ON SELECTING A SATISFYING TRUTH ASSIGNMENT

(Extended Abstract)

Christos H. Papadimitriou†

ABSTRACT: *We study the complexity of certain natural generalizations of satisfiability, in which one of the possibly exponentially many satisfying truth assignments must be selected. We consider two natural selection criteria,* default preference *and* minimality *(circumscription). The resulting computational problems are quite novel and intriguing. The thrust of our complexity results seems to be that hard problems become harder, while easy problems remain easy. Interestingly, this consideration also yields as a* byproduct a new and very natural polynomial–time randomized algorithm for 2SAT.

1. INTRODUCTION

The satisfiability problem for Boolean formulae in conjunctive normal form needs no introduction. It is perhaps one of the most fundamental and familiar computational problems. It is the archetypical NP–complete problem, and its polynomial special cases (e.g., when the clauses either are all Horn or all contain two literals) are well–known.

In this paper we study certain intricate generalizations of satisfiability, motivated by important current considerations in the study of common–sense reasoning (see the "Motivation" subsection at the end of this introduction). Given a Boolean formula Γ in conjunctive normal form, we are interested in *selecting one* among its potentially exponentially many satisfying truth assignments, according to some criterion. We examine two natural and important selection criteria. *Non–dominance with respect to specified defaults* was proposed and initially investigated in [SK]; *minimality* or *circumscription* is a more established notion of model "goodness" (we use the terms "model" and "satisfying truth assignment" interchangeably).

† Department of Computer Science and Engineering, University of California at San Diego. Research supported by the ESPRIT Basic Research Action No. 3075 ALCOM and by the NSF.

We introduce defaults first. A *default* is an object of the form $(x\overline{y}\overline{z} \xrightarrow{D} w)$, with possibly several literals before the $\xrightarrow{D}$ arrow and only one after. For example, the above default may encode the real–life heuristic rule that "if A is a graduate student, and not independently wealthy, and not a TA, then it can be assumed that (s)he is an RA." The semantics of this default are that any model of the world in which x holds, y and z do not, and w holds, is to be preferred over the model in which all is the same except that w fails to hold. Thus, a set of defaults Δ defines a directed graph over all possible models, that is, a directed subgraph of the hypercube. Each arc comes from a default, whose premises are satisfied in both the head and the tail, and the conclusion is satisfied by the head but not by the tail. If we are also given a formula Γ, the *preference graph* $G(\Gamma, \Delta)$ is the directed graph defined above restricted to the nodes that are models of Γ.

It should be emphasized here that $\xrightarrow{D}$, the "then it can be assumed by default" symbol, behaves differently than logical implication, in that *literals do not migrate freely over it, just switching signs.* For example, the default $(x\overline{y}\overline{z} \xrightarrow{D} w)$ is *not* equivalent to $(x\overline{w}\overline{z} \xrightarrow{D} y)$ (recall the example above). But having said that, it is clear that there is a close relationship between clauses and defaults. For example, the set of all three defaults $(\overline{x}\overline{y} \xrightarrow{D} z)$, $(\overline{x}\overline{z} \xrightarrow{D} y)$, $(\overline{y}\overline{z} \xrightarrow{D} x)$ clearly corresponds to the clause $(x \vee y \vee z)$. We call a set of defaults Δ *clausal* if it is the union of all defaults derived this way from a set of clauses.

Thus, we are given a formula Γ in conjunctive normal form, and a set Δ of defaults. If M and M' are truth assignments that satisfy Γ, we write $M \geq M'$, if there is a path from M to M' in $G(\Gamma, \Delta)$[1]. We say that M *dominates* M' if $M \geq M'$ and it is not

[1] [SK] define $\geq$ in a way that allows intermediate nodes in the path from M to M' not satisfying Γ. Since in our motivation satisfying Γ is the more fundamental property, we prefer the present formalism, independently proposed by [Se].

the case that $M' \geq M$. Finally, a model M of Γ is *undominated* if there is no other model that dominates it; in other words, if it is within a *sink strongly connected component* of $G(\Gamma, \Delta)$. This is our proposed notion of "goodness" of a model. Intuitively, an undominated model conforms to all of our default assumptions as much as possible, and thus presumably it will be more useful in refuting deductions. As we shall see in Section 4, undominated models turn out to be precisely those one is likely to arrive at by applying applicable defaults at random. The computational problem we study, the *undominated model problem*, is this:

Given a formula Γ and a set of defaults Δ, find an undominated model (obviously at least one always exists, if Γ is satisfiable).

