Two forms of dependence in propositional logic: controllability and definability

Jérôme Lang
IRIT-UPS
118 route de Narbonne
31062 Toulouse Cedex, France
e-mail: lang@irit.fr

Pierre Marquis
CRIL/Université d'Artois
rue de l'Université – S.P. 16
62307 Lens Cedex, France
e-mail: marquis@cril.univ-artois.fr

Abstract

We investigate two forms of dependence between variables and/or formulas within a propositional knowledge base: *controllability* (a set of variables X controls a formula Γ if there is a way to fix the truth value of the variables in X in order to achieve Γ to have a prescribed truth value) and *definability* (X defines a variable y if every truth assignment of the variables in X enables us finding out the truth value of y). Several characterization results are pointed out, complexity issues are analyzed, and some applications of both notions, including decision under incomplete knowledge and/or partial observability, and hypothesis discrimination, are sketched.

Introduction

For many reasoning tasks which make use of propositional logic, exhibiting structure can be of a great help. By "structure" we mean some relationships that exist between some sets of variables and/or formulas within a propositional knowledge base Σ. A nice example of such structure, which has received much attention recently, is *independence* (Darwiche 1997) (Lakemeyer 1997) and related structural properties such as relevance (Lakemeyer 1995), or causal independence (Darwiche & Pearl 1994). Revealing independence relations in Σ not only helps understanding Σ better but also is a great help for making easier some reasoning tasks such as satisfiability, deduction, abduction or diagnosis (Darwiche 1997). Apart from independence other kinds of structural properties are worth investigating, especially different kinds of *dependence* involving sets of variables and/or formulas. In this paper, two particular forms of dependence are studied:

- *controllability*: a set of variables X controls a formula Γ w.r.t. Σ if it is always possible to fix the values of some of the variables in X in order to achieve Γ to have a given truth value (true or false); the particular case where one is only interested in Γ being true corresponds to achieving a *goal* and relates to qualitative decision making under incomplete knowledge.
- *definability*: a set of variables X defines a variable y w.r.t. Σ if whatever the observed values of all variables of X are, they enable us finding out the truth value of y. This notion has many applications, including designing test policies in order to discriminate among hypotheses (such as plausible diagnoses).

For these two kinds of dependence, several definitions are introduced together with their specific interest, some characterizations are given, computational complexity issues are investigated, and some applications ranging from decision under partial observability to fault isolation in model-based diagnosis, are also sketched. As to definability, this paper completes a companion paper (Lang & Marquis 1998) in several directions, including the practical computation of definability relations.

Formal Preliminaries

Let PS be a countable set of propositional variables and $PROP_{PS}$ the propositional language built up from PS, the connectives and the boolean constants $true$ and $false$. For $X \subseteq PS$, $PROP_X$ denotes the sublanguage of $PROP_{PS}$ generated from the variables of X only. Elements (resp. subsets) of PS are denoted x, y, *etc.* (resp. X, Y, *etc).* Full instantiations of variables of $X \subseteq PS$ (called X–worlds) are denoted by ω_X and their set is denoted Ω_X. Σ denotes a finite propositional knowledge base, i.e., a conjunctively-interpreted finite set of propositional formulas from $PROP_{PS}$. $Var(\Sigma)$ is the set of propositional variables appearing in formula Σ.

For every formula Φ and every $x \in PS$, $\Phi_{x \leftarrow 0}$ (resp. $\Phi_{x \leftarrow 1}$) is the formula obtained by replacing in Φ every occurence of x by $false$ (resp. $true$).

Given a propositional knowledge base Σ, the set of prime implicants modulo Σ of a formula Φ over $PROP_X$ will be denoted by $PI^X_\Sigma(\Phi)$; this set is defined by $PI^X_\Sigma(\Phi) = max(\{PI(\Sigma \Rightarrow \Phi) \cap PROP_X\}, \models)$ where $PI(\Phi)$ denotes the set of prime implicants of Φ for every formula Φ. $PI^X(\Phi)$ (= $PI^X_{true}(\Phi)$) denotes the subset of $PI(\Phi)$ consisting of the terms built upon X, only. $IP(\Phi)$ denotes the set of prime implicates of Φ. For instance, if $\Sigma = \{a \vee b, \neg a \wedge c \Rightarrow e, d \Leftrightarrow e\}$ then $PI^{\{a,b,c,d\}}_\Sigma(e) = \{d, \neg a \wedge c, \neg a \wedge \neg b\}$, $PI^{\{a,c\}}_\Sigma(e) = \{\neg a \wedge c\}$, $PI^{\{b,c\}}_\Sigma(e) = \emptyset$, $PI^{\{a,b,c,d\}}_\Sigma(a \vee b) = \{true\}$, $IP(\Sigma) = \{a \vee b, a \vee \neg c \vee e, \neg d \vee e, d \vee \neg e, a \vee \neg c \vee d\}$.

