Non-monotonic reasoning: from complexity to algorithms

C. Cayrol and M.C. Lagasquie-Schiex
IRIT
118 route de Narbonne
31062 Toulouse Cedex
FRANCE
{testemal, lagasq}@irit.fr

T. Schiex
INRA
Chemin de Borde Rouge BP 27
31326 Castanet Tolosan Cedex
FRANCE
Thomas.Schiex@toulouse.inra.fr

Introduction

We focus on coherence-based approaches to non-monotonic reasoning. These approaches handle syntactical belief bases equipped with symbolic ordering structures and rely upon classical logic. The belief base, a finite set of propositional formulae, is considered syntactically, as in (Nebel 1991): each belief is a distinct piece of information and only beliefs which are explicitly present in the base are taken into account. It departs from the logical point of view where a base is identified with the set of its models. The belief base is not assumed consistent. Moreover, we assume that the belief base is equipped with a complete pre-ordering structure (a priority relation) which, contrarily to (Gärdenfors & Makinson 1994), is not related to the semantical entailment ordering. So, the base is stratified in a collection of subbases of different priority levels. Coherence-based non-monotonic entailment from a stratified belief base can be described as a two-step procedure which first restores the coherence by selecting preferred consistent subbases, and then applies classical entailment on some of these preferred subbases according to a so-called entailment principle. The selection of preferred subbases relies upon the definition of aggregation modes which enable to extend the priority ordering (defined on the initial belief base) into a preference relation between subbases. In the framework described above, our purpose is to propose a comparative study of various coherence-based entailment relations from the point of view of computational complexity[1]. As far as we know, only few papers have been devoted to computational complexity issues for non-monotonic reasoning. Syntax-based revision procedures were considered in (Nebel 1991). Abductive procedures and deduction in default logic were considered in (Gottlob 1992).

First, we briefly present the entailment relations and the associated problems which will be considered from the computational complexity point of view. Secondly, we give an informal and simplified presentation of the main concepts of the complexity theory. Then, we provide comparative results. Even in restricted cases such as strictly ordered belief bases or Horn bases, the results seem to be disappointing since the complexity in the worst case remains greater than the complexity of classical entailment unless both restrictions apply simultaneously, an unrealistic restriction. These results inclined us to look for an adapted tool for solving decision problems above NP: we show that Binary Decision Diagrams (Bryant 1992) tools can be extended for solving some classes of coherence-based entailment problems.

[1] Other points of view such as cautiousness and validity of deduction rules have been considered in (Cayrol & Lagasquie-Schiex 1995).

Coherence-based non-monotonic entailment

Let $(E, \leq)$ denote a stratified belief base. It may be equivalently represented by the collection of strata $(E_1, E_2, \ldots, E_n)$ where E_1 contains the formulae of highest priority (or relevance) and E_n those of lowest priority.

Usually, inconsistency of the belief base E is handled through maximal (for set-inclusion) consistent subbases of E. Different approaches have been proposed to use the priority relation in order to select "preferred" consistent subbases (see (Cayrol & Lagasquie-Schiex 1993) for a survey). Here, we focus on two preference relations which refine the set-inclusion: the inclusion-based preference which combines priorities and maximal consistent subbases and the lexicographic preference which combines priorities and subbases of maximal cardinality.

Let X be a consistent subbase of E. $X = X_1 \cup \ldots \cup X_n$, where $X_i = X \cap E_i$.

inclusion-based preference This is the strict partial ordering defined by $X \ll^{\text{incl}} Y$ iff there exists i such that Y_i strictly contains X_i and for any $1 \leq j < i$, $X_j = Y_j$. Inclusion-based preferred subbases (also called preferred sub-theories in (Brewka 1989)) are maximal consistent subbases.

lexicographic preference (see (Lehmann 1992; Benferhat *et al.* 1993)) This is the strict ordering defined by $X \ll^{\text{lex}} Y$ iff there exists i such that $|X_i| < |Y_i|$ and for any

