The Comparative Linguistics of Knowledge Representation

Goran Gogic *
University of California San Diego
CSE Department
La Jolla, California 92093, USA
email: goran@cs.ucsd.edu

Henry Kautz
AI Principles Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA
email: kautz@research.att.com

Christos Papadimitriou f
University of California San Diego
CSE Department
La Jolla, California 92093, USA
email: christos@cs.ucsd.edu

Bart Selman
AI Principles Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA
email: selman@research.att.com *

Abstract

We develop a methodology for comparing knowledge representation formalisms in terms of their "representational succinctness," that is, their ability to express knowledge situations relatively efficiently. We use this framework for comparing many important formalisms for knowledge base representation: propositional logic, default logic, circumscription, and model preference defaults; and, at a lower level, Horn formulas, characteristic models, decision trees, disjunctive normal form, and conjunctive normal form. We also show that adding new variables improves the effective expressibility of certain knowledge representation formalisms.

1 Introduction

Many important knowledge representation formalisms have been proposed, used, and studied during the past fifteen years, including various forms of propositional logic, nonmonotonic formalisms, decision trees, and so on. There is now a host of methods available for representing complex knowledge, and for reasoning about it. An interesting question thus arises: *How is one to evaluate and compare different knowledge representation formalisms?* Besides the practical aspect of this question with respect to choosing the "best" formalism for a given application, environment, and resource constraints, a methodology for comparing and evaluating knowledge representation methods may lead to useful introspection, new insights, and to the discovery of better approaches.

In this regard, one must consider several aspects of the desirability and effectiveness of a knowledge representation formalism:

*A part of the work was done while the author was employed at AT&T Bell Laboratories, Murray Hill. Research supported by an NSF grant.

* Research supported by an NSF grant.

(1) *Does it support efficient reasoning?* The most common use of a knowledge base is for deciding whether a statement can be inferred from the available knowledge, and hence this question is of central importance. Much research effort has been invested in recent years to clarifying this issue, and this aspect of knowledge representation is by now very well understood. In brief, all knowledge representation formalisms can be subdivided into three categories with respect to the complexity of their inference problem.

(la) Classical propositional logic, perhaps the most basic knowledge representation formalism, has an NP-complete inference problem [Cook, 1971].

(lb) More sophisticated and non-monotonic formalisms, such as circumscription, default logic, non-monotonic logic, autoepistemic logic, *etc.,* have inference problems that are even harder: complete for the second level of the polynomial hierarchy, see [Cadoli and Lenzerini, 1994; Eiter and Gottlob, 1993; Gottlob, 1992]; model preference default theories [Selman and Kautz, 1990] have even higher complexity [Papadimitriou, 1991].

(lc) Finally, weaker versions of the above formalisms provide polynomial-time inference at the expense of our next criterion, *expressibility.* These include Horn clauses [Dowling and Gallier, 1984], Horn model preference defaults [Selman and Kautz, 1990], restricted forms of default logic [Kautz and Selman, 1992], etc.

(2) An orthogonal criterion for the desirability and usefulness of a knowledge representation method is, *how expressive is it?* First we must formalize what we mean by "expressive." The most natural notion of expressiveness is provided by model theory: Since the propositional case of each of these formalisms has semantics in terms of *models* or *truth assignments,* any knowledge base can be thought of as representing a set of truth assignments, that is, of "possible worlds." This suggests a notion of equivalence: Two knowledge bases, possibly in different formalisms, are equivalent if they encode the same set

of possible worlds. Thus, here is a first try at a framework for comparing formalisms in terms of expressibility: *Consider formalism A at least as expressive as formalism B if and only if any knowledge base in B has an equivalent knowledge base in A.*

There are serious drawbacks to this proposal. First, all diverse formalisms in Ia and Ib above are trivially equally expressive, since any set of models can in principle be expressed by each of them. For example, propositional logic is exactly as expressive as the more sophisticated default logics, even the (soon to be proved) much more powerful preference defaults. Only the formalisms in (Ic) above are provably inferior; however, this is the result of a conscious sacrifice of expressibility in the interest of efficiency. As we shall point out in Section 4, where we compare sublanguages of propositional logic, it is more meaningful to compare knowledge representation formalisms in terms of their relative performance at sets of models that they can *both* express.

(3) A much more interesting, but also more subtle, question one can ask about a knowledge representation formalism is this: *How succinctly can the formalism express the sets of models that it can?* We think that this is the more interesting expressibility criterion; it is the main methodological contribution of this paper. That is, we consider formalism *A* to be stronger than formalism *B* if and only if any knowledge base in *B has an equivalent knowledge base in A that is only polynomially longer, while there is a knowledge base in A that can be translated to B only with an exponential blowup.* Using this criterion, we show that the known knowledge representation formalisms form a hierarchy (Fig. 1) which is rather surprising in its strictness, as well as in the outcomes of the particular comparisons.

(4) We should mention here that another important question is, *how does the knowledge representation formalism fare in the face of change?* Change is important in knowledge representation (for example, *nonmonotonicity* is a dynamic property). There are many formalisms in the literature for knowledge base updates and revisions; as was pointed out in [Eiter and Gottlob, 1992; Gogic *et* a/., 1994], none of the known formalisms supports efficient changes. [Gogic *et* a/., 1994] propose a tractable revision mechanism using the theory approximation technique of [Selman and Kautz, 1991; Selman and Kautz, 1996]. Incidentally, change has its own expressiveness aspect (which changes in the set *of* models can be expressed, and how succinctly?), which is not at all understood at present.