In this paper we refer to some complexity classes above NP and coNP, details about which can be found in Papadimitriou's textbook (Papadimitriou 1994).

Conditional controllability

Let Σ be a propositional knowledge base, $X, Z \subseteq PS$, and Γ be a formula. Intuitively, X positively controls Γ given Z w.r.t. Σ means that for any observed Z-world ω_Z there is a X-world ω_X which certainly achieves Γ. A very intuitive interpretation of positive controllability relates to decision under incomplete knowledge and partial observability: Z is the set of observable variables, Ω_Z the observation space, X the set of controllable variables, Ω_X the action space (an action being the composition of elementary actions, an elementary action assigning a variable of X to either $true$ or $false$), and Γ the goal.

Controllability in a logical setting has not received much attention so far. The first approach we know is in (Boutilier 1994) where an action model for qualitative decision theory is based on a partition between controllable and uncontrollable variables. While inspired by the latter, this study extends it in several directions, especially regarding to observability and complexity (some of our results applying to Boutilier's framework). Controllability appears also in the independent choice logic (Poole 1997), where each variable is assigned a specific agent controlling it, and in (Fargier, Lang, & Schiex 1996) in a constraint satisfaction framework.

We may think of defining conditional positive controllability as follows:

X positively controls Γ given Z w.r.t. Σ iff $\forall \omega_Z \in \Omega_Z$ $\exists \omega_X \in \Omega_X$ such that $\omega_Z \wedge \omega_X \wedge \Sigma \models \Gamma$.

Now, there are two points which must be considered before going further:

1. *What if $\omega_Z \wedge \Sigma$ is inconsistent?* This means that observation ω_Z is impossible: it can merely not happen. Hence assigning an action to ω_Z is needless.
2. *What if $\omega_Z \wedge \Sigma$ is consistent and $\omega_Z \wedge \omega_X \wedge \Sigma$ is inconsistent?* This is more difficult to interpret, or at least more ambiguous. The most intuitive interpretation is that when ω_Z is observed, the action ω_X is simply not available[2].

We now have the elements for defining formally conditional controllability and related notions:

Definition 1 (conditional controllability)
Let $\Sigma, \Gamma \in PROP_{PS}$ and $X, Z \subseteq PS$.
- X **positively controls** Γ *given Z w.r.t. Σ (denoted by $X \ll_\Sigma^Z \Gamma^+$) iff $\forall \omega_Z \in \Omega_Z$ s.t. $\omega_Z \wedge \Sigma$ is consistent $\exists \omega_X \in \Omega_X$ s.t. (i) $\omega_Z \wedge \omega_X \wedge \Sigma$ is consistent and (ii) $\omega_Z \wedge \omega_X \wedge \Sigma \models \Gamma$.*
- X **fully controls** Γ *given Z w.r.t. Σ (denoted by $X \ll_\Sigma^Z \Gamma$) iff both $X \ll_\Sigma^Z \Gamma^+$ and $X \ll_\Sigma^Z (\neg\Gamma)^+$.*

[2]As noticed by Fargier (personal communication), in the case where it is guaranteed that any action can always be performed then the specification of the decision problem *must* be s.t. for any ω_Z, if $\omega_Z \wedge \Sigma$ is consistent, then for every ω_X, $\omega_Z \wedge \omega_X \wedge \Sigma$ is consistent.

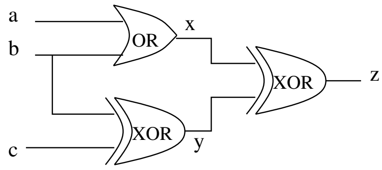

Figure 1: A circuit.

Positive controllability intuitively means that there is a way to fix the values of variables in X in order to make the goal Γ true; full controllability means that both Γ and $\neg\Gamma$ can be achieved. Note that the technical difference between this definition of positive controllability and the definition given above without consistency conditions does not rely on whether impossible observations are taken into account or not; indeed, if $\omega_Z \wedge \Sigma$ is inconsistent, then $\exists \omega_X$ s.t. $\omega_Z \wedge \omega_X \wedge \Sigma \models \Gamma$ is trivially satisfied (any ω_X does the job). So the only difference is requiring the action ω_X assigned to an observation to be available, i.e. consistent with the observation and the knowledge base.