	General propositional case	Base with a strict ordering	Horn base	Horn base with a strict ordering
UNI-T	Π_2^p-complete	not applicable	co-NP-complete	not applicable
EXI-T	Σ_2^p-complete	not applicable	NP-complete	not applicable
ARG-T	Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) if $\Sigma_2^p \neq \Pi_2^p$	not applicable	Δ_2^p - (NP $\cup$ co-NP) if NP $\neq$ co-NP	not applicable
UNI-INCL	Π_2^p-complete	Δ_2^p-complete	co-NP-complete	P
EXI-INCL	Σ_2^p-complete	Δ_2^p-complete	NP-complete	P
ARG-INCL	Δ_3^p - ($\Sigma_2^p \cup \Pi_2^p$) if $\Sigma_2^p \neq \Pi_2^p$	Δ_2^p-complete	Δ_2^p - (NP $\cup$ co-NP) if NP $\neq$ co-NP	P
UNI-LEX	Δ_2^p-complete	Δ_2^p-complete	Δ_2^p-complete	P
EXI-LEX	Σ_2^p-complete	Δ_2^p-complete	Σ_2^p	P
ARG-LEX	Δ_3^p and Σ_2^p -hard	Δ_2^p-complete	Δ_3^p	P

Table 1: Computational complexity

$1 \leq j < i$, $|X_j| = |Y_j|$ (where $|Y|$ denotes the cardinality of Y). Note that the associated lexicographic preordering is complete. For any subbases X and Y of E, either $X \equiv^{\text{lex}} Y$ (*i.e.*, $|X_i| = |Y_i|$ for $i = 1 \ldots n$) or one of them is lex-preferred to the other one ($X \ll^{\text{lex}} Y$ or $Y \ll^{\text{lex}} X$). Lex-preferred consistent subbases are incl-preferred subbases, but the converse is false.

As mentioned in the introduction, coherence-based non-monotonic entailment from a stratified belief base takes two steps. In the following, PREF will denote a preference relation on the set of consistent subbases of E (as well as the mechanism which produces the associated preferred subbases). As a particular case, T (resp. INCL, LEX) will denote the mechanism which produces the set of maximal (resp. incl-preferred, lex-preferred) consistent subbases of E. Coherence-based non-monotonic entailment relations are defined by: $E \mid\!\sim^{p,\text{PREF}} \Phi$ iff Φ is inferred from the set of preferred (w.r.t. PREF) subbases of E according to the principle p.

The best known entailment principles are (1) the skeptical or UNI principle: Φ is classically entailed by *each* preferred subbase ; (2) the credulous or EXI principle: Φ is classically entailed by *at least one* preferred subbase.

Since the EXI principle leads to unsafe consequence relations (i.e. pairwise contradictory conclusions may be produced), the intermediary argumentative or ARG principle has been considered: Φ is classically entailed by *at least one* preferred subbase *and no* preferred subbase classically entails $\neg\Phi$.

We are now able to define the entailment relations and the associated problems which will be considered from the computational complexity point of view:

- The problem UNI-T (resp. EXI-T, ARG-T) is defined by "verify that Φ is inferred from the set of maximal consistent subbases of E according to the UNI (resp. EXI, ARG) principle" (the pre-ordering on E is not used).
- The problem UNI-INCL (resp. EXI-INCL, ARG-INCL) is defined by "verify that Φ is inferred from the set of inclusion-based preferred subbases of E according to the UNI (resp. EXI, ARG) principle".
- The problem UNI-LEX (resp. EXI-LEX, ARG-LEX) is defined by "verify that Φ is inferred from the set of lex-preferred subbases of E according to the UNI (resp. EXI, ARG) principle".

Computational complexity

We assume the reader has some knowledge on computational complexity theory and more precisely about the classes P, NP, co-NP as well as the polynomial hierarchy or PH (see (Garey & Johnson 1979)). We recall the essential conjecture of the complexity theory: NP $\neq$ P. Finding one single polynomial time algorithm for any NP-complete problem would make this conjecture false. This conjecture is generalized to the PH with the following stronger conjectures: NP $\neq$ co-NP and $\forall k, \Sigma_k^p \neq \Pi_k^p$. Note that NP $=$ P implies that the PH collapses into P.

We consider the general case and three possible restrictions of the problems previously presented[2]. The first restriction assumes that the belief base is completely and strictly ordered. In that case, E is stratified with exactly one formula per stratum. In the second restriction, we suppose that E and Φ are restricted to Horn clauses. The third restriction is the combination of the first and the second ones. The complexity results are given in table 1. Proofs may be found in (Cayrol & Lagasquie-Schiex 1993)[3].

The results are very discouraging in the sense that the few polynomial classes are incredibly restrictive and of poor practical interest while most other problems are located between Δ_2^p and Δ_3^p. One appealing relation is the UNI-LEX relation, which is "only" Δ_2^p-complete in the general case, but its complexity is mostly unaffected by the restrictions we considered. Therefore, rather than focusing on unrealistic polynomial classes, we have chosen to directly tackle

[2] When E and Φ are CNF formulae (conjunctive normal form), the complexity results remain unchanged.