The undominated model problem is obviously a generalization of satisfiability (just take $\Delta = \emptyset$). But there is a slightly subtler observation to be made here: If there is no Γ, and Δ is clausal, then again the undominated model problem is a form of satisfiability: If a satisfying truth assignment of the clauses that yield Δ exists, then the only undominated models are satisfying truth assignments. This immediately implies that the undominated model problem, even with no Γ, is NP–hard (see Lemma 1 in Section 2; this was already observed in [SK]).

In Section 2 of this paper we study the complexity of the undominated model problem in depth. Our main complexity result establishes that the undominated model problem is PSPACE–complete *even if Γ is empty.* Our main algorithmic result is an $O(n^2)$ algorithm for solving the undominated model problem with no Γ and with Δ consisting of 2–defaults (all premises have one literal). Our algorithm is quite involved (it solves a major generalization of 2–satisfiability). It exploits the graph–theoretic nature of 2–defaults in a novel way, and is based on the concept of *immortal* literals. It can be extended to the cases in which Γ consists of atoms, and 2–clauses. Other complexity and algorithmic results fill in all entries of the 5×5 table of special cases of the problem with respect to clauses and defaults (none, atoms, 2–literal, Horn, or general); see Table 1 at the end of this paper.

Minimality is a very different notion of goodness for models: It requires that models be conservative, with as few true variables as possible, at least among certain distinguished variables. In Section 3 we prove two results for the problem of minimal model selection: It is Δ_2^p–complete in general, and in NC for the case of 2–satisfiability (it is immediate that the 2–satisfiability case is in P, as is the case of Horn formulae, for which the well–known greedy algorithm automatically yields minimal models).

There is a natural probabilistic point of view we can take about model selection. If we consider the preference graph as a Markov chain, model selection corresponds to finding *any ergodic (non–transient) state of this chain.* In fact, the randomized algorithm suggested by this chain (follow applicable defaults at random, repeat long enough) is arguably much more plausible as a model of a component of human reasoning. Unfortunately, we show that this algorithm may require (for certain starting points) exponentially long to converge to an undominated model in the case of 2–defaults (in the general, PSPACE–complete case, of course, there was never any hope). The more interesting result, however, concerns the same algorithm applied to *clausal* default sets. In this case the algorithm takes the following appealing form:

Start with any truth assignment. While there are unsatisfied clauses, pick any one, and flip a random literal in it.

We show in Section 4 that *this algorithm solves 2–satisfiability in $O(n^2)$ expected time.* The proof involves an aggregation of the states of the Markov chain so that the chain is mapped to the *gambler's ruin* chain. The result remains true even if two of the three choices (that of an initial truth assignment, and an unsatisfied clause at each step) are made by an adversary. This result is not exactly silly, even in view of the known linear–time and NC algorithms for this problem. Significantly, the above algorithm is extremely simple and natural, and thus a plausible scheme for modeling reasoning.

What is potentially more important, the same algorithm, or natural variants, seem to solve all kinds of other polynomial special cases of satisfiability. For example, we conjecture that it also solves the polynomial special case in which all clauses have three literals, and no variable appears more than three times, while a variant solves the case of Horn clauses. An important question thus arises: What are the limitations of this simple unified technique for solving satisfiability problems in randomized polynomial time? Are there new special forms of satisfiability that can be thus attacked? This and other open problems are discussed in Section 5.

MOTIVATION: Common–Sense Reasoning

It has been convincingly argued [Le] that human reasoning functions in two quite distinct modes. The *puzzle* mode supports complex decisions in sophisticated contexts using elaborate combinatorial reasoning as in puzzle–solving and game–playing. The far more common *routine* mode specializes in easy deductions like "I am in a city, so the roar I hear is not a tiger's." Such deductions are made rapidly and at a massive rate. The power of human reasoning, it can be argued, derives from the presence and interaction of these two modes (and, of course, all the shades in between). On the other hand, for the purpose of formal study, reasoning is usually abstracted as a deduction problem of the form "$\Gamma \models \phi$," where presumably Γ is a massive formula encompassing the reasoner's experience to–date, and ϕ is a smaller formula capturing the situation at hand. As this is equivalent to the problem of testing $\Gamma \wedge \neg\phi$ for satisfiability, such formalisms seem to model the puzzle mode, rather than the routine mode, of reasoning. It was proposed in [Le] that, in order to understand better the routine mode, we should come up with algorithms that *preprocess* Γ to obtain a data structure, called the *vivid form of* Γ. Fast heuristics can then be applied to the vivid form in order to quickly test (possibly with infrequent errors) whether $\Gamma \models \phi$, for any new ϕ arising in the rapidly changing situations of life[2].

