ gives us the vocabulary (the set of meaningful sentences) of $\vec{c}$ whereas $\Sigma(\vec{c}) \subseteq S_{\mathtt{vocab}(\vec{c})}$ reflects the (superficial) truth valuations which describe that particular $\vec{c}$.We drop the subscript $\mathtt{vocab}(\vec{c}_n)$ from σ, assuming that whenever $\sigma \in \Sigma(\vec{c}_n)$ then σ is a $\sigma_{\mathtt{vocab}(\vec{c}_n)}$.

The semantics is similar to Kripke models [4, 22]:

Definition 2 *A* superficial layered model *is a triple $\langle W, \Sigma, \mathtt{vocab}()\rangle$ where W is a set of superficial valuations (possible worlds) w.r.t. the vocabulary $\mathtt{vocab}()$ and Σ a function which maps sequences of contexts $\vec{c}_n$ into subsets of W.*

The term "layered" has been used since a sequence of contexts can be seen as a sequence of layers, each build up from a set of valuations. For instance Fig. 1 shows a possible model for our example of gen. Powell.

Here we have $W = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6\}$ and, for instance $\Sigma(\langle US, politics\rangle) = \{\sigma_3\}$ while $\sigma_6(leftist(P)) = \mathtt{true}$. The layer US corresponds to the possible worlds,

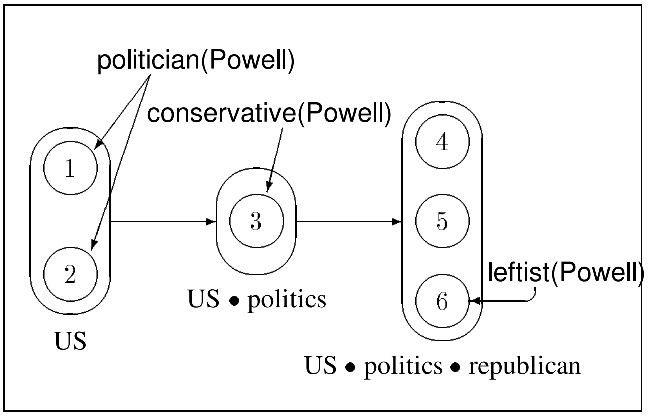

Figure 1. A layered model

which (in this model) characterise the context US. Intuitively each possible world corresponds to a state of affairs within the context of US. In our example we assumed $\mathtt{ist}\,(US, politician(P))$ and thus we intuitively expect that in all possible interpretation within US the atomic sentence $politician(P)$ should be true. Formally this corresponds to say that $\sigma_j(politician(P)) = \mathtt{true}$ for $j = 1, 2$. Note also that, unlike traditional Kripke models [8, 9, 17], we do *not* have an accessibility relation between single individual worlds. We only consider relations between contexts as sets of possible worlds (see [23] for the implications in term of computational complexity and expressiveness.).

Next we extend superficial valuations to formulae:

Definition 3 *Let $\langle W, \Sigma, \mathtt{vocab}(\,)\rangle$ be a layered model, $\vec{c}_n$ a sequence of contexts and $\sigma \in \Sigma(\vec{c}_n)$ a valuation, then the entailment relation $\vec{c}_n, \sigma \models \varphi$ is such that:*

