

Logics for Approximate Reasoning: Approximating Classical Logic “From Above”

Marcelo Finger and Renata Wassermann

Department of Computer Science
Institute of Mathematics and Statistics
University of São Paulo, Brazil
`{mfinger,renata}@ime.usp.br`

Abstract. Approximations are used for dealing with problems that are hard, usually NP-hard or coNP-hard. In this paper we describe the notion of approximating classical logic from above and from below, and concentrate in the first. We present the family s_1 of logics, and show it performs approximation of classical logic from above. The family s_1 can be used for disproving formulas (the SAT-problem) in a local way, concentrating only on the relevant part of a large set of formulas.

1 Introduction

Logic has been used in several areas of Artificial Intelligence as a tool for representing knowledge as well as a tool for problem solving. One of the main criticism to the use of logic as a tool for automatic problem solving refers to the computational complexity of logical problems. Even if we restrict ourselves to classical propositional logic, deciding whether a set of formulas logically implies a certain formula is a co-NP-complete problem [GJ79]. Another problem comes from the inadequacy of modelling real agents as logical beings. Ideal, logically omniscient agents know all the consequences of their beliefs. However, real agents are *limited* in their capabilities.

Cadoli and Schaerf have proposed the use of approximate entailment as a way of reaching at least partial results when solving a problem completely would be too expensive [SC95]. 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 [SC95], 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.

The approximations are based on the idea of a context set S of atoms. The atoms in S are the only ones whose consistency is taken into account in the process of verifying whether a given formula is entailed by a set of formulas. As we increase the size of the context set S , we get closer to classical entailment, but the computational complexity also increases.

Cadoli and Schaerf proposed two systems, intending to approximate classical entailment from two ends. The S_3 family approximates classical logic from below, in the following sense. Let \mathcal{P} be a set of propositions and $S^0 \subseteq S^1 \subseteq \dots \subseteq \mathcal{P}$; let $Th(L)$ indicate the set of theorems of a logic. Then:

$$Th(S_3(\emptyset)) \subseteq Th(S_3(S^0)) \subseteq Th(S_3(S^1)) \subseteq \dots \subseteq Th(S_3(\mathcal{P})) = Th(CL)$$

where CL is classical logic (in Section 3 this notion is extended to the entailment relation \models).

Approximating a classical logic from below is useful for efficient theorem proving. Conversely, approximating classical logic from above is useful for *disproving theorems*, which is the satisfiability (SAT) problem.

Unfortunately, Cadoli and Schaerf's other system, S_1 , does not approximate classical logic from above, as we will see in Section 3. In this paper, we study the family of logical entailments s_1 , which are approximations of classical logic from above. While S_1 only deals with formulas in negation normal form, s_1 covers full propositional logic.

The family of logic s_1 also tackles the problem of non-locality in S_1 , which implies that S_1 approximations do not concentrate on the *relevant formulas*. Discussions on locality are found in Section 5.

This paper proceeds as follows: in the next section, we briefly present Cadoli and Schaerf's work on approximate entailment. In Section 3 we present the notion of approximation that we are aiming at and show why Cadoli and Schaerf's system S_1 does not approximate classical logic from above. In Section 4 we present our system s_1 and in Section 5 some examples of its behaviour.

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. The letters S and s will denote sets of propositional letters.

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_p is a function $v_p : \mathcal{P} \rightarrow \{0, 1\}$.

2 Approximate Entailment

We briefly present here the notion of *approximate entailment* and summarise the main results obtained in [SC95].

Schaerf and Cadoli define two approximations of classical entailment: \models_S^1 which is complete but not sound, and \models_S^3 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. In the trivial extreme of approximate entailment, i.e., when $S = \mathcal{P}$, classical entailment is obtained. At the

other extreme, when $S = \emptyset$, \models_S^1 holds for any two formulas (i.e., for all α, β , we have $\alpha \models_S^1 \beta$) and \models_S^3 corresponds to Levesque's logic for explicit beliefs [Lev84], which bears a connection to relevance logics such as those of Anderson and Belnap [AB75].