Given a Boolean formula Γ, what kind of vivid form would favor rapid heuristic deductions? One appealing idea is that the vivid form of Γ need only be a *model* M of Γ, that is, a truth assignment that satisfies Γ. Then, one can immediately demonstrate that $\Gamma \not\models \phi$ by simply verifying that $M \not\models \phi$, a problem that can be rapidly solved. Naturally, one–sided errors are possible in this proposal. But they can be checked either by maintaining *many models*, and —finally arriving at the problem addressed in this paper— by *carefully selecting the models maintained* in order to maximize their usefulness in refuting deductions. That is, given a Boolean formula, we wish to select among its potentially exponentially many models one that is, informally, the most likely model of the real world, and thus presumably the most useful in falsifying false propositions.

As might be expected, there is no agreement as to what this means exactly. The two formalisms of model selection studied in this paper represent two important and technically interesting hypotheses, one recently proposed and studied, and the other well–established: *Default preference* [SK], where default rules are used to formalize model goodness, and *minimality* or *circumscription* [Mc, KP], in which "good model" means "model with few true variables" (the motivation for the latter is that we expect formulae ϕ above to be largely monotonic, and thus easier to disprove by models poor in true variables).

Our results, viewed from the point of view of the motivating application, can be stated thus: (a) In general, the addition of selection criteria make the complexity obstacles to satisfiability much more severe (Theorems 1 and 3). (b) However, solvable cases of the problem remain easy, even in the presence of non–trivial selection criteria (Theorems 2 and 4). (c) A randomized algorithm has been shown to work efficiently in certain cases (Theorem 5 and subsequent discussion). Furthermore, this algorithm is so simple and natural, that it is perhaps the first credible algorithmic model of common–sense reasoning; the full extent of its applicability is not yet understood.

2. UNDOMINATED MODELS

We are given a set (possibly empty) of defaults Δ, and a formula Γ in conjunctive normal form (also possibly empty). The directed graph $G(\Gamma, \Delta)$ has as nodes all models of Γ, and an edge from model M to M' if there is a default $(\alpha \xrightarrow{D} \lambda) \in \Delta$ where λ is a literal and α a set of literals, such that (a) M and M' both satisfy α, (b) M' satisfies λ, and M does not, and (c) M and M' differ only in λ. The undominated model problem is this: Given Γ and Δ, find a model in a sink strongly connected component of $G(\Gamma, \Delta)$. As seen in Table 1, the complexity of the problem depends heavily on the special form of Γ and Δ.

Let Γ be a set of clauses. The corresponding set of defaults is $\mathrm{DEF}(\Gamma) = \{(\alpha \xrightarrow{D} \lambda) : (\alpha \rightarrow \lambda) \in \Gamma\}$. Notice that each clause in Γ gives rise to as many defaults as it has literals. A set of defaults Δ is called *clausal* if there is a set of clauses Γ such that $\mathrm{DEF}(\Gamma) = \Delta$. The following is not hard to prove:

Lemma 1. Let Γ be a set of clauses. Then Γ is satisfiable if and only if all undominated models of $G(\emptyset, \mathrm{DEF}(\Gamma))$ satisfy Γ. $\square$

Corollary [SK]. It is NP–hard to find an undominated model in $G(\emptyset, \Delta)$. $\square$

[2] **Incidentally, for such algorithms to be plausible models of human reasoning, they should also be extremely simple and natural, an unfamiliar and novel criterion in algorithm design briefly discussed in [PY], and taken up again in Section 4 of the present paper.**

We show the following:

Theorem 1. It is PSPACE–complete to find an undominated model in $G(\emptyset, \Delta)$.

Sketch: The problem is in NPSPACE (even for general Γ) because an undominated model, if it exists, can be guessed, and then it can be checked (reusing space) that all other models are either unreachable from it in $G(\emptyset, \Delta)$, or are in the same strongly connected component with it.