- $\vec{c}_n, \sigma \models l$ *iff* $\sigma(l) = \mathtt{true}$ *where* $l = p$ *or* $l = \neg p$
- $\vec{c}_n, \sigma \models \varphi \wedge \psi$ *iff* $\vec{c}_n, \sigma \models \varphi$ *and* $\vec{c}_n, \sigma \models \psi$
- $\vec{c}_n, \sigma \models \neg(\varphi \wedge \psi)$ *iff* $\vec{c}_n, \sigma \models \neg\varphi$ *or* $\vec{c}_n, \sigma \models \neg\psi$
- $\vec{c}_n, \sigma \models \varphi \vee \psi$ *iff* $\vec{c}_n, \sigma \models \varphi$ *or* $\vec{c}_n, \sigma \models \psi$
- $\vec{c}_n, \sigma \models \neg(\varphi \vee \psi)$ *iff* $\vec{c}_n, \sigma \models \neg\varphi$ *and* $\vec{c}_n, \sigma \models \neg\psi$
- $\vec{c}_n, \sigma \models \neg\neg\varphi$ *iff* $\vec{c}_n, \sigma \models \varphi$
- $\vec{c}_n, \sigma \models \mathtt{ist}\,(c, \varphi)$ *iff for every* $\sigma^* \in \Sigma(\vec{c}_n \circ c)$ *it is* $\vec{c}_n \circ c, \sigma^* \models \varphi$
- $\vec{c}_n, \sigma \models \neg\mathtt{ist}\,(c, \varphi)$ *iff there exists* $\sigma^* \in \Sigma(\vec{c}_n \circ c)$ *such that* $\vec{c}_n \circ c, \sigma^* \models \neg\varphi$

The classical property $\vec{c}_n, \sigma \models \neg\varphi$ iff $\vec{c}_n, \sigma \not\models \varphi$ *does not hold* since φ may contain meaningless sentences.

Two other concepts are used for logical consequence:

- the *Local Contextual Database* LB contains the formulae holding in the *query context* $\vec{c}_q$;
- the *Global Contextual Database* GB contains the formulae holding in every contexts sequence $\vec{c}_n$ extending the query context $\vec{c}_q$.

Here we follow [4] and assume that each query is asked in a context $\vec{c}_q$ (determined by the system or the user). There may be more than one LB if we have many query contexts $\vec{c}_q$. In a simplified version we can assume it to be an "outer" context [4] $\langle\rangle$ where all deduction are carried and ignore it. Recalling the intuition of Sect. 3, GB and LB corresponds to knowledge contextualized at compile time (probably fixed) whereas the vocabulary corresponds to contextualizing at run-time and it may change as we change the formula φ we are testing for satisfiability (see Sect. 6).

Definition 4 *A formula φ is* satisfiable *for the contexts sequence $\vec{c}_q$ w.r.t. the local and global contextual database, LB and GB, and the vocabulary $\mathtt{vocab}()$, iff there is a layered model $\langle W, \Sigma, \mathtt{vocab}(\,)\rangle$ s.t.*

- *for every $\vec{c}_n$ extending $\vec{c}_q$ and every $\sigma \in \Sigma(\vec{c}_n)$ it is $\vec{c}_n, \sigma \models GB$*
- *for some $\sigma \in \Sigma(\vec{c}_q)$ it is $\vec{c}_q, \sigma \models LB$ and $\vec{c}_q, \sigma \models \varphi$.*

Where $\vec{c}_n, \sigma \models KB$ denotes $\forall\psi \in KB : \vec{c}_n, \sigma \models \psi$.

Definition 5 *A formula φ is* superficially valid *for the contexts sequence $\vec{c}_q$ w.r.t.the local and global contextual database LB and GB and the vocabulary $\mathtt{vocab}(\,)$iff $\neg\varphi$ is not satisfiable.*

Note that superficial validity is explicitly defined in term of lack of counterexamples, in the spirit of superficial entailment: a sentence φ should be (superficially) valid if we cannot provide an evidence of the contrary i.e.if we cannot find a model where $\neg\varphi$ holds.

In classical logic one defines φ as valid iff it holds in every model and then shows that if φ holds everywhere then one cannot find a model where $\neg\varphi$ holds and vice versa. This is not possible here, because meaningless formulae do not satisfies this property. For instance, consider a propositional letter p and a context $\vec{c}_n$ such that $p \notin \mathtt{vocab}(\vec{c}_n)$. Then, for every superficial valuation $\sigma \in \Sigma(\vec{c}_n)$ it is *both* $\vec{c}_n, \sigma \models p$ and $\vec{c}_n, \sigma \models \neg p$. Still, we would not like to define p as "valid". To obtain the desired behaviour we explicitly require the absence of counterexamples. With this slight but intuitive modification of validity, meaningless sentences are always satisfiable but never valid.