In an S_1 assignment, if $p \in S$, then p and $\neg p$ are given opposite truth values, while if $p \notin S$, both p and $\neg p$ get value 0. 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 names S_1 and S_3 come from the possible truth assignments for literals outside S . If $p \notin S$, there is only one S_1 assignment for p and $\neg p$, the one which makes them both false. 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. Satisfiability, entailment, and validity are defined in the usual way.

The following examples illustrate the use of approximate entailment. Since \models_S^3 is sound but incomplete, it can be used to approximate \models , i.e., if for some S we have that $B \models_S^3 \alpha$, then $B \models \alpha$. On the other hand, \models_S^1 is unsound but complete, and can be used for approximating $\not\models$, i.e., if for some S we have that $B \not\models_S^1 \alpha$, then $B \not\models \alpha$.

Example 1. ([SC95]) We want to check whether $B \models \alpha$, where $\alpha = \neg \text{cow} \vee \text{molar-teeth}$ and

$$B = \{ \neg \text{cow} \vee \text{grass-eater}, \neg \text{dog} \vee \text{carnivore}, \\ \neg \text{grass-eater} \vee \neg \text{canine-teeth}, \neg \text{carnivore} \vee \text{mammal}, \\ \neg \text{mammal} \vee \text{canine-teeth} \vee \text{molar-teeth}, \\ \neg \text{grass-eater} \vee \text{mammal}, \neg \text{mammal} \vee \text{vertebrate}, \\ \neg \text{vertebrate} \vee \text{animal} \}.$$

Using the S_3 -semantic defined above, we can see that for $S = \{\text{grass-eater}, \text{mammal}, \text{canine-teeth}\}$, we have that $B \models_S^3 \alpha$, hence $B \models \alpha$.

Example 2. ([SC95]) We want to check whether $B \not\models \beta$, where $\beta = \neg \text{child} \vee \text{pensioner}$ and

$$B = \{ \neg \text{person} \vee \text{child} \vee \text{youngster} \vee \text{adult} \vee \text{senior}, \\ \neg \text{adult} \vee \text{student} \vee \text{worker} \vee \text{unemployed}, \\ \neg \text{pensioner} \vee \text{senior}, \neg \text{youngster} \vee \text{student} \vee \text{worker}, \\ \neg \text{senior} \vee \text{pensioner} \vee \text{worker}, \neg \text{pensioner} \vee \neg \text{student}, \\ \neg \text{student} \vee \text{child} \vee \text{youngster} \vee \text{adult}, \\ \neg \text{pensioner} \vee \neg \text{worker} \}.$$

Using the S_1 -semantic above, for $S = \{\text{child}, \text{worker}, \text{pensioner}\}$, we have that $B \not\models_S^1 \beta$, and hence $B \not\models \beta$.

Note that in both examples above, S is a small part of the language. Schaerf and Cadoli obtain the following results for approximate inference.

Theorem 1 ([SC95]). *There exists an algorithm for deciding if $B \models_S^3 \alpha$ and deciding $B \models_S^1 \alpha$ which runs in $O(|B| \cdot |\alpha| \cdot 2^{|S|})$ time.*

The result above depends on a polynomial time satisfiability algorithm for belief bases and formulas in clausal form alone. This result has been extended in [CS95] for formulas in negation normal form, but is not extendable to formulas in arbitrary forms [CS96].

3 The Notion of Approximation

The notion of approximation proposed by Cadoli and Schaerf can be described in the following way. Let $\models_S^3: 2^{\mathcal{L}} \times \mathcal{L}$ be the entailment relation in the logic $S_3(S)$, that is, the member of the family of logics S_3 determined by parameter S . Then, we had the following property. For

$$\emptyset \subseteq S' \subseteq S'' \subseteq \dots \subseteq S'^n \subseteq \mathcal{P}$$

we have that

$$\models_{\emptyset}^3 \subseteq \models_{S'}^3 \subseteq \dots \subseteq \models_{S'^n}^3 \subseteq \models_{\mathcal{P}}^3 = \models_{\text{cl}}$$

where \models_{cl} is classical entailment, and hence this was justifiably called an *approximation of classical logic from below*.