[3] Some related proofs appear in (Nebel 1991; Gottlob 1992).

problems in the PH using adapted algorithms.

Binary decision diagrams

A BDD (Lee 1959; Bryant 1992) represents a boolean function ϕ using a labeled directed acyclic graph. The graph has two sink vertices labeled 0 and 1 representing the constant boolean function 0 and 1 respectively. Each non-sink vertex is labeled with a boolean variable v and has two out-edges labeled $then$ and $else$. The $then$ child corresponds to the case $v = 1$ and the $else$ child to the case $v = 0$. Given an order on the variables, an ordered BDD is a BDD with the constraint that all paths from the source to a sink visit the variables in an ascending order. Finally, a reduced ordered BDD is an ordered BDD such that each vertex represents a distinct boolean function. This representation is canonical (for a given order): an unsatisfiable formula will reduce to the sink 0 while a tautology will reduce to the sink 1. There is no restriction on the formulae represented (CNF or not).

A reduced ordered BDD (or ROBDD) may be viewed as a compressed decision tree for the formula. The decision tree may be transformed into the ROBDD by iteratively applying two reduction rules until quiescence: redundant vertices (such that the two out-edges point to the same vertex) are simply bypassed and deleted, pairs of vertices that denote the same function (same label and same $then$ and $else$ children if any) are merged. This reduction process is illustrated in Figure 1 (from (Bryant 1992)). In the ROBDD, each source-sink path encodes several truth assignments and each truth assignment is encoded in precisely one such path. In practice, BDD are *not* built by reducing a huge decision tree but incrementally: given the ROBDD representations of two formulae ϕ and ψ, the ROBDD representation for the formula $(\phi\langle op\rangle\psi)$, $\langle op\rangle$ being any binary boolean operator, can be computed in time $O(S_\phi \times S_\psi)$, where S_f denotes the number of vertices in the BDD representing f. The size of a ROBDD is therefore crucial for its applicability. This size is not necessarily exponential in the number of variables n and largely depends on the order of the variables. Despite the fact that the size is in $O(2^n)$, "there has been ample empirical evidence that many functions encountered in real applications can be represented efficiently as ROBDDs" (Bryant 1992).

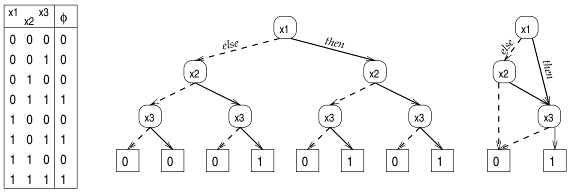

Figure 1: The truth table, decision tree and ROBDD of ϕ. A dotted edge corresponds to an assignment to 0, a solid edge to 1

Tackling the UNI-LEX problem with BDD

In the following, we show how ROBDD may be used to solve the UNI-LEX decision problem[4]. Consider a stratified belief base $E = \{\phi_i\}$ and V the set of variables that appear in the ϕ_i. As it has been done in the ATMS of de Kleer and in (Jégou 1994; Jégou & Bouquet 1995) for solving dynamic constraint satisfaction problems using BDD, we introduce one extra "assumption" variable ℓ_{ϕ_i} per formula in E. Let A be the set of assumption variables. We first build the ROBDD, noted R_E, which represents the conjunctions of all the formulae in $E_A = \{\ell_{\phi_i} \rightarrow \phi_i \mid \phi_i \in E\}$. Note that a satisfying truth assignment of E_A defines both a consistent subbase of E ($\ell_{\phi_i} = 1$ means that ϕ_i in included in the subbase), and a model of this subbase (if $\ell_{\phi_i} = 1$, then ϕ_i is necessarily satisfied since our truth assignment satisfies E_A). Thus, all the consistent subbases of E are encoded, along with their models, in all the paths from the source to sink 1 of R_E.

Given any order on V, (Jégou & Bouquet 1995) shows that inserting all the assumption variables after V yields a ROBDD R_E which uses less than $[2^n.(m+1) - 1]$ non terminal vertices, a much better result than the obvious $[2^{n+m} - 1]$. We propose instead to insert the variable $\ell_{\phi_i} \in A$ just after all the variables that appear in ϕ_i. This order is used in our simple example, at the end of the section. n_ℓ being the number of non-assumption variables before assumption ℓ in the resulting order, we have proven that the number of non terminal vertices of the BDD R_E using this order is bounded by $[(2^n - 1) + \sum_{\ell \in A} 2^{n_\ell}]$ which is always lower or equal to $2^n.(m+1) - 1$. Moreover, this order yields much better results than (Jégou & Bouquet 1995)'s order in practice (see experiments).