The interplay of meaningful and meaningless sentences can be better explained by some examples. For this we introduce completely meaningless formulae, which are formulae where not even a single literal is meaningful.

Definition 6 *A formula ψ is* completely meaningless *(CML) for* $\texttt{vocab}()/\vec{c}_n$ *iff it has the form*

- p *where* $p \notin \texttt{vocab}(\vec{c}_n)$*;*
- $\neg\psi_1$ *or* $\psi_1 \wedge \psi_2$ *where* ψ_1*,* ψ_2 *are CML for* $\texttt{vocab}()/\vec{c}_n$*;*
- $\texttt{ist}\,(c, \psi_1)$ *where* ψ_1 *is CML for* $\texttt{vocab}()/\vec{c}_n \circ c$*.*

In a dual way we can define *completely meaningful formulae* by changing the first condition in $p \in \texttt{vocab}(\vec{c}_n)$: all atoms occurring in the formula are meaningful.

Then it is easy to prove the following properties:

Proposition 1 *Let* ψ *be CML for* $\texttt{vocab}()/\vec{c}_q$*, then*

1. ψ *is satisfiable but not superficially valid;*
2. $\varphi \wedge \psi$ *is satisfiable iff* φ *is satisfiable.*
3. $\varphi \wedge \psi$ *is never superficially valid;*
4. $\varphi \vee \psi$ *(i.e.* $\neg(\neg\varphi \wedge \neg\psi)$*) is satisfiable;*
5. $\varphi \vee \psi$ *is superficially valid iff* φ *is superficially valid.*

Meaningless formulae rules out immediately one case from $\varphi \vee \psi$ and we can focus on the meaningful part. Since disjunction is the main source of complexity for tableau methods [25], it is clear that computational gains will be at hand.

Another properties (also present in [6, 20, 26]) is that *modus ponens may and may not hold*. For instance

Fact 1 p_m *is superficially valid for* $LB = \{p_1, \neg p_1 \vee p_2, \ldots \neg p_{m-1} \vee p_m\}$ *and context* $\vec{c}_q$ *and* $\texttt{vocab}()$ *iff all* p_i *are meaningful i.e.* $p_i \in \texttt{vocab}(\vec{c}_q)$*.*

This property corresponds to commonsense intuitions: we do not want to run in a long chain of deduction involving meaningless sentences for the user.

4. Approximate tableaux

The tableaux calculus – see [10] for an introduction – is based on prefixed tableaux [9, 21, 23].

For the logic of context, the labels must capture two semantical information: the sequence of contexts and the possible world. So, a *contextualized formula* is a pair $\langle \vec{c}_n[m] : \varphi \rangle$ where $\vec{c}_n$ is a sequence of contexts, m an integer and φ a formula. Intuitively, a *contextual prefix* $\vec{c}_n[m]$ "names" the m-th superficial valuation of $\Sigma(\vec{c}_n)$, where φ holds. In Fig. 1 we have

$$\langle US, politics \rangle [3] : conservative(Powell)$$

The definition of tableau is standard: a *tableau* $\mathcal{T}$ is a binary tree, whose nodes are labelled with contextualized formulae, and a *branch* $\mathcal{B}$ is a path from the root to a leaf.