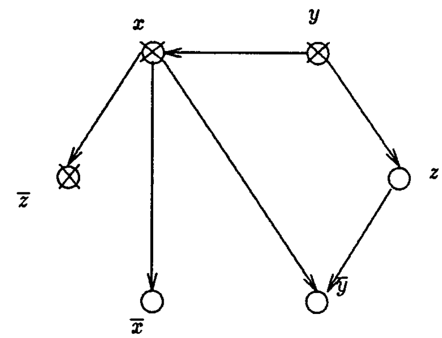
The PSPACE–completeness construction is in general terms the following: We consider a deterministic linear–bounded Turing machine M that stops after precisely 2^n steps having erased its tape. Consider the configuration graph $G(M, x)$ on input x. The initial, accepting, and rejecting configurations are denoted I, A, and R. We replicate $G(M, x)$ by having two copies of each configuration C, say $0C$ and $1C$. Then there is an arc from $0A$ and $1A$ to $0I$, and from $0R$ and $1R$ to $1I$. All configurations not on the computation path are led to the corresponding $0I$ or $1I$ as well. In this graph, if $0C$ is in a sink strongly connected component, then the machine accepts x, and if $1C$ is in a sink strongly connected component the machine rejects x. It takes some more construction to make this graph into a preference graph of an appropriate set of defaults, thus proving the Theorem. □

In contrast, we can prove the following:

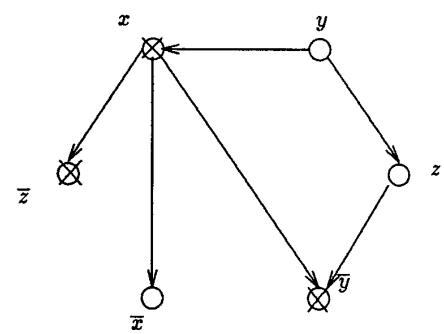
Theorem 2. If Γ and Δ consist of 2–clauses and 2–defaults, respectively, we can find an undominated model in $G(\Gamma, \Delta)$ in $O(n^2)$ time.

Sketch: We only sketch the case of $\Gamma = \emptyset$. The problem can be rethought as follows: We are given a directed graph whose nodes are the n variables and their negations, and the arcs are the defaults in Δ. A *state* S is a set of n nodes not containing both a variable and its negation (that is, a state is a truth assignment). We represent pictorially a state by marking all nodes in it (see Figure 1(a)). State transitions happen by selecting an arc going from a marked node to an unmarked one, and marking the head (thus unmarking its negation), see Figure 1(b). We are seeking a state that is in a sink component of this transition diagram.

Call a node *live* in a state if there is a reachable state containing the node (equivalently, if there is a path from a marked node to it). Call a node *permanent* if all reachable states contain it. Thus a node is permanent iff its negation is not live. Call a node *immortal* if it is live at all reachable states. Finally, call a state *stable* if all live nodes in it are immortal. Stable states are a subtle concept, but a complex yet useful characterization is possible:



(a)



(b)

Figure 1: States and transitions.

Lemma 2. We can test in $O(n^3)$ time whether a state is stable. □

More useful for our purposes is a construction that gets us a special kind of a stable state:

Lemma 3. We can find in $O(n^2)$ time a stable state in which the following is true: For each immortal node v either (a) there is a permanent node u such that there is a path from u to v, or (b) there is a pair $\{u, \overline{u}\}$ such that there are paths from u to $\overline{u}$ and v, and there are paths from $\overline{u}$ to u and v. □

Given a path (possibly with node repetitions) starting at a marked node, there is a sequence of state transitions that makes all nodes on the path marked one after the other, if they are not already marked. We call this transition sequence the *firing* of the path (it may perform fewer transitions than the length of the path, or even no transitions at all).

Continuing the proof of Theorem 2, we can obtain an undominated model thus: We first find a stable state according to Lemma 3. At this state, all live nodes have a path starting from a permanent node or a pair (in the later case, corresponding to the (b) clause of Lemma 3, we choose a path that goes at least twice between each of x and $\overline{x}$ before getting to v). Starting from this stable state, we successively fire all these paths (this takes time $O(n^2)$). It can then be shown that the resulting state is indeed undominated. □

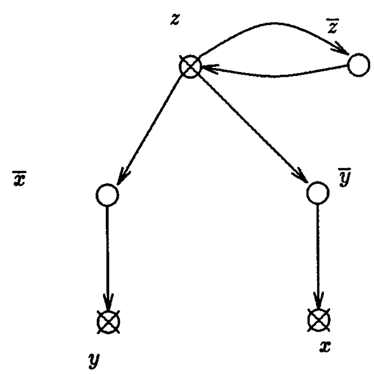


Figure 2: A state that is stable but dominated.

