Adding Clauses to Poor Man's Logic (without Increasing the Complexity)

Peter Jonsson
Department of Computer and Information Science
Linköping University
SE-581 83, Linköping, Sweden
email: Peter.Jonsson@ida.liu.se

Abstract

Partly motivated by description logics, *poor man's logics* have been proposed as an interesting fragment of modal logics. A poor man's logic is a propositional modal logic where only literals and the connectives $\wedge$, $\Box$, and $\Diamond$ are allowed. It is known that the complexity of the satisfiability problem may drop dramatically when going from a full modal logic to the corresponding poor man's logic, e.g., in the case of modal logic K one goes from PSPACE-complete to coNP-complete. We prove that it is sometimes possible to extend poor man's logics with restricted disjunctions (i.e., clauses) without increasing the computational complexity. For Horn and Krom clauses, we provide necessary and sufficient conditions for when the resulting logic is polynomial-time. Such extensions correspond to allowing a restricted use of the union operator in description logics.

Keywords: modal logic, poor man's logic, computational complexity, disjunction.

1 Introduction

The computational complexity of propositional modal logics has been given a fair amount of attention in the literature. Since all 'reasonable' propositional modal logics contain ordinary propositional logic, the satisfiability problem is always NP-hard. Ladner [14] shows in his pioneering paper that such logics may have much higher complexity (K, T, and S4 are PSPACE-complete) but that they can also be NP-complete (i.e., K45, KD45, and S5). Several extended and restricted versions have also been studied: multi-modal[1] variants have been considered by, for instance, Halpern & Moses [9]. One of

[1]Some authors use the phrase 'multi-agent' instead of 'multi-modal'.

their findings is that multi-modal S5 is PSPACE-complete, i.e., harder than single-modal S5. Halpern [8] studies the effect of bounding the number of propositions and the depth of modal operator nesting; among many other things, he shows that multi-modal S5 is still PSPACE-complete even if there is only one proposition in the language. However, if the nesting depth is bounded by some constant, then the complexity drops to NP-complete.

Another way of restricting the language of modal logics is to disallow certain connectives. In *poor man's logics* [10], the disjunction operator $\vee$ is not allowed. The interest in this kind of logics stems, at least in part, from description logics (a.k.a. concept languages). Using results by Schild [16], it is possible to view certain description logics as syntactically restricted modal logics. There is a correspondence between these two as follows:

description logic	**modal logic**
universal concept	truth-value *true*
empty concept	truth-value *false*
atomic concept	propositional variable
complementation	negation $\neg$
intersection	conjunction $\wedge$
union	conjunction $\vee$
existential role quantification	modal operator $\Diamond$
universal role quantification	modal operator $\Box$

By using this correspondence, it is, for instance, possible to prove that the description logic $\mathcal{ALE}$ [6] is equivalent to the poor man's version of multi-modal K. The first complexity-theoretic characterisation of a poor man's logic was obtained by Schmidt-Schauss & Smolka [17] and Donini et al. [6]. By combining their results, it follows that the satisfiability problem of poor man's K is coNP-complete (while the unrestricted logic is PSPACE-complete). Hemaspaandra [10] presents a whole range of complexity results for different modal logics under the poor man's restriction. She shows, for instance, that there exists modal logics where the full and poor man's logic are both PSPACE-complete, and that there exists logics where the full logic is PSPACE-complete and the corresponding poor man's logic is in PTIME.

The main issue that we address is the following: given a poor man's logic with polynomial-time solvable satisfiability problem, how can it be extended with restricted disjunctions and still having a polynomial-time satisfiability problem? There are several reasons for considering this problem and the obvious reason is that enhanced expressibility is always desirable. This can be exemplified by description logics - the addition of restricted disjunctions corresponds to adding a restricted use of concept union. Another reason is that logics allowing restricted clauses (i.e., a restricted use of disjunctions) have historically proven to be extremely useful: logic programming (where Horn clauses are used) is probably the most striking example. Yet another reason making this research worthwhile is that the techniques developed are by no means restricted to modal logics, in fact, we develop a very abstract theory of disjunctions that may be useful in many different logics and elsewhere.