As an illustration, let us consider the circuit depicted on Figure 1. Let $\Sigma = \{x \Leftrightarrow a \vee b, y \Leftrightarrow (b \Leftrightarrow \neg c), z \Leftrightarrow (x \Leftrightarrow \neg y)\}$. We have $\{a, c\} \ll_\Sigma^{\{b\}} x^+$ but *not* $\{a, c\} \ll_\Sigma^{\{b\}} x$; we also have $\{a, c\} \ll_\Sigma^{\{b\}} y$ and $\{a, c\} \ll_\Sigma^{\{b\}} z$. Note that we do *not* have $\{a, c\} \ll_\Sigma^{\emptyset} y^+$ (nor $(\neg y)^+$, nor z^+, nor $(\neg z)^+$) (we only have $\{a, c\} \ll_\Sigma^{\emptyset} x^+$).

Now, positive controllability can be characterized by means of prime implicants (similar results follow easily for other forms of controllability).

Proposition 1
$X \ll_\Sigma^Z \Gamma^+$ iff $\forall \omega_Z \in \Omega_Z$ s.t. $\omega_Z \wedge \Sigma$ is consistent, $\exists \delta \in PI_\Sigma^{X \cup Z}(\Gamma) \setminus PI^{X \cup Z}(\neg\Sigma)$ s.t. $\delta \supseteq \omega_Z$.

As already said, a practical application area of controllability is decision under incomplete knowledge and partial observability: $\mathcal{P} = \langle X, Z, \Sigma, \Gamma \rangle$ can be seen as a logical specification of a qualitative, single-step decision problem with incomplete knowledge and partial observability. The set of possible observations is $PossObs(\mathcal{P}) = \{\omega_Z \in \Omega_Z \mid \omega_Z \wedge \Sigma$ is consistent$\}$ and a *sound policy* for $\mathcal{P}$ is a mapping π from $PossObs(\mathcal{P})$ to Ω_X such that $\forall \omega_Z \in PossObs(\mathcal{P})$, (i) $\omega_Z \wedge \pi(\omega_Z) \wedge \Sigma$ is consistent, and (ii) $\omega_Z \wedge \pi(\omega_Z) \wedge \Sigma \models \Gamma$. Clearly, X positively controls Γ given Z w.r.t. Σ iff there exists a sound policy for $\mathcal{P}$.

We turn now to complexity issues. We first give a straightforward result which avoids studying separately positive and full controllability.

Proposition 2
$X \ll_\Sigma^Z \Gamma^+$ iff $X \cup \{new\} \ll_\Sigma^Z \Gamma \wedge new$, where $new \in PS \setminus (Var(\Sigma) \cup Var(\Gamma) \cup X \cup Z)$.

This reduction from full to positive controllability is clearly polynomial, and a polynomial reduction from positive to full controllability is a trivial consequence of the definitions. Thus, both notions are polynomially related. Consequently:

Corollary 1 (FULL) CONDITIONAL CONTROLLABILITY *and* POSITIVE CONDITIONAL CONTROLLABILITY *are in the same complexity classes.*

Since the reduction given by Proposition 2 preserves the restrictions that are considered in the following (i.e., $Z = \emptyset$ and $X \cup Z = Var(\Sigma)$), both problems remain in the same complexity classes for each of these restrictions. Accordingly, in the rest of the section, we will mainly focus on the complexity of full conditional controllability.

Proposition 3
CONDITIONAL CONTROLLABILITY *is* Π_3^p*-complete*

We are now going to investigate some particular restrictions of controllability, each of which corresponds to a particular type of decision problem. We start with *unconditional controllability*, obtained by letting $Z = \emptyset$.