A family of logics that approximates classical logic from below is useful for theorem proving. For in such case, if a $B \models_S^3 \alpha$ in logic $S_3(S)$, then we know that classically $B \models \alpha$. So if it is more efficient to do theorem proving in $S_3(S)$, we may prove some classical theorems at a “reduced cost” as theorem proving is a coNP-complete problem. If we fail to prove a theorem in $S_3(S)$, however, we don’t know its classical status; it may be provable in $S_3(S')$ for some $S' \supset S$, or it may be that classically $B \not\models \alpha$. The method for theorem proving in S_3 presented in [FW01] had the advantage of providing an incremental method of theorem proving; that is, if we failed to prove $B \models_S^3 \alpha$, a method was provided for incrementing S and continuing the proof without restarting the proof.

Besides the potential economy in theorem proving, logic $S_3(S)$, by means of its parameter S gives us a clear notion of what propositional symbols are *relevant* for the proof of $B \models \alpha$.

Similarly, we say that a family of parameterised logics $L(S)$ is an *approximation of classical logic from above* if we have:

$$\models_{\emptyset}^L \supseteq \models_{S'}^L \supseteq \dots \supseteq \models_{S'^n}^L \supseteq \models_{\mathcal{P}}^L = \models_{\text{cl}}$$

In a dual way, a family of logics that approximates classical logic from above is useful for *disproving* theorems. That is, if we show that $B \not\models_S^L \alpha$ then we classically know that $B \not\models \alpha$, with the advantage of disproving a theorem at a reduced cost, for the problem in classical logic is the SAT-problem, and therefore NP-complete. Similarly, the parameter S gives us a clear notion of what propositional symbols are *relevant* for disproving a theorem (i.e. for satisfying its negation).

Unfortunately, S_1 does not approximate classical logic from above. In fact, if S_1 approximated classical logic from above, one would expect any classical

theorem to be a theorem of $S_1(S)$ for any S . However, the formula $p \vee \neg p$ is false unless $p \in S$ and hence the logic S_1 does not qualify for an approximation of classical logic from above.

Besides not being an approximation of classical logic from above, there is another limitation in the Cadoli and Schaerf approach which is common to both S_1 and S_3 : The system is restricted to \rightarrow -free formulas and in negation normal form. For the case of S_3 , we have addressed this limitation in [FW01]. We are now going to address this limitation, while also trying to provide a logic that approximates classical logic from above.

Another problem of S_1 is that reasoning within S_1 is not local, at least one literal of each clause must be in S , as it was noted in [tTvH96]. This means that even clauses which are completely irrelevant for disproving the given formula will be examined.

In the next section, we present a system that approximates classical logic without suffering from these limitations.

4 The Family of Logics s_1

The problem of creating a logic that approximates classical logic from above comes from the following fact. Any logic that is defined in terms of a binary valuation $v : \mathcal{L} \rightarrow \{0, 1\}$ that properly extends classical logic is inconsistent. This is very simple to see. If it is a proper extension of classical logic, it will contradict a classical validity. Since it is an extension of classical logic, from this contradiction any formula is derivable.

The way Cadoli and Schaerf avoided this problem was not to make its binary valuation a full extension of classical logic. Here, we take a different approach, for we want to construct an extension of classical entailment, and define a ternary valuation, that is, we define a valuation $v_s^1(\alpha) \subseteq \{0, 1\}$; later we show that $v_s^1(\alpha) \neq \emptyset$.

For that, consider the full language of classical logic based on a set of proposition symbols \mathcal{P} . We define the family of logics $s_1(s)$, parameterised by the set $s \subseteq \mathcal{P}$. Let α be a formula and let $prop(\alpha)$ be the set of propositional symbols occurring in α . We say that $\alpha \in s$ iff $prop(\alpha) \subseteq s$.

