Consiglio Nazionale delle Ricerche

Istituto di Elaborazione dell'Informazione

A Note on the Complexity of "Simple" Heterogeneous Model-Preference Default Theories

Fabrizio Sebastiani & Umberto Straccia

Nota Interna IEI-B4-37-1990

Via S. Maria, 46 - 56126 Pisa (Italy)
Phone : +39.(0)50.553159
Fax : +39.(0)50.554342
Telex : 590305 IEICNR I

A Note on the Complexity of "Simple" Heterogeneous Model-Preference Default Theories

Fabrizio Sebastiani[1]
Istituto di Elaborazione dell'Informazione
Consiglio Nazionale delle Ricerche
Via S. Maria, 46 - 56126 Pisa (Italy)

Umberto Straccia
Dipartimento di Informatica
Corso Italia, 40 - 56126 Pisa (Italy)

Technical Report IEI-B4-37-1990

April 1990

Abstract

In a recent paper one of these authors argued that the notion of maximality (which he dubbed "≤-maximality") which Selman and Kautz's model-preference default systems rely upon leads to unintuitive results in the heterogeneous case; as a result, he proposed a new notion of "$\leq^i$-maximality" that fixes the problem. In this paper we show that in all model-preference default systems reasoning with $\leq^i$-maximality in the simple heterogeneous case is no harder than reasoning with ≤-maximality in the homogeneous case. This allows to extend to the simple heterogeneous case results found by Selman and Kautz for the homogeneous case. We also argue that, in practice, reasoning in the simple heterogeneous case is faster than in the homogeneous case.

1 Current address: Department of Computer Science, University of Toronto, M5S 1A4 Toronto, Ontario, Canada. E-mail: fabrizio@ai.toronto.edu

1 Introduction

In their 1988 paper "The complexity of model-preference default systems" (hereafter [MPD]) Selman and Kautz introduce model-preference default systems, a class of formalisms allowing the generation of a vivid knowledge base from a heterogeneous default theory $<D, T>$, where D is a set of defaults (or *default theory*) and T is a theory of propositional logic. In a previous paper (Sebastiani 1990 - hereafter [HMP]) one of these authors argued that in the "heterogeneous case" (i.e. the case in which T is non-empty) these formalisms may license unintuitive behaviour, failing thus to capture the correct interaction between certain and defeasible information. In the same paper he provides a modification to these formalisms based on the notion of "$\leq^{i}$-maximality", showing how it formalizes our intuitions of how this interaction should be accommodated.

In this brief note we describe a result on the complexity of reasoning in model-preference default systems that incorporates the modifications proposed in [HMP]. This note is meant as a brief postscript to [HMP] itself, and therefore assumes familiarity with the material presented therein. The main result we discuss in this note is that, for any model-preference default system considered in [HMP] and variations thereof, the search problem in the "simple heterogeneous case" (i.e. the heterogeneous case in which T is a consistent set of literals[2]) is no harder than the "homogeneous case" (i.e. the case in which T is empty) for the corresponding system of [MPD]. We prove this result by giving an algorithm that, given an HD-theory $<T, D>$ (where T is a consistent set of literals) defined on some alphabet P, builds a default theory D^{i} defined on $P^{i} \equiv P\text{-}P_T$ (where P_T is the set of propositional letters in P that occur in T) such that, whenever a model M^{i} is $\leq$-maximal wrt D^{i}, the model $M \equiv M^{i} \cup T$ is $\leq^{i}$-maximal wrt $<T, D>$. Essentially, what the algorithm does is first creating D^{i} as a copy of D, eliminating from it any default that does not contribute to the determination of $\leq^{i}$-maximality (either because its consequent is already satisfied by T, or because its antecedent or its consequent are inconsistent with T) and eliminating from the antecedents of the remaining defaults all literals that already appear in T. The models that satisfy T and the preference relations between them that are induced by D form a graph that is isomorphic to the one formed by all models of the propositional language $P\text{-}P_T$ and by the preference relations between them induced by D^{i}. Due to the isomorphism, there is a one-to-one correspondence between $\leq^{i}$-maximal models endorsed by the former graph and $\leq$-maximal models endorsed by the latter.

We also show that, although the homogeneous and simple heterogeneous case have in [HMP] the same theoretical complexity, the homogeneous case is in practice *much harder* (somehow in contrast with what happens in [MPD]). This accounts for the rather intuitive fact that, as the amount of certain, complete information increases, the amount of computational resources required to "flesh out" the knowledge base should proportionally decrease.