This work is inspired by recent progress in constraint satisfaction problems (CSP). In the basic CSP problem, we are given a set of variables that take their values from a finite domain and a set of constraints (e.g., relations) that constrain the values different variables can simultaneously take — the question is whether the variables can be assigned values that are consistent with all constraints. In a row of papers, (cf. [12, 5, 13, 3, 2]) the effects of adding disjunctions to CSP problems have been studied. Unfortunately, these techniques are not directly applicable to modal logics. In a CSP problem, there is a very simple correspondence between a variable and its associated value but this is not the case in modal logics; a model is a more intricate object than a valuation function. This difference is manifested in different ways but one interesting effect is that a model in a CSP problem is always of polynomial size while a model in a modal logic may be of exponential size (which is demonstrated by, for instance, Ladner [14]). In the first part of this paper, we give a general theory of disjunctions that is powerful enough to be applicable to modal (and other) logics. For technical reasons, we are not able to give both sufficient and necessary conditions of tractability (as is the case for CSPs) without adding a artificial restriction (which we call *normality*). However, we prove that disjunctive theories based on modal logics typically satisfy the normality condition and this enable us to prove fairly strong results.

To describe the results in some detail, let $\mathcal{L}$ denote the fomulae in a poor man's logic. Then, a *Krom clause* is a disjunction $(\alpha \vee \beta)$ where $\alpha, \beta \in \mathcal{L}$. Similarly, a *Horn clause* with respect to a set of formulae $X \subseteq \mathcal{L}$ is a disjunction $(\alpha_1 \vee \ldots \vee \alpha_k)$ where at most one α_i is not a member of X. Now, given a tractable poor man's logic, we begin by providing necessary and sufficient conditions for when it can be extended with Krom clauses while still having a tractable satisfiability problem. In the case of Horn clauses wrt. X, we have to make an additional assumption about X – if a formula ϕ is in X, then there must be an infinite number of formulae ϕ' in X such that ϕ' equals ϕ with renamed variables. Under this assumption, we can prove necessary and sufficient conditions for when adding Horn clauses wrt. X does not increase the complexity of the satisfiabililty problem. These results are illustrated by examples based on the modal logic KD, i.e., the standard system of deontic logic.

The paper is organised as follows: basic definitions can be found in Section 2 and the general framework for disjunctions and clauses in Section 3. The results are applied to poor man's logics in Section 4 and Section 5 contains some concluding remarks and open questions.

2 Preliminaries

We begin by describing the syntax and semantics of propositional logic. We assume the existence of a countably infinite set V of variable symbols. The set $\mathcal{L}$ of well-formed formulae is the smallest set satisfying the following inductive definition:

- p and $\neg p$ are formulae in $\mathcal{L}$ whenever $p \in V$; we identify $\neg\neg p$ with p.
- if $\phi \in \mathcal{L}$ and $\psi \in \mathcal{L}$, then so are $\phi \wedge \psi$ and $\phi \vee \psi$.

A *literal* is a formula on the form p or $\neg p$ where p is a variable. A *clause* is a disjunction of literals. A formula $\phi = \phi_1 \wedge \ldots \wedge \phi_n$ where each ϕ_i is a clause is said to be on *conjunctive normal form.*

The semantics is defined in the following way: an assignment M is a total function from propositional symbols to the truth values $\{0, 1\}$. The truth of a formula ϕ wrt. an assignment M, written as $M \models \phi$, is defined inductively:

$$
\begin{array}{lll}
M \models p \text{ for a variable } p & \text{iff} & M(p) = 1 \\
M \models \neg\phi' & \text{iff} & M \not\models \phi' \\
M \models \phi' \wedge \phi'' & \text{iff} & M \models \phi' \text{ and } M \models \phi'' \\
M \models \phi' \vee \phi'' & \text{iff} & M \models \phi' \text{ or } M \models \phi''
\end{array}
$$