In this paper we find that the representational succinctness criterion (3) above can tell us interesting and unexpected things about familiar knowledge representation formalisms. There is a tempting argument purporting to prove that our criterion of representational succinctness (criterion 3) is just a disguise of computational complexity of inference (criterion 1 above). The argument would be this:

> *// reasoning in formalism A is computationally harder then reasoning in formalism B (i.e., there is a polynomial-time reduction from the satisfiability problem in the latter to the former) then any sentence from A can be translated (with polynomial blow up) to an equivalent sentence in B.*

This argument is wrong. The difference between reductions and representational simulations between knowledge bases is subtle but important. A reduction must be computationally efficient, and must preserve the *answer* to the satisfiability problem (the emptyness/nonemptyness aspect of the corresponding set of models), whereas representational simulations must maintain the precise set of models, and need not be computationally efficient (the simulating knowledge base need not be efficiently computable, as long as it exists). This difference manifests itself in many comparisons. For example, one of our main results is that default logic is strictly more succinct than circumscription, despite the fact that their inference problems are known to be computationally equivalent [Gottlob, 1992; Eiter and Gottlob, 1993]. For another example, characteristic models (whose satisfiability problem is trivial) can be sometimes more succinct than CNF (whose satisfiability problem is, of course, NP-complete). Finally, we know how to translate default theories to model preference defaults only in a nonconstructive way (Proposition 2). Evidently, the complexity arguments involved in such comparisons have to be much more subtle than the crude one outlined above. (However, a more sophisticated version of this "translation via complexity" argument is used in Proposition 2.)

We next highlight our results on the representational succinctness criterion (see Fig. 1 for a full depiction of our results).

1. Circumscription and default logic are more powerful with respect to representational succinctness then propositional logic (this had been observed for the case of circumscription by [Cadoli *et* a/., 1994; Cadoli *et* a/., 1995]). This result provides a "silver lining" for the high intractability of circumscription and default logic: In these formalisms one may need exponentially more succinct expressions, and thus increased intractability is not necessarily a real threat.

2. Default logic can be exponentially more succinct than circumscription. This is a rather surprising result in view of the computational equivalence of the two formalisms, and it is perhaps quite revealing of the relative power and desirability of these formalisms (heretofore indistinguishable in terms of their inference complexity). Also, model preference defaults can be exponentially more succinct than default logic, *or any other formalism whose inference problem can be done in polynomial space;* this is rather unexpected, since model preference defaults had been considered as a rather crude and unsophisticated knowledge representation formalism.

3. Lower in the hierarchy, CNF and DNF have no advantage over Horn formulas and characteristic models, when a Horn set is to be represented (recall that we are comparing knowledge representation formalisms at the intersection of their expressibility domain); this suggests that we should choose a Horn formula representation

whenever at all possible. Similarly, every Horn set has a set of characteristic models of size no larger than the one that of the best DNF representation. What is more surprising, the reverse statement does not hold, i.e. there are some functions having a short characteristic models representation but no short DNF representation. On the other hand, Horn formulas and characteristic model representations are mutually orthogonal, in that they can take exponential advantage of each other.

Some of our comparison results are not proven to hold, but are shown to be "very likely," as their refutation would disprove certain widely accepted conjectures in complexity theory. The scientific community has been trying to resolve a number of class containments problems for years (the question of whether P=NP is the best known, but certainly not the only, important question in complexity theory alluded to here), and although the final answer to each of these questions continues to be elusive, a certain amount of intuition toward the correct answer has been obtained. For example, we think that P≠NP because, roughly speaking, if there were a polynomial algorithm for satisfiability or for the traveling salesman problem, we would probably have found it.[1] Similarly, we believe that NP≠co-NP, because we have not found a polynomially succinct variant of resolution that works (or a characterization of non-Hamiltonian graphs, say).

But what about the following problem: Given a propositional formula ϕ and a subset E of the set of all its variables, is it possible to fix the variables in E so that ϕ becomes a tautology? Not only there is no known polynomial algorithm for this problem, but there is strong evidence that it does not even belong to NP or in co-NP. The class for which the problem is complete is called Σ_2, and is at the second level of the *polynomial hierarchy*. The polynomial hierarchy is extended in this way to Σ_3, Σ_4, and so on; their complementary classes are denoted by Π_2, Π_3, etc. Finally, the class of all problems solvable in polynomial space is even broader, as it contains all of the polynomial hierarchy (and probably much more). Exactly as we strongly believe that P≠NP, although we have no proof, we also strongly believe that all these levels of the polynomial hierarchy, as well as polynomial space, are distinct. That is, it is considered very unlikely that the polynomial hierarchy *collapses* at any level (the lower the level, the stronger our confidence that collapse does not occur there).