$$\alpha : \frac{\vec{c}_n[m] : \varphi \wedge \psi}{\begin{array}{c}\vec{c}_n[m] : \varphi \\ \vec{c}_n[m] : \psi\end{array}} \qquad \text{dn} : \frac{\vec{c}_n[m] : \neg\neg\varphi}{\vec{c}_n[m] : \varphi}$$

$$\beta : \frac{\vec{c}_n[m] : \neg(\varphi \wedge \psi)}{\vec{c}_n\,[m] : \neg\varphi \quad | \quad \vec{c}_n[m] : \neg\psi}$$

$$\text{Pos-enter} : \frac{\vec{c}_n[m] : \texttt{ist}\,(c, \varphi)}{\vec{c}_n \circ c[k] : \varphi} \quad \vec{c}_n \circ c[k] \text{ is present}$$

$$\text{Neg-enter} : \frac{\vec{c}_n[m] : \neg\texttt{ist}\,(c, \varphi)}{\vec{c}_n \circ c[k] : \neg\varphi} \quad \vec{c}_n \circ c[k] \text{ is new}$$

$$\text{Loc} \;:\; \frac{\vdots}{\vec{c}_q[0] : \psi} \quad \text{if } \psi \in LB,$$

$$\text{Glob} \;:\; \frac{\vdots}{\vec{c}_n[m] : \psi} \quad \begin{array}{l}\text{if } \psi \in GB, \vec{c}_n[m] \text{ present} \\ \text{and extend } \vec{c}_q\end{array}$$

Figure 2. Tableaux rules

Tableaux rules (Fig. 2) transform a tree into another tree by adding nodes or branching the tree. Tableaux are based on refutational theorem proving: break down connectives and search for contradictions. If all possible choices lead to a contradiction, then the initial formula is unsatisfiable (and hence its negation superficially valid). In the sequel we always denote the query context with $\vec{c}_q$.

The rules for propositional connectives are standard, whereas the entering rules are the "truly" contextual ones. They require some terminology i.e. $\vec{c}_n[m]$ is *present in a branch* if there is a contextualized formula $\langle \vec{c}_n[m] : \varphi \rangle$ already in the branch, whereas it is *new for a branch* if it is not present. Intuitively, we enter into a context to perform some further deduction there.

The closure rule differs from classical tableaux[23]: only contradictions of meaningful sentences matter. Meaningless sentences may be contradictory but…who cares? After all they are meaningless!

Definition 7 (Approx Closure) *A* branch *is* closed *if two contextualized literals* $\langle \vec{c}_n[m] : p \rangle$ *and* $\langle \vec{c}_n[m] : \neg p \rangle$ *are both present in the branch for some* $\vec{c}_n[m]$ *and some* meaningful $p \in \texttt{vocab}(\vec{c}_n)$.

As usual a branch is *completed* if all rules have been applied. It is *open* if it is completed and not closed. A *tableau* is closed if all branch are closed. It is open if at least one branch is open.

Definition 8 *A* approximate validity proof *for the formula* φ *for the query context* $\vec{c}_q$ *is the closed tableau starting with*

$\langle \vec{c}_q[0] : \neg\varphi \rangle$. *In symbol* $GB \vdash \vec{c}_q : LB \to \varphi$. *A* satisfiability proof *for φ is an open branch of the tableau starting with $\langle \vec{c}_q[0] : \varphi \rangle$.*

Intuitively, to prove that φ is valid we try to construct a counter model and if we fail (the tableau closes) we conclude that it is valid.

Completeness and correctness of the calculus can be proved as in [9, 21, 23]: apply all possible rule and use an open branch to construct a layered model. The construction used in [23] for classical valuations can be extended to superficial valuations.

Theorem 2 (Soundness and Completeness) *A formula φ has an approximate validity proof w.r.t. the global database GB, the local database LB and the vocabulary* `vocab( )` *iff it is superficially valid.*

Approximated tableaux proof are *a sound approximation* of classical contextual reasoning [4, 15, 23]:

Theorem 3 (Sound Approx) *If formula φ has a approximated validity proof w.r.t. the vocabulary* `vocab()` *then it has a classical proof.*

The theorem follows from the simple observation that the closure rule for approximated tableau is also valid for classical contextual tableau [23], where the vocabulary is just the set of all propositions.

We can "trust" the approximate calculus on `YES` answers: if a certain query is approximately entailed by the KB than this would have been the case also if we used classical contextual reasoning.