We say that a formula ϕ is *satisfiable* iff there exists an assignment M such that $M \models \phi$. Such an assignment is called a *model.*

The satisfiability problem for propositional logic is NP-complete even when restricted to formulae in conjunctive normal form [7]. A number of tractable cases of this restricted problem have been identified, cf. [15]. In the *Krom* fragment, every clause is required to contain at most two literals. In the *Horn* fragment, each clause can contain at most one unnegated literal, and clauses in the *anti-Horn* fragment contain at most one negated literal.

We continue by briefly reviewing the syntax and semantics of modal logics. A more detailed introduction can be found in, for instance, Chellas [4]. The set $\mathcal{L}$ of well-formed formulae is the smallest set satisfying the following inductive definition:

- p and $\neg p$ are formulae in $\mathcal{L}$ whenever $p \in V$; we identify $\neg\neg p$ with p.
- if $\phi \in \mathcal{L}$ and $\psi \in \mathcal{L}$, then so are $\phi \wedge \psi$ and $\phi \vee \psi$.
- if $\phi \in \mathcal{L}$, then $\Box\phi \in \mathcal{L}$ and $\Diamond\phi \in \mathcal{L}$.

In other words, propositional modal logic has the same syntax as standard propositional logic except for two additional unary operators $\Box$ and $\Diamond$.

The most common way of providing a semantics for modal formulae is the *Kripke semantics*, where models $M = \langle W, R, \pi \rangle$ consists of a set W

(whose elements are called *worlds*), R (the *accessibility relation*) is a subset of $W \times W$ and π is a total function from worlds and propositional symbols to the truth values $\{0,1\}$.

The truth of a modal formula ϕ in a world w of a model M, written as $M, w \models \phi$, is defined inductively:

$$
\begin{array}{lll}
M,w \models p \text{ for a variable } p & \text{iff} & \pi(w,p) = 1 \\
M,w \models \neg\phi' & \text{iff} & M,w \not\models \phi' \\
M,w \models \phi' \wedge \phi'' & \text{iff} & M,w \models \phi' \text{ and } M,w \models \phi'' \\
M,w \models \phi' \vee \phi'' & \text{iff} & M,w \models \phi' \text{ or } M,w \models \phi'' \\
M,w \models \Diamond\phi' & \text{iff} & \exists u \in W \text{ such that } wRu \text{ and } M,u \models \phi' \\
M,w \models \Box\phi' & \text{iff} & \forall u \in W \text{ such that } wRu,\ M,u \models \phi'
\end{array}
$$

We say that a modal formula ϕ is *satisfiable* iff there exists a model $\langle W, R, \pi\rangle$ and a world $w \in W$ such that $M, w \models \phi$. We see that the meaning of a modal formula is closely related to the choice of accessibility relation. For instance, if R is reflexive, symmetric, and transitive, i.e., R is an equvialence relation, then we are dealing with the modal logic S5.

Poor man's logics are restricted variants of propositional modal logics where the formulae are not allowed to contain $\vee$. In this restricted language, we will view $\wedge$ as a multi-arity operator, and we will assume that all conjunctions are 'flattened', that is, a conjunct will not be a conjunction. Consequently, a formula such as $(\Box p \wedge \Box q) \wedge \Box r$ is considered to contain three conjuncts and not two. This allows us to view every formula ϕ in this language as being of the form

$$\phi = \Box\psi_1 \wedge \ldots \wedge \Box\psi_k \wedge \Diamond\xi_1 \wedge \ldots \wedge \Diamond\xi_m \wedge \ell_1 \wedge \ldots \wedge \ell_s$$

where $\ell_1, \ldots, \ell_s$ are literals. An example of such a formula is

$$\Box(\Diamond\Box p_2 \wedge \Diamond p_1 \wedge \Diamond\Box\Box\neg p_1) \wedge \Diamond\neg p_2 \wedge \neg p_3.$$

We assume throughout this article that the complexity classes PTIME, NP and PSPACE are distinct. For the sake of brevity, we say that problems in PTIME are *tractable*. We will use the two NP-complete problems 3-SAT and 3-COLOURABILITY [7] in our NP-completeness results. They are defined as follows:

3-SAT
INSTANCE: A propositional formula in conjunctive normal form $F = c_1 \wedge \ldots \wedge c_k$ where each clause c_i contains at most 3 literals.
QUESTION: Is there a model for F?