Let v_p be a classical propositional valuation. Starting from v_p , we build an s_1 -valuation $v_s^1 : \mathcal{L} \rightarrow 2^{\{0,1\}}$, by defining when $1 \in v_s^1(\alpha)$ and when $0 \in v_s^1(\alpha)$. This definition is parameterised by the set $s \subseteq \mathcal{P}$ in the following way. Initially, for propositional symbols, v_s^1 extends v_p :

$$\begin{aligned} 0 \in v_s^1(p) &\Leftrightarrow v_p(p) = 0 \\ 1 \in v_s^1(p) &\Leftrightarrow v_p(p) = 1 \text{ or } p \notin s \end{aligned}$$

That is, v_s^1 extends v_p but whenever we have an atom $p \notin s$, $1 \in v_s^1(p)$; if $p \notin s$ and $v_p(p) = 0$, we get $v_s^1(p) = \{0, 1\}$. The rest of the definition of v_s^1 proceeds in the same spirit, as follows:

$$\begin{aligned}
0 \in v_s^1(\neg\alpha) &\Leftrightarrow 1 \in v_s^1(\alpha) \\
0 \in v_s^1(\alpha \wedge \beta) &\Leftrightarrow 0 \in v_s^1(\alpha) \text{ or } 0 \in v_s^1(\beta) \\
0 \in v_s^1(\alpha \vee \beta) &\Leftrightarrow 0 \in v_s^1(\alpha) \text{ and } 0 \in v_s^1(\beta) \\
0 \in v_s^1(\alpha \rightarrow \beta) &\Leftrightarrow 1 \in v_s^1(\alpha) \text{ and } 0 \in v_s^1(\beta) \\
\\
1 \in v_s^1(\neg\alpha) &\Leftrightarrow 0 \in v_s^1(\alpha) \quad \text{or } \neg\alpha \notin s \\
1 \in v_s^1(\alpha \wedge \beta) &\Leftrightarrow 1 \in v_s^1(\alpha) \text{ and } 1 \in v_s^1(\beta) \text{ or } \alpha \wedge \beta \notin s \\
1 \in v_s^1(\alpha \vee \beta) &\Leftrightarrow 1 \in v_s^1(\alpha) \text{ or } 1 \in v_s^1(\beta) \text{ or } \alpha \vee \beta \notin s \\
1 \in v_s^1(\alpha \rightarrow \beta) &\Leftrightarrow 0 \in v_s^1(\alpha) \text{ or } 1 \in v_s^1(\beta) \text{ or } \alpha \rightarrow \beta \notin s
\end{aligned}$$

We start pointing out two basic properties of v_s^1 , namely that is a ternary relation and that $1 \in v_s^1(\alpha)$ whenever $\alpha \notin s$.

Lemma 1. *Let α be any formula. Then*

- (a) $v_s^1(\alpha) \neq \emptyset$.
- (b) *If $\alpha \notin s$ then $1 \in v_s^1(\alpha)$.*

Proof. Let α be any formula. Then:

- (a) First note that for any propositional symbol, $v_p(p) \in v_s^1(p)$, so $v_s^1(p) \neq \emptyset$. Then a simple structural induction on α shows that $v_s^1(\alpha) \neq \emptyset$.
- (b) Straight from the definition of v_s^1 . □

It is interesting to see that in one extreme, i.e., when $s = \emptyset$, s_1 -valuations trivialise, assigning the value 1 to every formula in the language. When $s = \mathcal{P}$, s_1 -valuations over the connectives correspond to Kleene's semantics for three valued logics [Kle38].

The next important property of v_s^1 is that it is an extension of classical logic in the following sense. Let v_s^1 be an s_1 -valuation; its *underlying propositional valuation*, v_p is given by

$$\begin{aligned}
v_p(p) &= 0, 0 \in v_s^1(p) \\
v_p(p) &= 1, 0 \notin v_s^1(p)
\end{aligned}$$

as can be inspected from definition of v_s^1 . Also note that v_p and s uniquely define v_s^1 .

Lemma 2. *Let $v_c : \mathcal{L} \rightarrow \{0, 1\}$ be a classical binary valuation extending v_p . Then, for every formula α , $v_c(\alpha) \in v_s^1(\alpha)$.*

Proof. By structural induction on α . It suffices to note that the property is valid for $p \in \mathcal{P}$. Then a simple inspection of the definition of v_s^1 gives us the inductive cases. □

Just note that Lemma 2 implies Lemma 1(a). We can also say that if $\alpha \in s$, then v_s^1 behaves *classically* in the following sense.