Therefore, once we show that a positive answer to a problem would imply that the polynomial hierarchy collapses, we can say that the answer to our problem *is very likely to be negative*. One well-known instance of this line of reasoning involves *non-uniform complexity classes*. Let P/poly be the set of all problems that can be solved in polynomial time by Turing machines that take, together with their input x, say of length n, an *advice string* $a(n)$, depending only on n and not on x, and of length polynomial in n. Similarly for NP/poly. Although these classes now contain unsolvable problems, it is known that they "very likely" do not extend far beyond the corresponding classes in the hierarchy. That is, it was shown in [Karp and Lipton, 1980; Yap, 1983] that, if NP is contained in P/poly, or even in co-NP/poly, then the polynomial hierarchy collapses. For more details about this and some other complexity issues mentioned in this paper see [Papadimitriou, 1993].

[1] Notice the nonmonotonic reasoning going on here; in fact, deriving results based on assumptions such as "the polynomial hierarchy does not collapse" is a fine example of nonmonotonic and counterfactual reasoning, or, equivalently, of building an extension supported by given defaults.

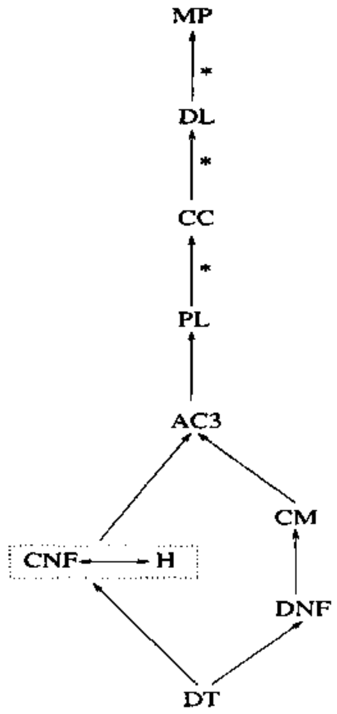

Figure 1: *The representational succinctness hierarchy. Arrows go from weaker to stronger formalisms. Two-way arrows denote representational equivalence. In arrows marked with a $*$ the lower bound ($A \not\rightsquigarrow B$) depends on the non-collapse of the polynomial hierarchy. In the uppermost arrow, our proof that $DL \rightsquigarrow MP$ requires the introduction of new variables.*

2 Definitions

In this section we introduce several well-known knowledge representation formalisms, the notion of the *size* of a knowledge base or expression in this formalism, and what it means for a bit vector to be a model of a knowledge base in the formalism.

PL is the class of all formulas in propositional logic. An assignment to the variables is *model* of a given propositional formula if it satisfies it. The size of a formula is the total number of connectives in it.

Let *CC* be the class of all circumscribed propositional formulas [McCarthy, 1980]. Vector α is a *model* of a formula F circumscribed over variables with indices from set M if α is a model of F, and no other model of F is contained in α with respect to M. The set of all such models is denoted as $CIRC_M(F)$, and when M contains all indices it is omitted and we write $CIRC(F)$. The size of a circumscribed propositional formula is that of the uncircumscribed formula.

DL is the class of all pairs (D, W) where D is a set of defaults [Reiter, 1980] and W is set of propositional formulas. We say that α is a *model* of (D, W) iff there is an extension of (D, W) satisfied by α (this is sometimes called *sceptical reasoning*). The size of a default theory is the total number of defaults in D, plus the total number of connectives appearing in W. Similarly for AEL, autoepistemic logic (this formalism is not treated extensively here).

MP is the class of all *sets of model-preference defaults* [Selman and Kautz, 1990]. A model preference default is an object of the form $(A \rightarrow b)$, where A is a set of literals and b is a literal. If D is a set of model-preference defaults, D induces a directed graph G_D on $\{0,1\}^n$ as follows: If $\alpha, \beta \in \{0,1\}^n$, then $(\alpha, \beta) \in G_D$ (we say that α *is preferred to* β *with respect to* D) if there is a default $(A \rightarrow b)$ in D such that (a) α and β are identical with respect to all variables outside b; (b) both α and β satisfy all literals in A; and (c) β satisfies b and α does not. Finally, we say that α is a model of D if whenever there is a path in G_D from α to some other bit vector β, then there is a path from β back to α. The size of a set of defaults is the total number of literals in all of its defaults.

We also examine certain important special cases of propositional logic. By DNF we denote the class of all formulas in disjunctive normal form. A formula is in disjunctive normal form if it is a disjunction of conjuncts, where conjunct is a conjunction of literals. The size of a DNF formula is the total number of connectives. A bit vector is *model* of given DNF formula if it satisfies at least one disjunct. Similarly, CNF is the class of all formulas in conjunctive normal form. The size is again the total number of connectives, and a vector is *model* of given CNF formula if it satisfies all conjuncts.

By H we denote the class of all formulas in conjunctive normal form such that each clause has at most one unnegated literal. The formalism is very popular because there is a fast algorithm for satisfiability checking of Horn formulas (see [Dowling and Gallier, 1984]) and because of its connection to logic programming. Unfortunately, not every set can be represented by Horn formulas.

CM denotes the class of all sets generated by *a set of characteristic models*. The idea originated in [Kautz *et al.*, 1993]. Since every Horn set is closed under bitwise multiplication, it makes sense to try to represent a Horn set not as the set of all its models but as the set of its "characteristic models," so that the original set is obtained as the closure of the set of characteristic models under bitwise multiplication.[2] For a set of vectors M we define closure(M) as the set of all vectors representable as bitwise product of some vectors from M. The set of models of a set of characteristic vectors M is precisely closure(M). In [Kautz *et al.*, 1993] it has been proved that H and CM are orthogonal, i.e. there are situations when one is better than the other for compact knowledge representation. However, in Section 4 we give an intriguing method for simulating CM by H. It is also proved in [Kautz *et al.*, 1993] that abduction in CM can be done in polynomial time (the same problem is NP-complete for Horn formulas). The size of a set of characteristic models $M \subseteq \{0,1\}^n$ is $|M| \cdot n$.