Proposition 4
UNCONDITIONAL CONTROLLABILITY *is* Σ_2^p*-complete.*

Intuitively, unconditional controllability means that there is no observable variable – thus the action to be undertaken must be taken unconditionally, which corresponds to non-observability. Interestingly, UNCONDITIONAL POSITIVE CONTROLLABILITY can be abductively characterized; indeed, $X \ll_\Sigma^\emptyset \Gamma^+$ holds iff there exists an abductive explanation for Γ given Σ, where the set of possible individual hypotheses is the set of literals built up from X. Accordingly, the Σ_2^P-completeness of UNCONDITIONAL POSITIVE CONTROLLABILITY is recovered as a consequence of Theorem 4.2 from (Eiter & Gottlob 1995). Thanks to such an abductive characterization, the complexity of many restricted subcases of UNCONDITIONAL POSITIVE CONTROLLABILITY can be easily derived from (Eiter & Gottlob 1995).

Another particular case is obtained by letting $X \cup Z = Var(\Sigma)$. We call the corresponding problem *ceteris paribus* controllability.

Proposition 5
CETERIS PARIBUS CONTROLLABILITY *is* Π_2^P*-complete.*

Propositions 4 and 5 are similar to some complexity results in (Fargier, Lang, & Schiex 1996) for mixed constraint satisfaction. Intuitively, *ceteris paribus* conditional controllability means that all variables are either controllable or observable (full observability). *Ceteris paribus* controllability has been first proposed by Boutilier (Boutilier 1994) for $\Sigma = \emptyset$. His appealing characterization of controllability (X controls Γ iff $PI^X(\Gamma) \neq \emptyset$ and any $\delta \in PI(\Gamma)$ mentions a variable of X) is equivalent to ours when $\Sigma = \emptyset$, using the fact that Γ is equivalent to the disjunction of all its (standard) prime implicants. Despite the additional restriction $\Sigma = \emptyset$, the complexity of checking this form of *ceteris paribus* controllability does not fall down.

Proposition 6
Boutilier's controllability is Π_2^P*-complete.*

Definability

Definitions and characterizations

Definability is a stronger form of dependence than controllability: while the latter states that there is a way to fix a variable y to the desired truth value, definability imposes that for every X–world, the truth value of y is determined. The computational complexity of definability has been investigated in a companion paper (Lang & Marquis 1998); herefater, the focus is mainly led on the practical computing of definability. We start by a series of definitions concerning *definability* and later on we give a closely related definition, *hypothesis discriminability.*

Definition 2 (definability) *(Lang & Marquis 1998) Let* $\Sigma \in PROP_{PS}$*,* $X \subseteq PS$ *and* $y \in PS$*.*

- X **defines** y *w.r.t.* Σ *(denoted by* $X \sqsubseteq_\Sigma y$*) iff* $\forall \omega_X \in \Omega_X, \omega_X \wedge \Sigma \models y$ *or* $\omega_X \wedge \Sigma \models \neg y$*.*
- X **defines minimally** y *w.r.t.* Σ *iff* $X \sqsubseteq_\Sigma y$ *and no proper subset of* X *does it.*
- X **defines nontrivially** y *w.r.t.* Σ *iff* $X \sqsubseteq_\Sigma y$ *and* Σ *is consistent.*
- X *is a* **basis** *for* y *w.r.t.* Σ *iff* X *defines minimally and nontrivially* y *w.r.t.* Σ*.*

While every X–world that is not consistent with Σ can be considered impossible, requiring $\omega_X \wedge \Sigma$ to be consistent in the definition above would be useless since $\omega_X \wedge \Sigma \models y$ and $\omega_X \wedge \Sigma \models \neg y$ hold whenever $\omega_X \wedge \Sigma$ is inconsistent. When no X-world consistent with Σ can be found, Σ is inconsistent. In this case, definability trivializes, i.e., $X \sqsubseteq_\Sigma y$ holds for every X and every y, and no basis for y can be pointed out.

Clearly enough, the four definability relations given above can be easily extended to sets Y of variables by $X \sqsubseteq_\Sigma Y$ iff $X \sqsubseteq_\Sigma y$ for every $y \in Y$, as well as to formulas Γ (replacing y by Γ in the definitions above, see (Lang & Marquis 1998) for details).

As an illustration, let us step back to our example (Fig. 1). We have $\{a, b\} \sqsubseteq_\Sigma x$, $\{b, c\} \sqsubseteq_\Sigma y$, $\{a, b, c\} \sqsubseteq_\Sigma \{x, y, z\}$; note that $\{a, b, c\}$ defines minimally z and also $\{x, y\}$ w.r.t. Σ but *not* x nor y. Here is the list of all bases for y w.r.t. Σ: $\{y\}$, $\{b, c\}$, $\{x, z\}$ and $\{a, b, z\}$; for z we get the following bases: $\{z\}$, $\{a, b, c\}$, $\{x, y\}$, $\{a, b, y\}$, $\{x, b, c\}$ and $\{a, c, y\}$.