5. Speed-up and computational pay-off

At this stage one may wonder where the computational pay-off is, since it seems that we have just a tableaux for modal logics (only more complicated by the vocabulary) and thus in PSPACE [17].

On the contrary it has been shown[23] that a semantics based on layers rather than a traditional tree-like Kripke structure has computational implications already for classical contextual reasoning:

Theorem 4 ([23]) *The satisfiability problem for classical logic of context is NP-complete.*

This result can be extended to superficial tableaux and thus we are in NP: we are not worse that propositional logics; but also not better since NP is "less" intractable than PSPACE, but not tractable.

Still we can use the contextual structure of the KB to simplify the deduction process as in [3, 14, 15].

Definition 9 *A KB is* contextually organised *if it as the form*

$$KB = \bigwedge_{c_i} \mathtt{ist}\,(c_i, \varphi_i) \wedge \bigwedge_{c_j} \neg\mathtt{ist}\,(c_j, \neg\varphi_j)$$

where each φ is either a disjunction or is contextually organised.

The computational pay-off of contextualizing KB at compile-time can be captured by the following

Theorem 5 (Compile-time pay-off) *If LB and GB are consistent and contextually organised then to prove that $GB \vdash \vec{c}_q : LB \to \mathtt{ist}\,(c, \varphi)$ one can ignore from LB all $\mathtt{ist}\,(c', \psi)$ (sub)formulae with $c \neq c'$ and from GB all $\mathtt{ist}\,(c'', \psi)$ (sub)formulae where c'' does not occur in $\mathtt{ist}\,(c, \varphi)$.*

Before starting a deduction we can reduce GB and LB to GB' and LB'. If the database is well organised and the overlap between different contexts small this can be already a substantial gain.

Another computational payoff can be obtained at run-time by a smarter use of the vocabulary i.e. by *anticipating the outcome of the reduction of a formula, without actually reducing it*. The key point is that, as expressed by Proposition,1, we already know that meaningless sentences are satisfiable and therefore we do not need to expand them.

Hence we modify tableaux rules in clever way as shown by the *speed-up rules* in Fig. 3. Intuitively we can describe these rules as follows: whenever we meet a completely meaningless formula we just write it off without expanding it. When `vocab()` is classical i.e. the set of all propositional letters then the condition NOT CML is trivially verified and we fall back on the rules of Fig. 2.

The most important simplification is the last branching rule for $\varphi \vee \psi$, since branching is the source of exponential complexity [25]. It is based on the following observation: whenever we met a disjunction $\vec{c}_n[m] : \varphi \vee \psi \in \mathcal{B}$ we should branch the tableaux and get $\mathcal{B}_1 = \mathcal{B} \cup \{\vec{c}_n[m] : \varphi\}$ and $\mathcal{B}_2 = \mathcal{B} \cup \{\vec{c}_n[m] : \psi\}$. However, suppose that φ is meaningless. This means that at the next iteration it will be simply written off. Thus, in practice, $\mathcal{B}_1$ is equivalent to $\mathcal{B}$. So we have two branches, one of the form $\mathcal{B}$ and one of the form $\mathcal{B}_2 = \mathcal{B} \cup \{\vec{c}_n[m] : \varphi\}$. In this case it is immediate to see that the tableau closes iff the branch $\mathcal{B}_1 \simeq \mathcal{B}$ closes. Thus, if φ is meaningless, there is no need to reduce $\varphi \vee \psi$ and we can simply write it off, *even if ψ is meaningful.*

By induction on the structure of formulae we have

Theorem 6 *A formula φ has an approximate validity proof with speed-up rules iff it has an approximate validity proof with original tableaux rules and approx closure.*