Lemma 3. *Let v_p be a propositional valuation and let v_s^1 and v_c be, respectively, its $s_1(s)$ and classical extensions. If $\alpha \in s$, $v_s^1(\alpha) = \{v_c(\alpha)\}$.*

Proof. A simple inspection of the definition of v_s^1 shows that if $\alpha \in s$, v_s^1 behaves classically. \square

Finally, we compare s_1 -valuations under expanding sets s .

Lemma 4. *Suppose $s \subseteq s'$ and let $v_s^1(\alpha)$ and $v_{s'}^1(\alpha)$ extend the same propositional valuation. Then $v_s^1(\alpha) \supseteq v_{s'}^1(\alpha)$.*

Proof. If $\alpha \in s$, $v_s^1(\alpha)$ and $v_{s'}^1(\alpha)$ behave classically. If $\alpha \notin s$, then $1 \in v_s^1(\alpha)$ and we have to analyse what happens when $0 \in v_{s'}^1(\alpha)$. By structural induction on α , we show that $0 \in v_s^1(\alpha)$.

For the base case, just note that v_s^1 and $v_{s'}^1$ have the same underlying propositional valuation.

Consider $0 \in v_{s'}^1(\neg\alpha)$, then $1 \in v_{s'}^1(\alpha)$. Since $\alpha \notin s$, $1 \in v_s^1(\alpha)$, so $0 \in v_s^1(\neg\alpha)$.

Consider $0 \in v_{s'}^1(\alpha \rightarrow \beta)$, then $1 \in v_{s'}^1(\alpha)$ and $0 \in v_{s'}^1(\beta)$. By the induction hypothesis, $0 \in v_s^1(\beta)$. If $\alpha \notin s$, $1 \in v_s^1(\alpha)$ and we are done. If $\alpha \in s$, then also $\alpha \in s'$, $v_s^1(\alpha)$ and $v_{s'}^1(\alpha)$ behave classically and agree with each other, so $1 \in v_s^1(\alpha)$ and we are done.

The cases where $0 \in v_{s'}^1(\alpha \wedge \beta)$ and $0 \in v_{s'}^1(\alpha \vee \beta)$ are straightforward consequences of the induction hypothesis. \square

The next step is to define the notion of a s_1 -entailment.

4.1 s_1 -Entailment

The idea is to define an entailment relation for s_1 , \models_s^1 , parameterised on the set $s \subseteq \mathcal{P}$ so as to extend for any s the classical entailment relation

$$B \models \alpha$$

To achieve that, we have to make valuations applying on the left handside of \models_s^1 to be *stricter* than classical valuations, and the valuations that apply to the right handside of \models_s^1 to be more *relaxed* than classical valuations, for every $s \subseteq \mathcal{P}$. This motivates the following definitions.

Definition 1. *Let $\alpha \in \mathcal{L}$ and let v_s^1 be a s_1 -valuation. Then:*

- *If $v_s^1(\alpha) = \{1\}$ then we say that α is strictly satisfied by v_s^1 .*
- *If $1 \in v_s^1(\alpha)$ then we say that α is relaxedly satisfied by v_s^1 .*

That these definitions are the desired ones follows from the following.

Lemma 5. *Let $\alpha \in \mathcal{L}$. Then:*

- (a) *α is strictly satisfiable implies that α is classically satisfiable.*
- (b) *α is classically satisfiable implies that α is relaxedly satisfiable.*

Proof.

- (a) Consider v_s^1 such that $v_s^1(\alpha) = \{1\}$. Let v_p be its underlying propositional valuation and let v_c be a classical valuation that extends v_p . Since $0 \notin v_s^1(\alpha)$, by Lemma 2 we have that $v_c(\alpha) \neq 0$, so $v_c(\alpha) = 1$.
- (b) Consider a classical valuation v_c such that $v_c(\alpha) = 1$. Let v_p be its underlying propositional valuation. Then directly from Lemma 2, $1 \in v_s^1(\alpha)$. \square

We are now in a position to define the notion of s_1 -entailment.