There is a clear link between (full) unconditional controllability and definability, which states that except in "pathological" cases, definability is stronger than unconditional controllability; indeed, $X \sqsubseteq_\Sigma y$ implies ($X \ll_\Sigma^\emptyset y$ or $\Sigma \models y$ or $\Sigma \models \neg y$). Furthermore, it is easily shown that if X defines *minimally* y w.r.t Σ, then X and y are marginally dependent in the sense of (Darwiche 1997) and that for any x in X then $\{x\}$ is relevant to $\{y\}$ in the sense of (Lakemeyer 1997)[3].

[3]Similar results would hold with a notion of *minimal* controllability which is omitted for considerations of space.

While definability has been intensively studied in mathematical logic (see e.g., (Beth 1953)), propositional definability (and its computational complexity) has received much less attention in AI, up to now. Let us nevertheless mention that similar notions have been introduced in the recent literature on causal reasoning (Darwiche & Pearl 1994) (Geffner 1996b), and especially (Geffner 1996a) who proposes a framework for ramification which makes use of a causality principle

> the values that a variable may take (...) is a function of the values of its causes

that is very similar to our notion of definability[4]. Another closely related work is by Ibaraki et al. (Ibaraki, Kogan, & Makino 1998), where the focus is laid on functional dependencies for Horn knowledge bases (functional dependency is definability).

The fact that definability has not yet been fully investigated by the AI community is somewhat surprising since it proves helpful for many AI applications. For instance, when reasoning about change, a way to address the well-known *frame problem* consists in finding out fluents that can be derived from primitive ones (called a frame, or a defining family in our framework) within the knowledge base, and to apply change on reduced world descriptions (composed of primitive fluents) (Lifschitz 1990). Many formalisms for reasoning about change, adhere to this approach that has been implemented in various planning systems (e.g., in the early system BUILD (Fahlman 1974)). The notion of basis can also prove valuable in automated reasoning. For instance, identifying functionally dependent variables is a way to find out variable orderings that may prevent the Binary Decision Diagram (BDD) representation of a formula from an exponential size blowup (Hu & Dill 1993). More recently, (Kautz, McAllester, & Selman 1997) have shown how variable dependency can be exploited in local search for the satisfiability problem.

We now turn back to logical characterizations of definability. As a corollary of Beth's theorem (Beth 1953) (stated in the more general framework of first-order logic), we get the equivalence between the *implicit* form of definability given above and the following *explicit* form: X (explicitly) defines y w.r.t. Σ iff there is a formula Φ_y s.t. $Var(\Phi_y) \subseteq X$ and $\Sigma \models (\Phi_y \Leftrightarrow y)$; when it exists, Φ_y is clearly unique up to Σ–equivalence (i.e., every Φ'_y s.t. $\Sigma \wedge \Phi_y \equiv \Sigma \wedge \Phi'_y$ holds does the job).

For instance, considering our circuit example again, we know that $\{a, b, c\} \sqsubseteq_\Sigma z$; the corresponding formula Φ_z is Σ–equivalent to $(a \vee b) \Leftrightarrow (b \Leftrightarrow c)$.

Interestingly, whenever X defines nontrivially y, the explicit definition of y from X in Σ can be derived thanks to the following result, that makes use of the notion of *forgetting* introduced by Lin and Reiter (Lin & Reiter 1994). Let us recall that $forget(\Sigma, X)$ is defined inductively by: (i) $forget(\Sigma, \emptyset) = \Sigma$; (ii) $forget(\Sigma, \{x\}) = (\Sigma_{x \leftarrow 0} \vee \Sigma_{x \leftarrow 1})$; (iii) $forget(\Sigma, X \cup \{x\}) = forget(forget(\Sigma, X), \{x\})$.

Proposition 7
$X \sqsubseteq_\Sigma y$ iff $\Sigma \models (\Phi_y \Leftrightarrow y)$, where $\Phi_y \in PROP_X$ is defined by $\Phi_y \equiv forget(\Sigma, Var(\Sigma) \setminus (X \cup \{y\}))_{y \leftarrow 1}$.