In practice we simply add (or cut away) the part of the tableau (including much of unnecessary branching) that regards completely meaningless formulae. The point is that if $\vec{c}_n[m] : \varphi$ and $\vec{c}_n[m] : \neg\varphi$ are both in $\mathcal{B}$ and φ is completely *meaningful* for $\mathtt{vocab}()/\vec{c}_n$ then close the branch $\mathcal{B}$;

else-if $\vec{c}_n[m] : \varphi \wedge \psi$ is in $\mathcal{B}$ then

if φ is NOT completely meaningless for $\mathtt{vocab}()/\vec{c}_n$ then add $\vec{c}_n[m] : \varphi$ to $\mathcal{B}$;

if ψ is NOT CML for $\mathtt{vocab}()/\vec{c}_n$ then add $\vec{c}_n[m] : \psi$ to $\mathcal{B}$;

else-if $\vec{c}_n[m] : \neg\neg\varphi$ is in $\mathcal{B}$ then

if φ is NOT CML for $\mathtt{vocab}()/\vec{c}_n$ then add $\vec{c}_n[m] : \varphi$ to $\mathcal{B}$;

else-if $\vec{c}_n[m] : \mathtt{ist}\,(c, \varphi)$ is in $\mathcal{B}$ and $\vec{c}_n \circ c[m']$ is in $\mathcal{B}$ then

if φ is NOT CML for $\mathtt{vocab}()/\vec{c}_n \circ c$ then add $\vec{c}_n \circ c[m'] : \varphi$ to $\mathcal{B}$;

else-if $\vec{c}_n[m] : \neg\mathtt{ist}\,(c, \varphi)$ is in $\mathcal{B}$ then

if φ is NOT CML for $\mathtt{vocab}()/\vec{c}_n \circ c$ then add $\vec{c}_n \circ c[m'] : \neg\varphi$ to $\mathcal{B}$ for a new m';

else-if $\psi \in LB$ then

if ψ is NOT CML for $\mathtt{vocab}()/\vec{c}_q$ then add $\vec{c}_q[0] : \psi$ to $\mathcal{B}$;

else-if $\psi \in GB$ and $\vec{c}_n[m]$ is in $\mathcal{B}$ then

if ψ is NOT CML for $\mathtt{vocab}()/\vec{c}_n$ then add $\vec{c}_n[m] : \psi$ to $\mathcal{B}$;

else-if $\vec{c}_n[m] : \neg(\varphi \wedge \psi)$ is in $\mathcal{B}$ then

if both $\neg\varphi$ and $\neg\psi$ are NOT CML for $\mathtt{vocab}()/\vec{c}_n$ then
branch $\mathcal{B}$ in $\mathcal{B}_1$ and $\mathcal{B}_2$;
add $\vec{c}_n[m] : \neg\varphi$ to $\mathcal{B}_1$;
add $\vec{c}_n[m] : \neg\psi$ to $\mathcal{B}_2$;

Figure 3. Speed-up rules

they cannot close any branch, by Def. 7, so a validity proof is determined only by the meaningful literals (or completely meaningful sentences).

Then it is easy to modify the proof of computational complexity for the tableaux calculus (e.g. [25])::

Theorem 7 (Run-time payoff) *An approximate validity proof with speed-up rules is (at most) exponential* only *in the number of meaningful literals.*

An important point is the cost of determining whether a formula is CML. In the worst case we must read each formula completely and for each propositional letter check whether it is in the vocabulary. The first part is linear in the size of the KB and the second takes constant time if the size of the vocabulary is small w.r.t. the KB (which is likely to be) or if hash functions are used. We can "get rid" of meaningless sentences in linear time. If inefficient data structures are used (e.g. a tree of nested contexts and for each context a list of meaningful propositional letters) with linear search, a search takes at most $O(|C|^2 \times |P|)$ where C is the number of contexts and P the number of propositional letters in the database. It is still polynomial in the size of the KB.

Hence the combination of approximate reasoning and contextual database, beyond a better representation of knowledge, meets our initial goal: exponential resources needs to be spent only for the relevant part of the KB, and only polynomial time (possibly linear) on the rest.

6 Setting a vocabulary

One of the key issues of this approach is setting up a vocabulary. Once again, the intuition is that a vocabulary corresponds to a contextualization at run-time so we expect the system (or the user) to change vocabulary when the query is changed (at least as a possibility).