Definition 2. We say that $\beta_1, \dots, \beta_m \models_s^1 \alpha$ iff all s_1 -valuation v_s^1 that strictly satisfies all β_i , $1 \leq i \leq m$, relaxedly satisfies α .

The following are important properties of s_1 -entailment.

Lemma 6.

- (a) $B \models_{\emptyset}^1 \alpha$, for every $\alpha \in \mathcal{L}$.
- (b) $\models_{\mathcal{P}}^1 = \models_{\text{cl}}$
- (c) If $s \subseteq s'$, $\models_s^1 \supseteq \models_{s'}^1$.

Proof.

- (a) By Lemma 1(b), $1 \in v_{\emptyset}^1(\alpha)$, for every $\alpha \in \mathcal{L}$.
- (b) By Lemma 3, $v_{\mathcal{P}}^1$ is a classical valuation, and the notions of strict, relaxed and classical valuation coincide.
- (c) Suppose $s \subseteq s'$, $B \models_{s'}^1 \alpha$ but $B \not\models_s^1 \alpha$. Then exists v_s^1 such that $v_s^1(\beta_i) = \{1\}$, for all $\beta_i \in B$ but $v_s^1(\alpha) = \{0\}$. Let $v_{s'}^1$ be the s_1 -valuation generated by v_s^1 underlying propositional valuation. From Lemma 4 we have that $v_{s'}^1(\beta_i) = \{1\}$, for all $\beta_i \in B$. Since $B \models_{s'}^1 \alpha$, we have that $1 \in v_{s'}^1(\alpha)$. Again by Lemma 4 we get $1 \in v_s^1(\alpha)$, which contradicts $v_s^1(\alpha) = \{0\}$. So $B \models_s^1 \alpha$. \square

From what has been shown, it follows directly that this notion of entailment is the desired one.

Theorem 2. The family of s_1 -logics approximates classical entailment from above, that is:

$$\models_{\emptyset}^1 \supseteq \models_{s'}^1 \supseteq \dots \supseteq \models_{s'^n}^1 \supseteq \models_{\mathcal{P}}^1 = \models_{\text{cl}}$$

Proof. Directly from Lemma 6. \square

It is interesting to point that if v_s^1 is a s_1 -valuation falsifying $B \models_s^1 \alpha$, we have a classical valuation v_c that falsifies $B \models \alpha$ built as an extension of the propositional valuation v_p such that $v_p(p) = 1 \Leftrightarrow v_s^1(p) = \{1\}$.

One interesting property that fails for s_1 -entailment is the *deduction theorem*. One half of it is still true, namely that

$$B \models_s^1 \alpha \Rightarrow \models_s^1 (\bigwedge B) \rightarrow \alpha$$

However, the converse is not true. Here is a counterexample. Suppose $q \notin s$ and $p \in s$, so $q \rightarrow p \notin s$. Then $\models_s^1 q \rightarrow p$; take a valuation that makes $v_s^1(q) = 1$ and $v_s^1(p) = 0$, hence $q \not\models_s^1 p$.

5 Examples

In this section, we examine some examples and compare s_1 to Cadoli and Schaerf's S_1 . We have already seen that, unlike S_1 entailment, s_1 entailment truly approximates classical entailment from above.

Let us have a look at what happens with Example 2 when we use s_1 entailment:

Example 3 (Example 2 revisited).

We want to check whether $B \not\models \beta$, where $\beta = \neg \text{child} \vee \text{pensioner}$ and

$$B = \{ \neg \text{person} \vee \text{child} \vee \text{youngster} \vee \text{adult} \vee \text{senior}, \\ \neg \text{adult} \vee \text{student} \vee \text{worker} \vee \text{unemployed}, \\ \neg \text{pensioner} \vee \text{senior}, \neg \text{youngster} \vee \text{student} \vee \text{worker}, \\ \neg \text{senior} \vee \text{pensioner} \vee \text{worker}, \neg \text{pensioner} \vee \neg \text{student}, \\ \neg \text{student} \vee \text{child} \vee \text{youngster} \vee \text{adult}, \\ \neg \text{pensioner} \vee \neg \text{worker} \}.$$

It is not difficult to see that with $s = \{\text{child}, \text{pensioner}\}$, we can take a propositional valuation v_p such that $v_p(\text{pensioner}) = 0$ and $v_p(p) = 1$ for p any other propositional letter, such that the s_1 -valuation obtained from v_p strictly satisfies every formula in B but does not relaxedly satisfy β . Hence, we have that $B \not\models_s^1 \beta$, and $B \not\models \beta$.