Notice that all undominated states are stable, but not all stable states are undominated. For example, the state in Figure 2 is stable because all nodes are immortal, but if $(z, \overline{x})$ fires there is no way to return to the same state. The reason is that there is a set B of marked nodes (in this case $B = \{x, y\}$) such that (a) all paths from marked nodes to them have as next to last node a node in $\overline{B} = \{\overline{x} : x \in B\}$; (b) at least one such path exists; and (c) there is no arc $(\overline{b}, b)$ for $b \in B$. We call such a set S of marked nodes *blocked* (it can be checked in $O(n^3)$ time whether there is a blocked set in a state). A state with a blocked set is dominated, as there can be no last state transition that restores the firing of one of the paths. We conjecture that *a state is undominated if and only if it is stable and has no blocked set.* By Lemma 2 and the above observation, this would give an $O(n^3)$ algorithm for determining whether a model is undominated in the case of 2–literal defaults; at present the existence of such an algorithm is open. The same construction as in the proof of Theorem 1 establishes that the problem of recognizing whether a model is undominated is PSPACE–complete for general defaults.

Finally, in order to prove that the undominated model problem for Horn clauses and 2–defaults is NP–hard, and similarly for the case of Horn defaults and 2–clauses (these are the two last entries of Table 1 we must discuss), we need just observe that *the satisfiability problem for formulae that contain a mixture of Horn clauses and 2–clauses is NP–complete* (proof: using 2–clauses we can rename the variables, thereby making all original clauses Horn), and then use Lemma 1.

3. MINIMAL MODELS

We next define the propositional analog of *circumscription*, an influential approach to common–sense reasoning proposed by McCarthy [Mc]. We are given a formula Γ in conjunctive normal form, and a subset X of the set of variables. We say that a model M of Γ is *X–minimal* if there is no other model M' of Γ having a set of true variables among those in X that are a proper subset of those of M. We are asking for an X–minimal model M of Γ.

Theorem 3. Finding an X–minimal satisfying truth assignment for a formula is Δ_2^p–complete.

Sketch: In Δ_2^p we can find the lexicographically smallest satisfying truth assignment, which is X–minimal. To prove completeness, we reduce the problem *deterministic satisfiability (DSAT)* shown Δ_2^p–complete in [Pa], to the problem of finding an X–minimal model. □

It is open whether this result holds even when X contains all variables. The same argument that proves that the general problem is in Δ_2^p, also establishes that the problem in the special case of 2–clauses is in P. However, the following is more interesting (shown by a careful implementation of the strong components algorithm for 2–satisfiability):

Theorem 4. Finding an X–minimal model for a conjunction of 2–clauses is in NC. □

4. THE RANDOMIZED ALGORITHM

If undominated models are indeed the instruments of vivid knowledge in human reasoning, how are we proposing that they are obtained? Certainly not by the complicated algorithm in the proof of Theorem 2! For an algorithm to qualify as a plausible model of human reasoning, it must satisfy much subtler criteria than correctness and efficiency. It should be natural (whatever this means) and "convincingly simple." The greedy algorithm and the monotonic algorithm for satisfying Horn clauses are good examples; so is the first-fit decreasing heuristic for bin-packing, and the exploration heuristics in [PY].

There is a randomized algorithm[3] for the undominated model problem that is suggested by the preference graph $G(\Gamma, \Delta)$: Consider this graph as a Markov chain (with equal transition probabilities on the arcs leaving each state, and adding self-loops to the sinks). In this Markov chain, the states with non-zero asymptotic probability are precisely the undominated models sought. Therefore, if we run this Markov chain long enough, we shall obtain almost surely an undominated model. The problem is, of course, that the number of transitions required may be as large as the number of states, which is exponential in the size of Γ and Δ. In fact, we can show the following disappointing result:

Proposition 1. There is a family of sets of 2-defaults such that, even without clauses, from a given starting state the expected time to reach an undominated model is exponential in the number of variables. □

The counterexample in this result must be heavily asymmetric, and thus non-clausal. Let us consider, however, a clausal set of defaults. The Markov chain, applied to it, can be thought of as the following randomized algorithm applied to the original set of clauses:

Start with any truth assignment. While there are unsatisfied clauses, pick one and flip a random literal in it.

Theorem 5. The above algorithm applied to 2-satisfiability and run for $O(n^2)$ steps, where n is the number of variables, will find a satisfying truth assignment, if one exists, with probability arbitrarily close to one.