3-COLOURABILITY
INSTANCE: An undirected graph $G = (V, E)$.
QUESTION: Is there a function $f : V \to \{1, 2, 3\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$.

3 An Abstract View on Disjunctions and Clauses

In this section, we develop an abstract therory of disjunctions and clauses – a theory that is more general than the corresponding work on disjunctive CSPs. As a running example, we will use ordinary propositional logic and we present our framework in a way that emphasises the connections between the abstract formulation of clauses and clauses in propositional logic.

Let P be a set and let $\mathcal{S}$ be a total function from finite subsets of P to $\{0,1\}$. As an example, P may be a set of formulae in some logic and $\mathcal{S}$ a function that decides whether a finite subset of $\mathcal{S}$ is satisfiable or not. It is important to realise that $\mathcal{S}$ is *not* a function assigning truth-values to variables in the logic or something similar. If we want to be able to identify *polynomial-time* fragments of logics, then we must assume that $\mathcal{S}$ is polynomial-time computable – thus, $\mathcal{S}$ is tacitly assumed to have this property throughout the article. For a concrete example, let P_{prop} denote the set of literals over the propositions $\{x_1, x_2, \dots\}$. Obviously, the satisfiability function $\mathcal{S}_{\mathrm{prop}} : 2^{P_{\mathrm{prop}}} \to \{0,1\}$ is polynomial-time computable in this case.

We now turn our attention to *clauses* and *formulae*. In propositional logic, a clause is a disjunction of the form $(\ell_1 \vee \dots \vee \ell_k)$ where $\ell_i \in P_{\mathrm{prop}}$. Thus, we simply define clauses over an arbitrary set P as follows: a *clause over* P is a finite set $C \subseteq P$. A propositional formula on *conjunctive normal form* (CNF) is a conjunction of clauses and such a formula is satisfiable if and only if there exists an assignment M that satisfies every clause. These notions can also easily be extended to an arbitrary set P: a *formula* is a finite set of clauses and a formula $F = \{F_1, \dots, F_k\}$ is *satisfiable* iff

$$\exists f_1 \in F_1, \dots, f_k \in F_k \text{ such that } \mathcal{S}(\{f_1, \dots, f_k\}) = 1.$$

That is, we pick one member out of every clause and check that they are simultaneously satisfiable – this implies that the formula itself is satisfiable. We say that $\{f_1, \dots, f_k\}$ is a *support set* for F. We use the term 'support set' instead of 'model' in order to avoid some confusion that may arise otherwise. We note that the problem of deciding whether a formula has a support set or not is in NP since $\mathcal{S}$ is polynomial-time computable.

We will now reconsider propositional Krom and (anti)-Horn clauses that were defined in Section 2: a Krom clause contains at most two literals, a Horn clause contains at most one unnegated literal, and an *anti-Horn* clauses contains at most one negated literal. Defining Horn and Krom clauses abstractly is fairly straightforward:

Definition 3.1 *Let P be a set and let $\mathcal{S} : 2^P \to \{0,1\}$ be a total function. A* K-structure *is the pair $(P, \mathcal{S})$ and an* H-structure *is a triple $(P, H, \mathcal{S})$ where $\varnothing \neq H \subseteq P$. A* Krom clause over $(P, \mathcal{S})$ *is a set $C \subseteq P$ such that $|C| \leq 2$. A* Horn clause over $(P, H, \mathcal{S})$ *is a finite set $C \subseteq P$ containing at most one element which is not in H.*