This example shows that we can obtain an answer to the question of whether $B \not\models \beta$ with a set s smaller than the set S needed for S_1 .

Another concern was the fact that S_1 did not allow for local reasoning. Consider the following example, borrowed from [CPW01]:

Example 4. The following represents beliefs about a young student, Hans.

$$B = \{ \text{student}, \text{student} \rightarrow \text{young}, \text{young} \rightarrow \neg \text{pensioner}, \\ \text{worker}, \text{worker} \rightarrow \neg \text{pensioner}, \\ \text{blue-eyes}, \text{likes-dancing}, \text{six-feet-tall} \}.$$

We want to know whether Hans is a pensioner.

We have seen that in order to use Cadoli and Schaerf's S_1 , we had to start with a set S containing at least one atom of each clause. This means that when we build S , we have to take into account even clauses which are completely irrelevant to the query, as *likes-dancing*.

In our system, formulas not in s will be automatically set to 1. If we have $s = \{\text{pensioner}\}$, a propositional valuation such that $v_p(\text{pensioner}) = 0$ and $v_p(p) = 1$ for p any other propositional letter, can be extended to an s_1 -valuation that strictly satisfies B but does not relaxedly satisfy *pensioner*. Hence, $B \not\models \text{pensioner}$.

It is not difficult to see that, unlike in Cadoli and Schaerf's S_1 and S_3 , the classical equivalences of the connectives hold in s_1 , which means that we do not have any gains in terms of the size of the set s using different equivalent forms of the same knowledge base.

6 Conclusions and Future Work

We have proposed a system for approximate entailment that can be used for approximating classical logic “from above”, in the sense that at each step, we prove less theorems, until we reach classical logic. The system proposed is based on a three-valued valuation and a different notion of entailment, where the logic on the right hand side of the entailment relation does not have to be the same as the logic on the left hand side. This sort of “hybrid” entailment relation has been proposed before in Quasi-Classical Logic [Hun00].

Future work includes the study of the formal relationship between our system and other three-valued semantics and the design of a tableaux proof method for the logic, following the line of [FW01].

References

- [AB75] A.R. Anderson and N.D. Belnap. *Entailment: The Logic of Relevance and Necessity, Vol. 1*. Princeton University Press, 1975.
- [CPW01] Samir Chopra, Rohit Parikh, and Renata Wassermann. Approximate belief revision. *Logic Journal of the IGPL*, 9(6):755–768, 2001.
- [CS95] Marco Cadoli and Marco Schaerf. Approximate inference in default logic and circumscription. *Fundamenta Informaticae*, 23:123–143, 1995.
- [CS96] Marco Cadoli and Marco Schaerf. The complexity of entailment in propositional multivalued logics. *Annals of Mathematics and Artificial Intelligence*, 18(1):29–50, 1996.
- [FW01] Marcelo Finger and Renata Wassermann. Tableaux for approximate reasoning. In Leopoldo Bertossi and Jan Chomicki, editors, *IJCAI-2001 Workshop on Inconsistency in Data and Knowledge*, pages 71–79, Seattle, August 6–10 2001.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.
- [Hun00] A. Hunter. Reasoning with contradictory information in quasi-classical logic. *Journal of Logic and Computation*, 10(5):677–703, 2000.
- [Kle38] S. C. Kleene. On a notation for ordinal numbers. *Journal of Symbolic Logic*, 1938.
- [Lev84] Hector Levesque. A logic of implicit and explicit belief. In *Proceedings of AAAI-84*, 1984.
- [SC95] Marco Schaerf and Marco Cadoli. Tractable reasoning via approximation. *Artificial Intelligence*, 74(2):249–310, 1995.
- [tTvH96] Annette ten Teije and Frank van Harmelen. Computing approximate diagnoses by using approximate entailment. In *Proceedings of KR’96*, 1996.