DT is the class of functions represented by *decision trees*. A decision tree is a binary tree such that each internal node has two edges emanating out of it —one labeled x_i, the other labeled $\overline{x_i}$. Each leaf is labeled 0 or 1. A decision tree computes a Boolean function $f(\alpha)$, where $\alpha = (\alpha_1, \ldots, \alpha_n)$, as follows: we start at the root of the tree and repeatedly follow the edge that evaluates to 1 (i.e. if $\alpha_i = 0$ then we chose edge labeled $\overline{\alpha_i}$, if $\alpha_i = 1$ we chose the edge labeled α_i). We repeat the step until we arrive at a leaf, and the label of the leaf tells us the value that function at α. An assignment is a *model* of a given decision tree if it evaluates to 1. Notice that for every α there is exactly one path that can be traversed starting from the root. The size of a decision tree is the number of nodes in it.

Finally, let AC^3 be the class of all disjunctions of CNF and conjunctions of DNF. In other words, this is the class of all depth-3 unbounded fan-in circuits with OR and AND gates, and literals as leaves. The size of an AC^3 circuit is its number of gates. This is an interesting generalization of both CNF and DNF.

Let A and B be any knowledge representation formalisms such as the ones defined above. We say that A is *at least as representationally succinct as* B, written $B \leadsto A$ if the following is true: For each knowledge base ϕ_B in B there is a knowledge base ϕ_A in A such that (a) ϕ_A and ϕ_B are defined over the same variables and have the same set of models, and (b) the size of ϕ_A is polynomially bounded in the size of ϕ_B. Notice that we have no provision that ϕ_A should be efficiently computable, give ϕ_B. Also, $A \not\leadsto B$ means that there is a knowledge base in A that has no equivalent polynomial knowledge base in B. We say that A is *representationally strictly more succinct than* B if $B \leadsto A$ and $A \not\leadsto B$ (these are the arrows in Fig. 1). We say that A and B are *representationally equivalent* if $A \leadsto B$ and $B \leadsto A$. We say that A and B are *representationally incomparable* if $A \not\leadsto B$ and $B \not\leadsto A$. Since H and CM are not complete formalisms, in that they cannot represent all sets of models but only Horn sets, comparison with other formalisms will be restricted to Horn sets. That is, we compare incomplete formalisms only at the intersection of the sets of models that they can express.

3 The Representational Succinctness Hierarchy

In this section we show that the formalisms defined above are related in terms of representational succinctness as shown in Fig. 1.

[2]The idea is later generalized in [Khardon and Roth, 1994] to capture non-Horn sets, and was successfully applied to model-based reasoning.

1. $CC \not\leadsto PL$ unless NP$\subseteq$ P/poly (which would imply that the polynomial hierarchy collapses).

Lemma 1 *For every n there is a propositional formula T_n of size polynomial in n such that to every 3CNF propositional formula F over n variables can be assigned a vector m_F (in polynomial time) such that F is unsal----able iff m_F is in $CIRC(T_n)$.*

Proof: The variables that T_n will contain are: ci,-, *ci* where $1 \leq i \leq 8\binom{n}{3}$ (*i.e.*, one c, and c_i for each possible 3-variable clause C_i), $x_1, x_2, \ldots, x_n$ (one variable for each variable from *F)* and *y* (an extra variable). We define the set of clauses *H* as $H = H_1 \cup H_2 \cup H_3$ where:

$H_1 = \{(\overline{c_i} \vee y \vee a_p \vee a_q \vee a_r) : \text{for every } C_i = (a_p \vee a_q \vee a_r)\}$ $H_2 = \{(c_i \vee c_i') : \text{for } 1 \leq i \leq 8\binom{n}{3}\}$, and $H_3 = \{(y \rightarrow a_i) : \text{for } 1 \leq i \leq n\}$

Our formula T_n will be the conjunction of clauses in *H.* For given *F* we chose m_F to have c_i, set to 1 iff C_i appears in *F,* c_i being the opposite of c,-, and $y = a_1 = \ldots = a_n = 1$.

Claim 1 *F is satisfiable iff m_F is NOT a minimal model of T.*

Proof: (=>) If *F* has a satisfying assignment *a* then we built an assignment *m'* of T_n by extending *a* with *y* — 0 and setting each c, and *c'i* like in *mp.*

(i) Each clause *C* from H_1 is satisfied because either

- c_i set to 0 or
- $c_i = 1$ which implies that C_i is in F and knowing that $C_i \subseteq C$ we have that C is satisfied.

(ii) Each clause in H_2 is satisfied by *mp* so it is satisfied by *m!*

(iii) Each clause in H_3 is satisfied because *y* = 0

We have now that *m'* is a model of T_n and it is less than *mp* which means that *mp* is not in $CIRC(T_n)$.