K- and H-structures are used for compactly describing the syntax and semantics of Krom and Horn clauses – clearly, by specifying such structures, we have exactly described the syntax and semantics of the corresponding clauses. In a H-structure $(P, H, \mathcal{S})$, we see that the members of H corresponds to the negated literals in propositional Horn clauses; a clause may contain at most one element *not* in H. We define Krom and Horn formulae in the obvious way, e.g. a Krom formula is a set consisting of only Krom clauses, and consequently their syntax and semantics is also completely described by the underlying structure.

We continue to define two important properties of structures – it will be proved later on that the satisfiability problem for Krom or Horn formulae is tractable when these conditions hold.

Definition 3.2 *Let $\Sigma = (P, \mathcal{S})$ be a K-structure. We say that Σ is* Krom *iff for every finite subset $P' \subseteq P$,*

$$\mathcal{S}(P') = 0 \text{ iff } \exists p, q \in P' : \mathcal{S}(\{p, q\}) = 0.$$

We see that the K-structure $(P_{\text{prop}}, \mathcal{S}_{\text{prop}})$ is Krom: a set of literals L is not satisfiable if and only if there exists a variable p such that $p \in L$ and $\neg p \in L$.

Definition 3.3 *Let $\Sigma = (P, H, \mathcal{S})$ be an H-structure. We say that Σ is* Horn *iff for every finite subset $H' = \{h_1, \ldots, h_n\} \subseteq H$ and every finite subset $P' \subseteq P$,*

$$\text{if } \mathcal{S}(P' \cup \{h_1\}) = 1 \text{ and } \ldots \text{ and } \mathcal{S}(P' \cup \{h_n\}) = 1, \text{ then } \mathcal{S}(P' \cup H') = 1.$$

It is straightforward to show that the H-structure $(P_{\text{prop}}, H, \mathcal{S}_{\text{prop}})$ is Horn if we choose $H = \{\neg p \mid p \text{ is a variable}\}$. In this case, the resulting logic with clauses is the Horn fragment of propositional logic. If we instead choose $H = \{p \mid p \text{ is a variable}\}$, we end up with the anti-Horn fragment.

We conclude with two technical definitions which will be important when it comes to proving necessary conditions for tractability of disjunctive formulae. Basically, these *normality* conditions state that if there is a set violating the Krom or Horn property, then there are infinitely many such sets.

Definition 3.4 *A K-structure $\Sigma = (P, \mathcal{S})$ is* normal *if and only if Σ is either Krom or satisfies the following property: there exists an infinite set $\{P'_1, P'_2, \ldots\}$ such that for all i, j*

1. *P'_i is a finite subset of P;*
2. *$P'_i \cap P'_j = \varnothing$; and*
3. *$\mathcal{S}(P'_i) = 0$ but $\mathcal{S}(X) = 1$ for every two-element subset X of P'_i.*

If Σ is not Krom, then each P'_i contains an example of elements that violates the Krom property. Since $P'_i \cap P'_j = \varnothing$ for all i, j, there is an infinite number of such examples.

Definition 3.5 *A H-structure* $\Sigma = (P, H, \mathcal{S})$ *is* normal *if and only if* Σ *is Horn or satisfies the following property: there is an infinite set* $\{(P'_1, H'_1), (P'_2, H'_2), \dots\}$ *such that for all* i, j*:*

1. P'_i *and* H'_i *are finite subsets of* P *and* H*, respectively;*
2. $(P'_i \cup H'_i) \cap (P'_j \cup H'_j) = \varnothing$*; and*
3. $\mathcal{S}(P'_i \cup \{h_1\}) = 1, \dots, \mathcal{S}(P'_i \cup \{h_k\}) = 1$ *(where* $H'_i = \{h_1, \dots, h_k\}$*) but* $\mathcal{S}(P'_i \cup H'_i) = 0$.