The simplest way to select meaningful literal for each $\vec{c}_n$ is at *random*. This relays on a characterisation of NP as the set of problems checkable with a logarithmic number of random bits [1]. So if P_c is the number of propositional letter under the scope of $\mathtt{ist}\,(c, \_)$ the system can select randomly $\log(P_c)$ of them and carry a proof accordingly.

Another possibility is *query-linking* which is similar to the techniques used in [3, 14] for the generation of the problem solving context in GETFOL. We add to the vocabulary only the propositional letters that occurs in the query (under the appropriate scope of $\mathtt{ist}\,(c, \_)$). We can use *iterated query linking*: at first we add the literals in the query, and recursively we add the literals which occur in a clause where a meaningful literal already occurs. This process can be stopped when certain depth (prefixed or user dependent) is reached or a fix-point is found. These constructions are polynomial in the size of the KB.

If the theorem prover is interactive, then $\mathtt{vocab}()$ can also be constructed *interactively*. The system initialises it in some way (maybe one of the above) and starts a proof. If a proof is found, then by Thm 3 and 6 we can stop. If no proof is found, the system may look back at a step where a speed-up rule has written off a formula $\varphi \vee \psi$ but only one disjunct was CML. Then it asks the users which other literals (e.g. in ψ) can be considered meaningful. This process may be repeated until a proof is found.

The design of a vocabulary also involves other issues which are related to the application domain. For instance the functional dependency of $\mathtt{vocab}()$ w.r.t. $\vec{c}_n$ can be:

specialising: $\mathtt{vocab}(\vec{c}_n) \subseteq \mathtt{vocab}(\vec{c}_n \circ c_A)$,

generalising: $\mathtt{vocab}(\vec{c}_n \circ c_A) \subseteq \mathtt{vocab}(\vec{c}_n)$,

additive: $\mathtt{vocab}(\langle c_1, \ldots, c_n \rangle) = \bigcup_{i=1}^{n} \mathtt{vocab}(\langle c_i \rangle)$,

flat: $\mathtt{vocab}(\vec{c}_n \circ c_A) = \mathtt{vocab}(\langle c_A \rangle)$,

constant: $\mathtt{vocab}(\vec{c}_n) = \mathtt{Vocab}$.

For instance one can choose the option of generalizing, if context $\vec{c}_n$ is interpreted as more general (and therefore more powerful) that the subcontext $\vec{c}_n \circ c_A$. An opposite view is taken by specialising, where the specialised context has more knowledge (and therefore more meaningful letters) than the general one. These choices are application dependent but only affects the polynomial part of the computation.

7. Conclusions

There has been a number of proposals to formalise contextual reasoning [7, 4, 13, 14, 15, 24]; many of them also motivated by the attempt to provide a formal framework for already existing or developing systems such as `GETFOL` [29, 11, 14] and `Cyc` [16, 15, 18].

In this setting one of the mayor focus of the contextual reasoning area has been the development of commonsense knowledge bases which could be able to solve query answering quickly [3, 14, 15, 18].

At the same time the computational complexity of reasoning has led to various proposal of logical approximation [6, 19, 20, 26]. In this field of research the focus has been on approximation of classical reasoning to be incremental, semantically grounded and computational appealing.

In this paper we have combined these two approaches and show the advantages of such a tool for the development of intelligent *and* effective databases.

With approximate contextual reasoning we gain both from an epistemic viewpoint and from a computational perspective: an user is provided with rules representing the properties (entering into a context, database assumptions etc.) s/he may be interested to model; contextualized formulae makes it simpler to build a counter model for non valid formulae; superficial valuations describe contexts with a partial vocabulary and can be used to reason about formulae containing both meaningful and meaningless sentences. From a computational viewpoint it provide an exponential speed up w.r.t. classical logic and provide a sound approximation of classical contextual reasoning.