(<=) Suppose now that T_n has a model *m!* less than *mp.* It is easy to see that *mp* and *m'* must coincide on each Cj and c\. Notice that *m'* must have *y* = 0 (otherwise #3 forces *mp — m!)* and after plugging all c^, *c* and *y* from *m!* into *In* we will be left with some clauses yet to be satisfied. Those are exactly the clauses from *F* and they can be satisfied only with a proper choice of a i , . . . , a_n which means that *F* is satisfiable. ■

This proves the lemma. ■

Suppose now that for any formula *T* there is a poly-size formula *V* such that *CIRC(T)* - T', *i.e.,* the set of models for *T'* is the set of minimal models for *T.* From Lemma 1 we can then conclude that any problem from co-NPhas a poly-size circuit in the following way: reduce given instance of co-NPproblem to the question of whether a formula *F* is a tautology (which is a co-NPcomplete problem,) find m_F and T_n from the lemma, take then T' and check whether m_F satisfies T' (which can be done in polynomial time). Therefore, our supposition would then imply co-NP$\subseteq$ P/poly, which implies that NP$\subseteq$ P/poly because P/poly is closed under the complement. This theorem was proven independently in [Cadoli et al., 1995]. The above proof builds on a construction introduced in [Kautz and Selman, 1992], and our result strengthens an earlier result in [Cadoli *et al.*, 1994].[3]

2. $CC \leadsto DL$

Proof: Let us take a propositional formula T with variables $\{p_1, p_2, \ldots, p_n\}$. We define default logic $\Delta = (D, W)$, with $W = T$ and $D = \{d_i = \frac{:\overline{p_i}}{\overline{p_i}} : 1 \leq i \leq n\}$. Suppose now that $m \in CIRC(T)$. Then there is an extension obtained by applying those d_i's for which the i-th bit of m is 0, and that extension has only one model, m. No more defaults can fire because m would not be the minimal model. The other direction follows the same line: if S is an extension of Δ then it has only one model (the one that has bit i set to 0 only if d_i has fired), and that model is minimal in T. This reduction belongs to the folklore of NMR community. ■

3. $DL \not\leadsto CC$ unless NP $\subseteq$ co-NP/poly (which would imply that the polynomial hierarchy collapses).

Proof sketch: Because of space limitations, we can only provide an outline of the proof.

Definition: A formula in CNF is called *pure* if each clause has at most three literals and either all of them are negative or all of them are positive. The problem of deciding whether a pure formula is satisfiable is known to be NP-complete [Garey and Johnson, 1979].

We next obtain a lemma concerning the complexity of model checking in default logic.

Lemma 2 *For every n there is a default logic theory $\Delta_n = (D_n, W_n)$ such that to every pure formula F over n variables can be assigned (in polynomial time) a model m_F that belongs to some extension of Δ_n iff F is satisfiable.*

Proof: Assume that $F(a_1, \ldots, a_n)$ is a pure formula. Our default theory will be formed over a set of variables $c_1, \ldots, c_k, b, z_1, z_2, \ldots, z_n$, with $k = 2\binom{n}{3}$. To simplify the presentation we use X_i to stand for the formula

$b \wedge z_1 \wedge \ldots \wedge z_{i-1} \wedge \overline{z_i} \wedge z_{i+1} \wedge \ldots \wedge z_n$ for $1 \leq i \leq n$

and Y_i for

$\overline{b} \wedge z_1 \wedge \ldots \wedge z_{i-1} \wedge \overline{z_i} \wedge z_{i+1} \wedge \ldots \wedge z_n$ for $1 \leq i \leq n$

Now we define a set D of defaults:

$$d_i^T = \frac{:X_i}{Y_i} \text{ and } d_i^F = \frac{:Y_i}{X_i} \text{ for } 1 \leq i \leq n$$

[3]We thank Marco Cadoli for useful discussions on the issue of the size of representations.

For every positive clause $C_i = (a_{i_1} \vee a_{i_2} \vee a_{i_3})$ we introduce default d_i^1:

$$\frac{\overline{Y_{i_1}} \vee \overline{Y_{i_2}} \vee \overline{Y_{i_3}} : c_i}{c_i}$$

For every negative clause $C_i = (\overline{a_{i_1}} \vee \overline{a_{i_2}} \vee \overline{a_{i_3}})$ we introduce default d_i^1:

$$\frac{\overline{X_{i_1}} \vee \overline{X_{i_2}} \vee \overline{X_{i_3}} : c_i}{c_i}$$

For each clause C_i we will also introduce a default $d_i^0 = : \overline{c_i}/\overline{c_i}$. We take

$$W = X_1 \vee \ldots \vee X_n \vee Y_1 \vee \ldots \vee Y_n \vee (b \wedge z_1 \wedge \ldots \wedge z_n)$$

Our lemma now follows directly from the following claim (details appear in full version of paper).

Set c_i to 1 if C_i is in F and to 0 otherwise. Then F is satisfiable if and only if $m = c_1 c_2 \ldots c_k 11 \ldots 1$ is in some extension of (D, W). ∎

Suppose now that for any default logic Δ there is a formula whose circumscription produces the same set as the union of extensions for Δ. Then by Lemma 2 we conclude that checking satisfiability of every pure formula of fixed size can be reduced to checking whether a given model belongs to circumscription of a fixed formula, which is in co-NP. (Note that to show that a model is *not* a minimal model, one simply has to provide a model that is contained in the original model). In other words, it would imply that NP⊆ co-NP/poly (the fixed formula is the advice $a(n)$ in the definition of the P/poly etc. classes [Karp and Lipton, 1980]). ∎

4. $MP \not\leadsto DL$ unless the polynomial hierarchy collapses.