---

2 Let $T = \{q_1, q_2, ..., q_m\}$ be a set of literals defined on some alphabet $P = \{p_1, p_2, ..., p_n\}$, $m \leq n$; T is *consistent* if for no $i = 1, ..., n$ it happens that both p_i and $\neg p_i$ belong to T.

2 The complexity of reasoning with simple heterogeneous theories

In trying to understand how far the complexity results of [MPD] carry over to systems where $\leq^i$-maximality is used, and to understand whether new results can be established for such systems, we will be exclusively interested in results concerning the heterogeneous case, as in the homogeneous case $\leq$-maximality and $\leq^i$-maximality trivially coincide. The two cases have been shown in [MPD] to have, in general, different complexity. For example, the problem of reasoning in $\mathcal{DH}$ is linear in the homogeneous case, polynomial in the simple heterogeneous case and NP-hard in the "Horn heterogeneous case" (i.e. the hetherogeneous case in which T is a set of Horn clauses). Also, for some of these systems results are known for the homogeneous case but not for the various heterogeneous cases; in particular, for the homogeneous case NP-hardness results for $\mathcal{D}$ and $\mathcal{DH}^+$ and a polynomial result for $\mathcal{DH}_a{}^+$ are reported in [MPD], but the corresponding results for the various heterogeneous cases are still unknown.

As remarked in [MPD], the problem of finding a model which is maximal wrt a pair $<D, T>$ is best viewed as a search problem. A *search problem* Π is defined by Garey and Johnson [1979] as consisting of a set D_Π of *instances* and, for each I in D_Π, a set $S_\Pi[I]$ of *solutions* for I; an algorithm is said to solve Π if, when given as input an arbitrary I in D_Π, returns "no" whenever $S_\Pi[I]$ is empty and otherwise returns some s in $S_\Pi[I]$. The main result of this paper is that, for any model-preference default system considered in [MPD] and variations thereof, if $\leq^i$-maximality is used the search problem in the simple heterogeneous case is no harder than the homogeneous case. The result is based on the key observation that in both cases only internal paths are allowed (internal to the set of all models of the language P in the former case, and to the set of all models satisfying T in the latter), whereas, according to $\leq$-maximality, external paths are allowed in the heterogeneous cases while they are not in the homogeneous case.

Theorem The search problem for $\mathcal{S}^i$ in the simple heterogeneous case has the same complexity as the search problem for $\mathcal{S}$ in the homogeneous case, where $\mathcal{S}$ is any model-preference default system discussed in [MPD] and $\mathcal{S}^i$ is the version of $\mathcal{S}$ relying on $\leq^i$-maximality. ■

Proof We prove this theorem by giving an algorithm (described in Appendix A, and whose Common Lisp implementation is detailed in Appendix B) that:

1. given an HD-theory $<T, D>$ (where T is a consistent set of literals) defined on some alphabet P, builds a default theory D^i defined on $P^i \equiv P\text{-}P_T$ (where P_T is the set of propositional letters in P that occur in T) such that, whenever a model M^i is $\leq$-maximal wrt D^i, the

model $M \equiv M^{i} \oplus T$ is $\leq^{i}$-maximal wrt <T, D>[3];

2. is linear in the number of occurrences of literals in D.

In other words, we reduce reasoning about $\leq^{i}$-maximality to reasoning about $\leq$-maximality, and the reduction is accomplished in linear time. Since reasoning about $\leq$-maximality is at least linear in all model-preference default systems, this proves the theorem.

To show that property 1 holds we show that there is an isomorphism between the relation "$\leq +$" induced by D on the models of P that also satisfy T and the relation "$\leq +$" induced by D^{i} on the models of P^{i}, where D^{i} is the default theory returned by the algorithm. As the two sets of models are the ones on which $\leq^{i}$-maximality wrt <D, T> and $\leq$-maximality wrt T, respectively, are computed, and since the two maximality notions only depend on the topological properties of the sets of models on which they are computed, this will prove that the algorithm has the described property.