Future research is in the direction of extending this system with an `is-a` relation to incorporate object oriented features and solve in more expressive ways the problem of generality.

References

[1] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. In *Proc. of FOCS-92*, pages 2–13, 1992.

[2] G. Ausiello, P. Crescenzi, and M. Protasi. Approximate solution of NP optimization problems. *TCS*, 150(1):1–55, 1995.

[3] P. Bouquet and F. Giunchiglia. Reasoning about theory adequacy. a new solution to the qualification problem. *Fund. Inf.*, 23(2-3-4), 1995.

[4] S. Buvač, V. Buvač, and I. Mason. Metamathematics of contexts. *Fund. Inf.*, 23(3):263–301, 1995.

[5] S. Buvač and I. Mason. Propositional logic of context. In *Proc. of AAAI-93*, pages 412–419, 1993.

[6] M. Cadoli. *Tractable Reasoning in Artificial Intelligence*, *LNAI* 941. Springer-Verlag, 1995.

[7] L. Carlucci Aiello and G. Levi. The uses of metaknowledge in AI systems. In *Proc. of ECAI-84*, pages 705–717, 1984.

[8] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[9] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, 1983.

[10] M. Fitting. *First Order Logic and Automated Theorem Proving*. Springer-Verlag, 1990.

[11] F. Giunchiglia. `GETFOL` user manual, vers. 1.0. Tech. Rep. 9204-01, DIST, Univ. of Genova, Italy, 1992.

[12] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics or: How can we do without modal logics. *AIJ*, 65:29–70, 1994.

[13] F. Giunchiglia, L. Serafini, E. Giunchiglia, and M. Frixione. Non-omniscient belief as context-based reasoning. In *Proc. of IJCAI-93*. pages 548–554, 1993.

[14] F. Giunchiglia and P. Traverso. A multi-context architecture for formalizing complex reasoning. *Int. J. of Intelligent Systems*, 10:501–539, 1995.

[15] R. Guha. *Contexts: A Formalization and Some Applications*. PhD thesis, STAN-CS-91-1399, Stanford Univ., 1991.

[16] R. Guha and D. Lenat. `Cyc`: A midterm report. *AI Magazine*, 11(3):32–59, 1990.

[17] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *AIJ*, 54:319–379, 1992.

[18] D. Lenat. `Cyc`: A large-scale investment in knowledge infrastructure. *CACM*, 38(11), Nov. 1995.

[19] H. Levesque. A logic of implicit and explicit belief. In *Proc. of AAAI-84*, pages 198–202, 1984.

[20] H. Levesque. Logic and the complexity of reasoning. *J. of Philosophical Logic*, 17:355–389, 1988.

[21] F. Massacci. Strongly analytic tableaux for normal modal logics. In *Proc. of CADE-94*, *LNAI* 814, pages 723–737, 1994.

[22] F. Massacci. Superficial tableaux for contextual reasoning. In *Proc. of the AAAI-95 Fall Symp. on "Formalizing Context"*, pages 60–66, 1995.

[23] F. Massacci. Contextual reasoning is NP-complete. In *Proc. of AAAI-96*, 1996. To appear.

[24] J. McCarthy. Notes on formalizing context. In *Proc. of IJCAI-93*, pages 555–560. 1993.

[25] N. Murray and E. Rosenthal. On the computational intractability of analytic tableau methods. *Bulletin of the IGPL*, 2(2):205–228. Max-Plank Institute, Imperial College, 1994.

[26] M. Schaerf and M. Cadoli. Tractable reasoning via approximation. *AIJ*, 74:1–62, 1995.

[27] B. Selman. Stochastic search and phase transitions: AI meets physics. In *Proc. of IJCAI-95*, pages 998–1001, 1995.

[28] Y. Shoham. Varieties of context. In *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 393–407. Academic Press, 1991.

[29] R. Weyhrauch. Prolegomena to a theory of mechanized formal reasoning. *AIJ*, 13(1):133–176, 1980.