The above result proves particularly helpful when Σ is given by its prime implicates. In this situation, $forget(\Sigma, Var(\Sigma) \setminus (X \cup \{y\}))$ can be computed efficiently by selecting from $IP(\Sigma)$ the prime implicates from $PROP_{X \cup \{y\}}$ (see Lemma 8 from (Lakemeyer 1995)). Once this formula has been computed, provided that Σ is consistent, the truth value of y can be computed in linear time as the truth value of $forget(\Sigma, Var(\Sigma) \setminus (X \cup \{y\}))_{y \leftarrow 1}$ for every $\omega_X \in \Omega_X$.

Definability can be characterized in several other ways. In (Lang & Marquis 1998), we show how checking definability comes down to a deduction check thanks to Padoa's method (Padoa 1903). Hereafter, we show how definability can also be characterized by means of prime implicants (where $\bigvee PI^X_\Sigma(y)$ denotes the disjunction of all prime implicants in $PI^X_\Sigma(y)$).

Proposition 8
$X \sqsubseteq_\Sigma y$ iff $\Sigma \models (\bigvee PI^X_\Sigma(y)) \vee (\bigvee PI^X_\Sigma(\neg y))$.

Of course, in the general case $PI^X_\Sigma(y)$ and $PI^X_\Sigma(\neg y)$ can be exponentially long. However, provided that these prime implicants have been computed off-line (and that Σ is consistent), the truth value of y can be computed in polynomial time for every X–world.

Computing bases

In this section, we propose an algorithm for computing bases. The following result shows that computing a basis (resp. the set of all bases) for sets Y of variables comes down to compute it (resp. them) for each variable individually.

Proposition 9 *$X \sqsubseteq_\Sigma \{y_1, \ldots, y_p\}$ iff $\exists X_1, \ldots, X_p$ s.t. $X = X_1 \cup \ldots \cup X_p$ and $X_i \sqsubseteq_\Sigma y_i$ for every $i \in \{1, \ldots, p\}$.*

As a corollary, $\texttt{Bases}(Y)$ denoting the set of all bases for Y w.r.t Σ, $\texttt{Bases}(\{y_1, \ldots, y_p\})$ is the minimization w.r.t. set inclusion of $\{\cup_{i=1..p} B_i \mid B_i \in \texttt{Bases}(\{y_i\})\}$.

Thanks to Proposition 9, focusing on bases for individual variables is sufficient; this is the purpose of the algorithm below. Without any loss of generality, we assume that X is contained in a fixed set of "relevant" variables V^*. The utility of V^* is to focus on relevant bases, only; for instance, in a discriminability problem, V^* is the set of testable variables.

This is a greedy algorithm which considers all the variables of X in any order and throws them away when they are not necessary for forming a basis from the current set of relevant variables.

Proposition 10 *Provided that the function* `Defines` *returns* `true` *iff X defines y w.r.t. Σ, the algorithm returns a V^*-relevant basis for y w.r.t. Σ if there exists one, "failure" otherwise.*

[4]From a practical point of view, in the literature on causal reasoning, *searching* for bases is a priori useless – they are induced from the causal structure of the knowledge base.

There are several possible ways to implement the function `Defines`. In the case where the syntactic restrictions on Σ makes definability testable in polynomial time, the search for a V^*-relevant basis is itself polynomial because it consists in $|V^*|$ definability tests (plus one consistency test). In the general case, an approach to compute `Defines` that takes advantage of Proposition 8 consists in compiling Σ under the form of its prime implicants. Thus, the prime implicant lists $PI_{\Sigma}^{V^*}(y), PI_{\Sigma}^{V^*}(\neg y)$ and $PI^{V^*}(\neg\Sigma)$ (for identifying impossible V^*-worlds) are computed off-line. These lists are updated each time a variable is picked up (namely, prime implicants mentioning these variables are filtered out from the lists). Now, `Defines` makes use of this updated prime implicant lists and explores a search tree the leaves of which correspond to (partial or complete) instantiations over X, labelled by the corresponding value of y whenever possible. Not only this function checks whether X defines y but it also generates a "definition tree" for y from X.

```
Begin
If Σ is inconsistent
then return "failure";
X ← V*;
If not(Defines(X, y))
then return "failure"
else
    Z ← X;
    Repeat
        pick a x in Z;
        Z ← Z \ {x};
        if Defines(X \ {x}, y)
        then X ← X \ {x};
    Until Z = ∅;
    Return X;
End
```

Clearly enough, the worst case complexity of this algorithm is high. The contrary would be surprising, since the corresponding decision problem is hard:

Proposition 11 *(Lang & Marquis 1998)*
The results are synthesized in the following table[5]*:*

definability	standard	+ minimality
standard	coNP-complete	BH_2-complete
+ Σ consistent	BH_2-complete	BH_2-complete

Fortunately, some tractable restrictions exist. Especially, as a consequence of Propositions 33 and 34 from (Lang & Marquis 1998), our algorithm for computing a basis for y runs in time polynomial in the size of V^* plus the size of Σ whenever Σ is a set of binary clauses, or a renamable Horn formula or a DNF formula.