To identify preferred subbases among all the consistent subbases, we follow the idea of (Jégou & Bouquet 1995) by weighting the $else$ edge of each vertex labeled by an assumption. The weight represents the "cost" of rejecting the formula ϕ_i from the base E to get a consistent subbase. If the length of a path is defined as the sum of the weights of its edges[5], preferred (least cost) subbases are simply encoded in shortest paths from the source to sink 1 in the DAG of the ROBDD R_E.

To encode the LEX preference, the weight will be related to the stratum in which ϕ_i appears in E. This weight $w(n)$ is 1 for the formulae in the least prioritary stratum n and is inductively defined as $w(i) = 1 + \sum_{i<j\leq n}[w(j) \times |E_j|] = w(i+1) \times (|E_{i+1}| + 1)$. In this way, the weight associated to a formula in stratum i is larger than the sum of all the weights of all the formulae in less prioritary strata and this ensures that least cost subbases are LEX-preferred subbases (see a related transformation in (Dupin de Saint Cyr,

[4] EXI-LEX and ARG-LEX can also be tackled in a similar way, with some additional tricks.

[5] Edges without weight count for 0 in that sum.

Test	Sizes		CPU (R_E)		Size	CPU (R''_E)	
	(Jégou & Bouquet 1995)	our order	(Jégou & Bouquet 1995)	our order	R''_E	(Jégou & Bouquet 1995)	our order
Ping.	12	10	≈ 0	≈ 0	4	≈ 0	≈ 0
31 f.	252 932	62 028	1'07"	4.7"	17	12.8"	3.1"
77 f.	748 461	83 040	4'25"	17.3"	17	38"	4.2"

Lang, & Schiex 1994)).

As indicated in (Jégou & Bouquet 1995), a linear time algorithm, derived from Bellman's algorithm can then be applied to compute, for each vertex, the length of the shortest path from this vertex to sink 1. We have enhanced the algorithm in order to simultaneously build a new ROBDD R'_E that contains only the shortest paths from the source to 1 (all non shortest paths are simply redirected to sink 0). The modified algorithm remains linear. The ROBDD R'_E can then be used to solve all LEX based problems. In the specific case of UNI-LEX, we can quantify existentially R'_E on all the assumptions[6] and obtain a new ROBDD R''_E whose paths from source to sink 1 implicitly represent all the models of all the preferred subbases of E. Then a formula is UNI-LEX entailed by E iff it is *classically* entailed by R''_E: the belief base is effectively compiled in the BDD R''_E. This process is illustrated on a simple version of the penguin problem defined by the belief base E with 2 strata and 4 formulae: $\phi_1 = p, \phi_2 = p \rightarrow b, \phi_3 = p \rightarrow \neg f, \phi_4 = b \rightarrow f$. ϕ_1 and ϕ_2 are in the most prioritary stratum. The initial order on V is defined by $p < b < f$. The 4 variables c_1 to c_4 are used as assumptions for the 4 previous formulae. Using our ordering process, the final order used in the ROBDD is $p < c_1 < b < c_2 < f < c_3 < c4$. Applying our weighting process, we get $w(2) = 1$ and $w(1) = 3$.

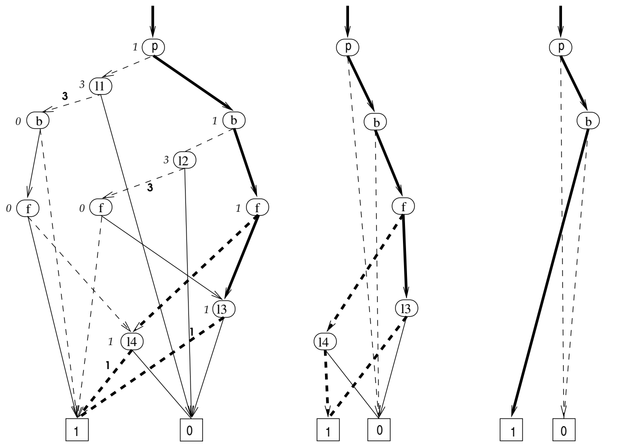

Figure 2: The ROBDD R_E, R'_E and R''_E on the penguin problem

The Figure 2 successively presents the ROBDD R_E (bold numbers indicating weight on dashed $else$ edges and italic number indicating the length of the shortest path from the vertex to sink 1), R'_E and R''_E. R'_E shows that the belief base has two LEX preferred subbases that respectively reject ϕ_3 and ϕ_4. R''_E implicitly represents the two models of these two subbases.