(=>) We first show that whenever a default $d \equiv \alpha \rightarrow q$ induces the relation $M_j \leq + M_l$, where M_j and M_l are models of P that also satisfy T, there exists a default $d' \equiv \beta \rightarrow q$ in D^{i} that induces the relation $M_j^{i} \leq + M_l^{i}$, with $M_j^{i} \equiv M_j - T$ and $M_l^{i} \equiv M_l - T$ being models of P^{i}. Our algorithm initially inserts a copy of d, renamed d', in D^{i}. According to the hypothesis M_j does not satisfy q[4], hence q is not in T, hence action (1) in the algorithm does not delete d' from D^{i}. M_l satisfies q, hence $\{q\} \cup T$ is consistent, hence action (2) in the algorithm does not delete d' from D^{i}. Both M_j and M_l satisfy α, hence $\alpha \cup T$ is consistent, hence no negation of any literal in α may be in T, hence action (3) in the algorithm does not delete d' from D^{i}. Given this, there exists a default $d' \equiv \beta \rightarrow q$ in D^{i} such that $\alpha \equiv \beta \oplus \gamma$ where β does not contain literals built out of P_T, and where the literals in γ have been deleted from d' by action (4). As $\alpha \equiv \beta \oplus \gamma$ is satisfied by both M_j and M_l, so is β; as β does not contain literals built out of P_T, β is also satisfied by $M_j^{i} \equiv M_j - T$ and $M_l^{i} \equiv M_l - T$. q is not satisfied by M_j and is not built out of letters in P_T, hence it is not satisfied by $M_j^{i} \equiv M_j - T$ either. Similarly, q is satisfied by M_l and is not built out of letters in P_T, hence it is also satisfied by $M_l^{i} \equiv M_l - T$. Hence d' induces a "$\leq +$" relation between M_j^{i} and M_l^{i}.

(<=) We now show that whenever $d' \equiv \beta \rightarrow q$ induces a relation $M_j^{i} \leq + M_l^{i}$, where M_j^{i} and M_l^{i} are models of P^{i}, there exists a default $d \equiv \alpha \rightarrow q$ in D that induces the relation $M_j \leq + M_l$ with $M_j \equiv M_j^{i} \oplus T$ and $M_l \equiv M_l^{i} \oplus T$ models of P. Let $d \equiv \alpha \rightarrow q$ be a default in D that originated d' through the application of the algorithm; then $\alpha \equiv \beta \oplus \gamma$, with γ a subset of T. As β is satisfied by M_j^{i} and M_l^{i}, and as γ is a subset of T, then $\alpha \equiv \beta \oplus \gamma$ is satisfied by $M_j \equiv M_j^{i} \oplus T$ and $M_l \equiv M_l^{i} \oplus T$. As q is not satisfied by M_j^{i} and is not built out of letters in P_T, it is not satisfied by $M_j \equiv M_j^{i} \oplus T$ either. Instead, q is satisfied by M_l^{i} and is not built out of letters in P_T, so q is satisfied by $M_l \equiv M_l^{i} \oplus T$. Hence d induces a "$\leq +$" relation between M_j and M_l.

3 For convenience, in this proof we sometimes equate a model with the largest set of literals that is satisfied by it (e.g. we equate the model a¬bc with the set of literals $\{a, \neg b, c\}$); this allows us to use set-theoretic operations to create models of P from models of P^{i} and viceversa. Also, "$\oplus$" refers here to the union of disjoint sets.

4 We disregard the irrelevant case in which $M_j = M_l$.

The algorithm is linear in the number of occurrences of literals in D; in fact every such occurrence needs to be examined at most once, and each time a literal is examined its appartenance to the theory is checked in constant time. ■

The following table summarizes complexity results in the heterogeneous case (with T a consistent set of literals) for the main model-preference default systems.

	$\mathcal{D}$	$\mathcal{DH}$	$\mathcal{DH}^+$	$\mathcal{DH}_a^+$
$\leq$-maximality	[unknown]	polynomial	[unknown]	[unknown]
$\leq^i$-maximality	NP-hard	linear	NP-hard	polynomial

Also, note that although the homogeneous and heterogeneous case have the same theoretical complexity, the heterogeneous case is in practice *much simpler* (somehow in contrast with what happens for $\leq$-maximality). For example, the search problem for the homogeneous case of $\mathcal{DH}$ is $O(n)$, where n is the number of occurrences of literals in D. Although the heterogeneous case (with $\leq^i$-maximality) of $\mathcal{DH}$ is still $O(n)$, n is now the the number of occurrences of literals in D^i, a subset of D. That by adopting $\leq^i$-maximality the heterogeneous case should be simpler is also apparent from the fact that the presence of a theory T consisting of a set of k literals transforms the problem of searching the set of all models of the propositional language P into the one of searching only the set of models that satisfy T: the latter graph has 2^k times less nodes than the former. This accounts for the rather intuitive fact that, as the amount of certain information increases, the amount of computational resources required to “flesh out" the knowledge base should proportionally decrease.

Bibliography

[Garey & Johnson, 1979] Garey, Michael R. & Johnson, David S. *Computers and intractability: a guide to the theory of NP-completeness.* S. Francisco, CA: Freeman.

[Sebastiani, 1990] Sebastiani, Fabrizio. On heterogeneous model-preference default theories. In *Proceedings of CSCSI/SCEIO'90, the 8th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Ottawa, Canada, pp. 84-91.