We need the following lemma.

Lemma 3 *Fix M to be any Turing machine that operates in linear space. For every n there is a set of model preference defaults Δ_n such that to every input x of length n we can assign (in polynomial time) a model m_x such that m_x is a model of Δ_n if and only if M accepts x.*

Proof: Assume without loss of generality that M, after accepting, restores its input tape to the initial configuration (which it has saved) and starts its computation anew on the same input; when it rejects, it halts. Suppose that there are at most 2^k symbol-state combinations in M, and p moves (each move is of the form "if three contiguous symbol-state combinations are (a, b, c), then replace them by (a', b', c')". We can assume that in any reachable configuration only one such move applies. We shall represent any configuration of M with n tape squares as $2nk + p$ Boolean variables $x_{ij}, y_{ij}, i = 1, \ldots, n, j = 1, \ldots, k$ and $z_j, j = 1, \ldots p$. The x variables will represent the tape contents; the y variables will ordinarily be equal to the x variables but will facilitate the implementation of the moves; and the z variables will be ordinarily zero, unless a move is in progress, in which case the z variable corresponding to the move being implemented will be one.

We shall next define the set of defaults Δ_n that simulates the moves of M. That is, if a configuration C of M yields another C', then in the graph G_{Δ_n} there is an arc from bit vector C to bit vector C'. Suppose that the qth move of M is of the form $(a, b, c) \Rightarrow (a', b', c')$. Since the total number of symbols is at most 2^k, we can think of a etc. as k-tuples of bits. Let $x_r[a, b, c]$ be the set of literals stating that the values of x_{ij}, with $i = r-1, r, r+1$ and $j = 1, \ldots, k$, spell abc, and let $y_r[a, b, c]$ be the set of literals stating that the values of y_{ij}, $i = r-1, r, r+1, j = 1, \ldots, k$ spell abc. Then we add the following defaults to Δ_n:

For each i, $1 < i < n$, we have a default $(x_r[a, b, c], y_r[a, b, c], \bar{z} \rightarrow z_q)$ (this default sets the bit z_q that says "move q in progress"). We have $3k$ defaults that change, one-by-one, the y-bits corresponding to squares $i-1$ to $i+1$ from (a, b, c) to (a', b', c'), whenever move q is in progress (we omit the straightforward details of these defaults). Similarly, we have $3k$ more defaults that copy the (a', b', c') from the y bits back to the x bits. Finally, a default resets z_q to zero, whenever $x_r[a', b', c']$, $y_r[a', b', c']$, and only z_q among the z'z is one: The move has been implemented.

Let x be an input of length n of M, and let m_x be the starting configuration of M on input x (and the corresponding bit vector in our x, y, z variables). It is easy to see that the following holds: *The part of the graph G_{Δ_n} reachable from m_x is either a cycle or a path, depending on whether M accepts the input or not.*

Hence m_x is a model of Δ_n if and only if M accepts x, concluding the proof of the lemma. ∎

Hence, if for any given set of model preference defaults there is an equivalent default theory (or autoepistemic logic theory, etc.), this would mean that there is a Σ_2 method for simulating any polynomial-space machine, which would collapse the hierarchy (polynomial space includes all levels of the polynomial hierarchy).

Notice that we have not proved that $DL \leadsto MP$; we do not know how to directly simulate default theories by model preference defaults. However, in Section 5 we show something slightly weaker but almost as compelling: MP can simulate any knowledge representation formalism whose model-checking problem is solvable in polynomial space *by adding extra variables* —and this includes default logic, autoepistemic logic, and a host of others [Gottlob, 1992; Eiter and Gottlob, 1993]. On the other hand, 4 holds even if adding extra variables is allowed (because polynomial space is very likely more powerful than any level of polynomial hierarchy), which gives evidence that MP is in some sense stronger than default logic.

4 Sublanguages of propositional logic

Although some of the following relations follow from well-known properties of propositional logic, proofs of all are provided for completeness.

5. $H \not\rightsquigarrow CM$, $H \not\rightsquigarrow DNF$

Proof: Let $f(x_1, x_2, \ldots, x_{2n}) = (\overline{x_1} \bigvee \overline{x_2})(\overline{x_3} \bigvee \overline{x_4}) \ldots (\overline{x_{2n-1}} \bigvee \overline{x_{2n}})$. This function is already given as a conjunction of n Horn clauses so it has a short representation in H . Notice that any set of characteristic models for f must contain the maximal models of f and it is easy to see that f has 2^n maximal models (for any $i \in \{1, \ldots, n\}$ we can take $x_{2i-1} = 0$, $x_{2i} = 1$ or $x_{2i-1} = 1$, $x_{2i-1} = 0$). The size of its DNF representation is also exponential, because it is a monotone formula and any term we get by multiplying one variable from each clause is a prime implicant (so there are 2^n prime implicants). So, f has a polynomial representation in H whereas every representation of f in CM or DNF is of exponential size. ∎

6. $CM \not\rightsquigarrow DNF$

Proof: Let $f(x_1, x_2, \ldots, x_{2n}) = (\overline{x_1} \bigvee x_2)(\overline{x_3} \bigvee x_4) \ldots (\overline{x_{2n-1}} \bigvee x_{2n})$. Let $k \leq n$, and let $1 \leq i_1 < \ldots < i_k \leq n$ be a given set of indices. Then we define the vector $e^n_{i_1,\ldots,i_k}$ to have bits $i_1, \ldots, i_k$ equal to 0 and all other bits equal to 1.