Hypothesis discriminability

We investigate now a notion, slightly generalizing definability, which has many practical applications ranging from fault isolation in diagnosis to decision under partial observability. Intuitively, given a set of hypotheses variables $H = \{h_1 \ldots h_n\}$, X discriminates H w.r.t. Σ if knowing the truth values of variables of X helps finding out *one of the h_i being true*. This statement may look strange – one may have preferred to read "finding out *which one of the h_i is true*". However, while for many problems hypotheses are mutually exclusive w.r.t. Σ ($\forall h_i \neq h_j, \Sigma \models \neg(h_i \wedge h_j)$) and covering all possible cases ($\Sigma \models h_1 \vee ... \vee h_n\}$, this is not always the case (see below).

Definition 3 *A* **discrimination problem** *consists in a consistent knowledge base Σ, $X \subseteq PS$ and a set of hypotheses variables $H = \{h_1, \ldots h_n\}$. X* **discriminates** *H w.r.t. Σ iff $\forall \omega_X \in \Omega_X \ \exists h \in H$ s.t. $\omega_X \wedge \Sigma \models h$. X* **discriminates minimally** *H w.r.t. Σ iff X discriminates H w.r.t. Σ and no proper subset of X does it.*

Clearly, each variable x of X corresponds to an available test, and performing this test consists in measuring the truth value of x.

There is a straightforward link between hypothesis discriminability and definability, in presence of exclusive and covering hypotheses. Indeed, in this restricted case, finding out a true variable h_i is exactly as hard as finding out the truth value of all the h_k's in H, since for any $k \neq i$ we have $\Sigma \wedge h_i \models \neg h_k$. Thus hypothesis discriminability is more general than definability. However, there is no complexity gap between both problems:

Proposition 12
HYPOTHESIS DISCRIMINABILITY *is* coNP*-complete.*

Thus, when dealing with mutually exclusive and covering hypotheses, defining families can be used to design minimal test inputs (Struss 1994) (McIlraith 1994) in order to isolate faulty components in model-based diagnosis (in this case hypotheses are candidate diagnoses, and testable variables correspond most often to available measurements). Note that McIlraith's notions of relevant or necessary tests (McIlraith 1994) have some counterparts in our framework (for instance, a necessary test corresponds to a variable without which the hypotheses space cannot be discriminated). Lastly, the algorithm for computing bases described above can be used to design conditional test policies (where tests are performed sequentially and conditioned by the outcomes of previous tests – see (Lang 1997)).

Another application of hypothesis discrimination is *decision making (and planning) under partial observability*. The logical formulation of a (one-stage) decision problem consists in a description of the initial state by a propositional formula, a fixed set $\mathcal{A}$ of available actions together with the descriptions of their (context-dependent) effects – for instance by a list of STRIPS-like expressions) and a set of goals (described for instance by a set of literals). These data enable computing, for each action a, the context h_a in which performing a certainly leads to a goal state. In order to find out a satisfying action, we need to discriminate between the h_a's, by observing enough of the initial state, knowing that some variables are measurable and some are not. This is a

[5] BH_2 (also known as DP) is the class of all languages L such that $L = L_1 \cap L_2$, where L_1 is in NP and L_2 in coNP.

discrimination problem, and without the assumption that hypotheses are exclusive nor covering all possible situations; this is important because it is generally useless to find out the truth value of all the h_a's once any of them has been shown up true – which makes it different (and easier) than computing a defining family for $\{h_a, a \in \mathcal{A}\}$. Now, having in mind that both conditional controllability and hypothesis discrimination could be applied to qualitative decision under partial observability, one may wonder why the complexities do not coincide. Why the latter is much easier then the former relies on the fact that the "context" of each action is precomputed and part of the input, but also on the structure of the decision space: the set of possible decisions is 2^X (thus exponentially large) for the former while it is fixed (and thus has a constant size) for the latter.