[Selman & Kautz, 1988] Selman, Bart & Kautz, Henry A. The complexity of model-preference default theories. In *Proceedings of CSCSI/SCEIO'88, the 7th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, Edmonton, Alberta, pp. 102-109, 1988. [1] Appears also in Reinfrank, Michael; De Kleer, Johan; Ginsberg, Matthew L. & Sandewall, Erik (eds.) (1989), Nonmonotonic reasoning, Heidelberg, BRD: Springer, pp. 115-130. [2] An extended version is forthcoming as “Model-preference default theories" in Artificial Intelligence.

Appendix A Transformation algorithm

Input : an HD-theory $H = <T, D>$ defined on an alphabet P, with T a consistent set of literals and D a set of defaults;

Output : a default theory D' defined on P-P_T (where P_T is the set of propositional letters in P that occur in T) such that, whenever a model M is ≤-maximal wrt D', the model $M \cup T$ is $\leq^i$-maximal wrt $<T, D>$.

```
begin
  D' ::= D                                  % renaming D
  for all d_i = α_i → q_i in D'
    if q_i is in T
       then  delete d_i from D'             % action (1) : the default would not convey new
             information
    endif
    if the negation of q_i is in T
       then  delete d_i from D'             % action (2) : the conclusion would contradict the theory
    endif
    for all q_ij in α_i
       if  the negation of q_ij is in T
          then  delete d_i from D'          % action (3) : the default would not be applicable
    endif
       if  q_ij is in T
          then  delete q_ij from α_i        % action (4) : the literal does not belong to the language
    endif
    end for
  end for
end
```

Appendix B Common Lisp code

```
%---------------------------------------------------------------------
%             conversion procedure
%---------------------------------------------------------------------
```

% returns the default theory D_B and the alphabet P-P_T upon which D_B is defined

```
(defun generate-pair  (defaults theory alphabet)
        (cons (conversion defaults theory alphabet) (diff alphabet theory)))

% converts the default theory
(defun conversion (defaults theory alphabet)
        (cond ((null defaults) nil)
                (t (cons (single-conversion (car defaults) theory alphabet)
                           (conversion (cdr defaults) theory alphabet)))))

% converts a single default
(defun single-conversion (default theory alphabet)
        (cons (test-antecedent (antecedent default) theory alphabet)
                (test-consequent (consequent default) theory alphabet)))

% makes sure default has not been ruled out by the conversion procedure performed on the
% antecedent
(defun test-antecedent (antecedent theory alphabet)
        (cond ((removed-p antecedent) nil)
                (t antecedent)))

% makes sure default has not been ruled out by the conversion procedure performed on the
% consequent
(defun test-consequent (consequent theory alphabet)
        (cond ((removed-p consequent) nil)
                (t consequent)))

% converts the antecedent of a default
(defun convert-antecedent (antecedent theory alphabet)
        (cond ((null antecedent) nil)
                ((member (car antecedent) theory)
                   (convert-antecedent ((cdr antecedent) theory alphabet)))
                   % disregard literal already endorsed by theory
                ((member (negation (car antecedent)) theory) (list 'remove-default))
                   % disregard default with antecedent contradicted by theory
                (t (cons (car antecedent) (convert-antecedent (cdr antecedent)))))))

% converts the consequent of a default
(defun convert-consequent (consequent theory alphabet)
```

```
    (cond ((member consequent theory) (list 'remove-default))
              % disregard default with consequent already endorsed by theory
          ((member (negation consequent) theory) (list 'remove-default))
              % disregard default with consequent contradicted by theory
          (t consequent)))

%-------------------------------------------------------------------
%            selectors
%-------------------------------------------------------------------

(defun antecedent (default)
    (car default))

(defun consequent (default)
    (cdr default))

%-------------------------------------------------------------------
%            utilities
%-------------------------------------------------------------------

% returns the result of removing all elements of list2 from list1
(defun diff (list1 list2)
    (cond ((null list1) nil))
          ((member (car list1) list2) (diff (cdr list1) list2))
          (t (cons (car list1) (diff (cdr list1) list2)))))

% returns the negation of the given literal
(defun negation (literal)
    (cond ((eq (car literal) 'not) (cdr literal))
          (t (cons 'not literal))))

% searches recursively for the presence of 'remove-default in a nested list
(defun removed-p (list)
    (cond ((null list) nil)
          ((list (car list))  (cond  ((removed-p (car list)))
                                     (t (removed-p (cdr list)))))
          ((eq (car list) 'remove-default))
          (t (removed-p (cdr list)))))
```