The set of characteristic models that generates f is

$$\{e_{2i-1} : 1 \leq i \leq n\} \bigcup \{e_{2i-1,2i} : 1 \leq i \leq n\} \bigcup \{1^{2n}\}$$

(it has $2n+1$ vectors). By the same line of reasoning as in the previous proof we can conclude that the DNF representation of f will have exponential size. ∎

7. $CM \not\rightsquigarrow H$, $DNF \not\rightsquigarrow H$

Proof sketch: [Kautz *et al.*, 1993] use the function $f(x_1, x_2, \ldots, x_{2n}) = (\overline{x_1} \wedge \overline{x_2}) \vee \ldots \vee (\overline{x_{2n-1}} \wedge \overline{x_{2n}})$ to separate CM from H. f has a short DNF and CM representation. The function is dual to the function used in **5**, and by applying the same idea we conclude that f must have an exponential size representation in CNF , and therefore in H. ∎

8. $CNF \rightsquigarrow H$, $H \rightsquigarrow CNF$

Proof: Let M be a Horn set of models. We will prove that shortest CNF representing M must be a Horn formula. Suppose the shortest CNF representing M contains a non-Horn clause C. Then there is a Horn clause $C_H \subseteq C$ that can be entailed from M (see [Selman and Kautz, 1991; Selman and Kautz, 1996] for the proof) and we can put C_H in the CNF instead of C (the set of models will not change). This gives an even shorter formula representing M; contradiction. ∎

9. $DNF \rightsquigarrow CM$

The proof is given in the full version of the paper. Note that since CM cannot represents all possible sets of models, we consider here only sets of models representable in both formalisms. The relationship between DNF and CM has also been analyzed in [Khardon and Roth, 1994] in a slightly different setting.

10. $DT \rightsquigarrow DNF$, $DT \rightsquigarrow CNF$, $DNF \not\rightsquigarrow DT$, $CNF \not\rightsquigarrow DT$

Proof sketch: In the full version of the paper, we show how to construct a short CNF formula and a DNF formula directly from a given decision tree. It follows that each polynomial size decision tree has both a short DNF and a short CNF representation. Since there exist short DNF formulas that do not have a short CNF representation, it follows that there cannot exist a short decision tree for such formulas, and thus $DNF \not\rightsquigarrow DT$. Similarly, $CNF \not\rightsquigarrow DT$. ∎

11. $AC^3 \rightsquigarrow PL$, $CNF \rightsquigarrow AC^3$, $CM \rightsquigarrow AC^3$, $PL \not\rightsquigarrow AC^3$, $AC^3 \not\rightsquigarrow CNF$, $AC^3 \not\rightsquigarrow CM$

Proof sketch: The first two relations follow directly from the definition of AC^3 . For the third observation, we can show that, given a set of characteristic models M, one can construct a short formula in AC^3 that represents M. From a result in [Hastad, 1986], it follows that the parity function does not have a short AC^3 representation. Since the parity function has a short encoding in PL , it follows that $PL \not\rightsquigarrow AC^3$. Combining **8.** and **9.** we get $DNF \not\rightsquigarrow CNF$ that AC^3 is a generalization of DNF we conclude that $AC^3 \not\rightsquigarrow CNF$. Similarly, AC^3 is a generalization of H which combined with **5.** gives $AC^3 \not\rightsquigarrow CNF$. ∎

5 Adding Extra Variables

In this section, we show that adding extra variables improves the representational succinctness of certain formalisms. We state our results here without the proofs. Our first result shows that by introducing additional variables, Horn formulas can efficiently encode sets of characteristic models. This result should be contrasted with that obtained in [Kautz *et al.*, 1993], where it is shown that without the introduction of extra variables there are sets of models that have short characteristic model encodings but no short Horn formula encoding.

Definition: For a set S of m-bit vectors and given number $n \leq m$ we define $proj_n(S)$ as the set of n-bit vectors $x = (x_1, \ldots, x_n)$ for which there exists a vector $s = (s_1, \ldots, s_m) \in S$ such that $x_i = s_{m-n+i}$ for $1 \leq i \leq n$.

Proposition 1 *For every set $M = \{m^1, \ldots, m^k\}$ of n-bit characteristic models there is a set S with a Horn representation of size polynomial in $n + k$ such that* $proj_n(S) = \text{closure}(M)$.

We can in fact also show that using additional variables is *provably* more powerful. That is, we can prove that by adding extra variables it is possible to represent sets that otherwise require an exponential representation in H and CM .