[3] Many thanks to Charles Elkan for directing my attention to this issue.

Sketch:[4] If the formula is unsatisfiable, then nothing can go wrong. If it is satisfiable, consider a satisfying truth assignment, and the Hamming distance of our current assignment from it. It turns out that this distance is at least as likely to be decreased as to be increased, at all steps. And, naturally, this distance cannot be increased beyond n, the number of varables. Thus, this is a "gambler's ruin" chain with reflecting barrier (that is, the house cannot lose its last dollar). It follows that the gambler must be ruined after $O(n^2)$ steps, with probability arbitrarily close to one. □

5. OPEN PROBLEMS

Can undominated models of 2-literal defaults be recognized in polynomial time? (Also, recall the related conjecture at the end of Section 2.) Can the NP-hardness entries of Table 1 be tightened to PSPACE? We expect that the answer is "yes" in both questions.

Perhaps the most intriguing open problems suggested by this work are related to the probabilistic approach in the previous section. There are other polynomially solved special cases of satisfiability, besides 2SAT and Horn clauses, that this algorithm appears to solve (for example, the one in which we have 3-clauses with each variable appearing at most three times). Thus, the randomized algorithm seems to be a *unifying algorithmic approach* to satisfiability! Are there other special cases of satisfiability that can be handled by the randomized algorithm? One would hope that the emerging theory of rapidly mixing Markov chains (see, for example, [Mi, LS]) could in principle be employed to extend the applicability of this technique. However, the presence of sinks in our Markov chains (an important ingredient of our application) seems to prohibit direct application of those techniques.

Finally, a broader research direction is pointed to by this work: To advance the use of the methodology and norms of the theory of algorithms and complexity in the exploration of common-sense reasoning and intelligence (as has been done for learning, for example). This is particularly promising, as new influential theories of reasoning (such as the puzzle/routine dichotomy of [Le] that motivated this paper) have notions of algorithmic efficiency as their very basis.

[4] Tomas Feder recently pointed out to me that this result can also be proved by arguing along the same lines as in [Fe].

REFERENCES

[Fe] T. Feder "A New Fixed Point Approach to Stable Networks and Stable Marriages," *Proceedings 1989 STOC,* pp. 513–522; full version to appear in the special issue of *J.CSS.*

[KP] P. Kolaitis, C. H. Papadimitriou "Some Computational Aspects of Circumscription," *J.ACM 37,* 1, 1990.

[Le] H. Levesque "Making Believers out of Computers," *Artificial Intelligence, 30,* pp. 81–108, 1986.

[LS] L. Lovaśz, M. Simonovits "The Mixing Rate of Markov Chains, an Isoperimetric Inequality, and Computing the Volume," *Proc. 1990 FOCS,* pp. 346–354.

[Mc] John McCarthy "Circumscription —a Form of Nonmonotonic Reasoning," *AI, 13,* 27–39, 1980.

[Mi] M. Mihail "The Conductance and Convergence of Markov Chains —a Combinatorial Treatment of Expanders," *Proc. 1989 FOCS,* pp. 526-531.

[Pa] C. H. Papadimitriou "The complexity of Unique Solutions," *Proc. 1982 FOCS*; also, *JACM, 31,* 2, pp. 392–400, 1984.

[PY] C. H. Papadimitriou and M. Yannakakis "Shortest Paths without a Map," *Proc. 1989 ICALP*; to appear in the special issue of *Theor. Computer Science.*

[Se] F. Sebastiani "On Heterogeneous Model-preference Default Theories," *Proc. of the Eighth Biennial Conf. of the Canadian Society for Computational Studies of Intelligence* (CSCSI-90), Ottawa, pp. 84–91, 1990.

[SK] B. Selman and H. A. Kautz, "Model-preference Default Theories" *Art. Intelligence,* 45, 1990, pp. 287–322.

defaults:	none	atomic	2–literal	Horn	general defaults
no clauses	P^a	P^a	P^c	P^b	PSPACE–complete[c]
atomic clauses	P^a	P^a	P^c	P^b	PSPACE–complete[c]
2–clauses	P^a	P^a	P^c	NP–hard[c]	PSPACE–complete[c]
Horn clauses	P^a	P^a	NP–hard[c]	NP–hard[b]	PSPACE–complete[c]
general clauses	NP–complete[a]	NP–complete[a]	NP–hard[a]	NP–hard[a]	PSPACE–complete

Table 1: [a] well–known or easy; [b] [SK]; [c] this paper.