[6] Because of the variable order that places an assumption after all the variables of the associated formula,the $then$ children of all assumption vertices in R'_E are equal to terminal vertex 0. This makes the quantification operation very easy: it can even be performed during the application of the modified Bellman's algorithm, without destroying its linear time complexity.

Experimentations

We have extended the BDD package described in (Brace, Rudell, & Bryant 1990) with the previously sketched algorithms and applied it to three stratified belief-bases: the (in)famous penguin problem and two formalizations of a small real common-sense reasoning problem that respectively involve 31 and 77 formulae, distributed in respectively 7 and 9 strata. The table above gives successively: the size (in number of non-terminal vertices) of the BDD R_E using (Jégou & Bouquet 1995)'s order and our order (starting from the same order on V), the CPU time needed to compute R_E for each order, the size of the BDD R''_E (same for the two orders) and the CPU time needed to compute R''_E in each case. The CPU times have been measured on a SparcServer 1000 (slightly slower than a Sparc 10).

Note that the order we propose yields much smaller BDD. Better results could possibly be obtained by optimizing the initial order on V in order to minimize the term $\sum_{\ell \in A}(2^{n_\ell})$. For the problems considered, the size of the BDD R_E is reasonable and yields a final BDD R''_E which is very small. Obviously, larger problems could be tackled but the field of non-monotonic reasoning lacks benchmarks. Large UNI-LEX instances could be first "compiled" in the BDD R''_E which could then be used for entailment.

References

Benferhat, S.; Cayrol, C.; Dubois, D.; Lang, J.; and Prade, H. 1993. Inconsistency management and prioritized syntax-based entailment. In Bajcsy, R., ed., *Proc. of the 13th IJCAI*, 640–645. Chambéry, France: Morgan-Kaufmann.

Brace, K. S.; Rudell, R. R.; and Bryant, R. E. 1990. Efficient implementation of a BDD package. In *27th ACM/IEEE Design Automation Conference*, 40–45.

Brewka, G. 1989. Preferred subtheories : An extended logical framework for default reasoning. In Sridharan, N., ed., *Proc. of the 11th IJCAI*, 1043–1048. Detroit, MI: Morgan-Kaufmann.

Bryant, R. E. 1992. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comp. Surveys* 24(3):293–318.

Cayrol, C., and Lagasquie-Schiex, M.-C. 1993. Comparaison de relations d'inférence non-monotone : étude de complexité. Technical Report 93-23R, Institut de Recherche en Informatique de Toulouse (I.R.I.T.), France.

Cayrol, C., and Lagasquie-Schiex, M.-C. 1995. Non-monotonic syntax-based entailment: A classification of consequence relations. In Froidevaux, C., and Kohlas, J., eds., *Proc. of ECSQARU'95 (Lectures Notes in Artificial Intelligence 946)*, 107–114. Fribourg, Switzerland: Springer Verlag.

Dupin de Saint Cyr, F.; Lang, J.; and Schiex, T. 1994. Gestion de l'inconsistance dans les bases de connaissances : une approche syntaxique basée sur la logique des pénalités. In *Actes de RFIA'94*, 507–518.

Gärdenfors, P., and Makinson, D. 1994. Nonmonotonic inference based on expectations. *Artificial Intelligence* 65:197–245.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability : A Guide to the Theory of NP-completeness*. New York: W.H. Freeman and Company.

Gottlob, G. 1992. Complexity results for nonmonotonic logics. *Journal of Logic and Computation* 2(3):397–425.

Jégou, P., and Bouquet, F. 1995. Solving over-constrained CSP using weighted OBDD. In *Proceedings of the CP'95 Workshop on Over-Constrained Systems*.

Jégou, P. 1994. A logical approach to solve dynamic CSPs: Preliminary report. In *ECAI'94 Workshop on Constraint Satisfaction Issues raised by Practical Applications*, 87–94.

Lee, C. Y. 1959. Representation of switching circuits by binary-decision programs. *Bell System Tech. J.* 38(4):985–999.

Lehmann, D. 1992. Another perspective on default reasoning. Technical Report 92-12, Leibniz Center for Research in Computer Science. Hebrew University of Jerusalem, Israel.

Nebel, B. 1991. Belief revision and default reasoning: Syntax-based approaches. In *Proc. of the 2^{nd} KR*, 417–428. Cambridge, MA: Morgan-Kaufmann.