Conclusion

In this paper, a variety of results for conditional controllability, definability, and closely related problems such as hypothesis discrimination, have been pointed out.

Regarding computational complexity, definability appears to be much easier than controllability. The high complexity of controllability is not surprising since decision under incomplete knowledge with succinct representations (such as logic) is hard. For instance, Fargier et al. (Fargier, Lang, & Schiex 1996) give two notions of consistency of a "mixed CSP" that are close to our notions of *ceteris paribus* and unconditional controllability, and show them (respectively) Π_2^p-complete and in Σ_2^p. Our results are also related (to some extent) to recent results about the complexity of probabilistic planning with succinct representations (Littman 1997); in particular, the latter problem is PSPACE-complete if the number of stages is polynomially bounded (and EXPTIME-complete otherwise); since our notions of controllability correspond more or less to "one-stage" planning under (qualitative) uncertainty, we can expect that the complexity of controllability would climb up in the polynomial hierarchy (up to PSPACE) if polynomially many stages were allowed.

It is clear that controllability and definability are two strong forms of dependence. We believe that relating them to various notions of relevance can be useful. The companion paper (Lang & Marquis 1998) is a first step in this direction.

Acknowledgements

We would like to thank Hélène Fargier for helpful discussions about controllability. The second author has been partly supported by the "IUT de Lens" and a "Contrat d'objectifs de la Région Nord/Pas-de-Calais".

References

Beth, E. 1953. On padoa's method in the theory of definition. *Indigationes mathematicae* 15:330–339.

Boutilier, C. 1994. Toward a logic for qualitative decision theory. In *Proc. KR'94*, 75–86.

Darwiche, A., and Pearl, J. 1994. Symbolic causal networks. In *Proc. AAAI'94*, 238–244.

Darwiche, A. 1997. A logical notion of conditional independence: properties and applications. *Artificial Intelligence* 97:45–82.

Eiter, T., and Gottlob, G. 1995. The complexity of logic–based abduction. *JACM* 42(1):3–42.

Fahlman, S. 1974. A planning system for robot construction tasks. *Artificial Intelligence* 5:1–49.

Fargier, H.; Lang, J.; and Schiex, T. 1996. Mixed constraint satisfaction: a framework for decision making under incomplete knowledge. In *Proc. AAAI'96*, 175–180.

Geffner, H. 1996a. Causality, constraints and the indirect effects of actions. In *Proc. AAAI'97*, 208–222.

Geffner, H. 1996b. A formal approach for causal modeling and argumentation. In *Proc. FAPR'96*, 208–222.

Hu, A., and Dill, D. 1993. Reducing bdd size by exploiting functional dependencies. In *Proc. ACM/IEEE DAC'93*, 266–271.

Ibaraki, T.; Kogan, A.; and Makino, K. 1998. Functional dependencies in horn theories. In *Proc. AI&Math'98*.

Kautz, H.; McAllester, D.; and Selman, B. 1997. Exploiting variable dependency in local search (abstract). In *Proc. IJCAI'97 (poster session)*, 57.

Lakemeyer, G. 1995. A logical account of relevance. In *Proc. IJCAI'95*, 853–859.

Lakemeyer, G. 1997. Relevance from an epistemic point of view. *Artificial Intelligence* 97:137–167.

Lang, J., and Marquis, P. 1998. Complexity results for independence and definability in propositional logic. In *Proc. KR'98*. to appear.

Lang, J. 1997. Planning to discriminate diagnoses. In *Proc. DX'97*, 135–139.

Lifschitz, V. 1990. Frames in the space of situations (research note). *Artificial Intelligence* 46:365–376.

Lin, F., and Reiter, R. 1994. Forget it! In *Proc. AAAI Fall Symposium on Relevance*, 154–159.

Littman, M. 1997. Probabilistic planning: representation and complexity. In *Proc. AAAI'97*, 748–754.

McIlraith, S. 1994. Generating tests using abduction. In *Proc. KR'94*, 449–460.

Padoa, A. 1903. Essai d'une théorie algébrique des nombres entiers, précédé d'une introduction logique à une théorie déductive quelconque. In *Bibliothèque du Congrès International de Philosophie*, 309–365.

Papadimitriou, C. H. 1994. *Computational complexity*. Addison-Wesley.

Poole, D. 1997. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence* 94(1):7–56.

Struss, P. 1994. Testing for the discrimination of diagnoses. In *Proc. DX'94*, 312–320.