Finally, we consider the power of model preference defaults, when we allow for additional variables. Here we obtain a surprisingly general result:

Proposition 2 *Let A be any knowledge representation formalism such that the model-checking problem of A can be carried out in polynomial space in the number of variables and the size of the representation. Then for any knowledge base K in A on n variables and representation size s there is a set of model preference defaults* A *of size at most polynomial inn + s such that the projection of the set of models of A to the original n variables is precisely the set of models of K.*

As noted earlier, this result implies that by allowing for additional variables *MP* can simulate any knowledge representation formalism whose model-checking problem is solvable in polynomial space. Default logic and autoepistemic logic are just two examples of such formalisms [Gottlob, 1992; Eiter and Gottlob, 1993].

6 Conclusions

Knowledge representation formalisms are usually compared with respect to their computational properties and expressive power. Expressive power is characterized in terms of what can and cannot be represented in a formalism. Often little consideration is given to the question to what extent the formalisms allow for a compact encoding of information. We presented a series of results showing that systems with similar expressive power can differ dramatically in the size of the shortest encoding of certain kinds of information.

Fig. 1 summarizes our main results. Each upward arrow leads to a provably more succinct representation formalism. (Some results are based on certain standard complexity theoretic assumptions.) For example, we have shown that there exist sets of models with short (polynomial) encodings in default logic, but that can only be captured by exponential size circumscriptive theories. On the other hand, however, any set of models with a compact encoding using circumscription can also be captured by a short default logic theory. One surprising aspect of our analysis is that we found many strict separations between formalisms. This suggests that succinctness is indeed useful dimension along which to compare representation formalisms.

References

[Cadoli *et ai,* 1994] M. Cadoli, F.M. Donini, M. Schaerf. Is intractability of non-monotonic reasoning a real drawback. *Proceedings ofAAAI,* 1994.

[Cadoli *et al,* 1995] M. Cadoli, F.M. Donini, M. Schaerf. On compact representations of propositional circumscription. *Proceedings of STAGS,* 1995.

[Cadoli and Lenzerini, 1994] M. Cadoli, M. Lenzerini. The complexity of propositional closed world reasoning and circumscription. *Journal of Computer and System Sciences,* 2:255-310, 1994.

[Cook, 1971] S.A. Cook. The complexity of theorem proving preocedures. *Proceedings of STOC,* 151-158, 1971.

[Dowling and Gallier, 1984] W.F. Dowling, J. Gallier. Linear time algorithms for testing the satisfiability of propositional Horn formula. *Journal of Logic Programming,* 3:267-84, 1984.

[Eiter and Gottlob, 1992] T. Eiter, G. Gottlob. On the Complexity of Propositional Knowledge Base revision, Updates, and Counterfactuals. *Artificial Intelligence,* 57(3):227-270, 1992.

[Eiter and Gottlob, 1993] T. Eiter, G. Gottlob. Propositional circumscription and extended closed-world reasoning are Π_2-complete. *Theoretical Computer Science,* 114:231-245, 1993.

[Gottlob, 1992] G. Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation,* 2:397-425,1992.

[Garey and Johnson, 1979] M.R. Garey, D.S. Johnson. Computers and Intractability: A Guide to the Theory of A^P-completness. W.H.Freeman and Company, 1979.

[Gogic *et ai,* 1994] G. Gogic, C.H. Papadimitriou, M. Sideri. Incremental recompilation of knowledge. *Proceedings of AAAI,* 1994.

[Hastad, 1986] J. Hastad. Improved lower bounds for small depth circuits. *Proceedings of STOC,* 6-20, 1986.

[Kautz *et ai,* 1993] H.A. Kautz, M.J. Kearns, B. Selman. Reasoning with characteristic models. *Proceedings of AAAI,* 1993.

[Kautz *et ai,* 1994] H.A. Kautz, M.J. Kearns, B. Selman. Horn Approximations of Empirical Data. *Artificial Intelligence,* 1994.

[Kautz and Selman, 1991] H.A. Kautz, B. Selman. Hard Problems for Simple Default Logics (Expanded Version). *Artificial Intelligence,* 49:243-279, 1991.

[Karp and Lipton, 1980] R.M. Karp, R.J. Lipton. Some conections between nonuniform and uniform complexity classes. In *Proceedings of STOC,* 302-309, 1980.

[Khardon and Roth, 1994] R. Khardon, D. Roth. Reasoning with models. *Proceedings of AAAI,* 1994.

[Kautz and Selman, 1992] H.A. Kautz, B. Selman. Forming concepts for fast inference. *Proceedings of AAAI,* 786-793, 1992.

[McCartny, 1980] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence,* 13:27-39, 1980.

[Papadimitriou, 1991] C.H. Papadimitriou. On selecting a satisfying truth assignment. *Proceedings of FOCS,* 1991.

[Papadimitriou, 1993] C.H. Papadimitriou. *Computational complexity.* Addison Wesley, 1993.

[Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence,* 13:81-132, 1980.

[Selman and Kautz, 1990] B. Selman, H.A. Kautz. Model-preference default theories. *Artificial Intelligence,* 45:287-322,1990.

[Selman and Kautz, 199l] B. Selman, H.A. Kautz. Knowledge compilation using Horn approximations. *Proceedings of AAAI,* 1991.

[Selman and Kautz, 1996] B. Selman, H.A. Kautz. Knowledge compilation and theory approximation. *Journal of ACM,* to appear.

[Yap, 1983] H.P. Yap. Some consequences of nonuniform conditions on uniform classes. *Theoretical Computer Science,* 26:287-300, 1983.