3.1 Krom formulae

We will now provide necessary and sufficient conditions for the tractability of deciding satisfiability of Krom formulae of normal K-structures; the main result can be found in Theorem 3.7. This theorem builds on a polynomial-time algorithm KromSAT for deciding satisfiability of Krom formulae. The algorithm is presented in Figure 1 and its correctness is proved in Theorem 3.6. KromSAT transforms the input formula into a set of propositional Krom clauses that are satisfiable if and only if the input is satisfiable. Since the satisfiability of Krom clauses can be decided in polynomial time [1], tractability of the algorithm follows. Note that the normality condition is not required for proving the correctness of KromSAT.

Theorem 3.6 *Let* $\Sigma_K = (P, \mathcal{S})$ *be an arbitrary K-structure that is Krom. Deciding whether a Krom formula over* Σ_K *is satisfiable or not is a polynomial time task.*

Proof: We claim that algorithm KromSAT (Figure 1) can solve this problem in polynomial time. We first show that KromSAT is a polynomial time algorithm. Let n be the length of the input formula. Then, $|R| \leq n$ so A can be computed in $O(n^2)$ time while B and U can be computed in $O(n)$ time. The set $A \cup B \cup U$ contains at most $O(n^2)$ clauses so the test in line 6 can be performed in polynomial time since the satisfiability of a set of propositional Krom clauses can be determined in polynomial time [1]. We conclude that KromSAT runs in polynomial time.

Next we prove the KromSAT(F) accepts if and only if F is satisfiable.

only-if: We assume that KromSAT(F) accepts and construct a support set N for F. Let M be a satisfying assignment of $A \cup B \cup U$, $R = \bigcup F$, and

$$N = \{p \in R \mid M(q_p) = 1\}.$$

algorithm KromSAT(F)
$R = \bigcup F$
$A \leftarrow \{(\neg q_p \vee \neg q_{p'}) \mid p, p' \in R$ and $\mathcal{S}(\{p, p'\}) = 0\}$
$B \leftarrow \{(q_p \vee q_{p'}) \mid \{p, p'\} \in F\}$
$U \leftarrow \{(q_p) \mid \{p\} \in F\}$
if $A \cup B \cup U$ is satisfiable **then return** 'yes'
else return 'no'

Figure 1: Algorithm for deciding satisfiability of Krom theories.

We first show that $\mathcal{S}(N) = 1$. Since F is a Krom formula, $\mathcal{S}(N) = 0$ iff

$$\exists p, q \in N : \mathcal{S}(\{p, q\}) = 0.$$

However, this is prevented since M satisfies the formulae in the set A.

Arbitrarily choose a clause $\{p, p'\} \in F$. By the clauses in B, at least one of p, p' will be a member of N. Similarly, for each clause of the type $\{p\} \in F$, p is a member of N by the clauses in U.

if: Assume that F is satisfiable and let N be a support set for F. We show that there exists a satisfying truth assignment M of $A \cup B \cup U$. Define M as follows:

$$M(q_p) = 1 \text{ iff } p \in N.$$

We show that M is a satisfying truth assignment by considering three cases:

1. Arbitrarily choose a clause $(\neg q_p \vee \neg q_{p'}) \in A$. If $M(q_p) = M(q_{p'}) = 1$, then $p, p' \in N$ and $\mathcal{S}(N) = 0$ since $\mathcal{S}(\{p, p'\}) = 0$. Thus, $(\neg q_p \vee \neg q_{p'})$ is satisfied by M.

2. Arbitrarily choose a clause $(q_p \vee q_{p'}) \in B$ and note that $\{p, p'\} \in F$. Since N is a support set for F, at least one of p, p' is in N so M assigns 1 to at least one of $q_p, q_{p'}$.

3. Arbitrarily choose a clause $(q_p) \in U$ and note that $\{p\} \in F$. Since $p \in N$, $M(q_p) = 1$.

□

Next, we prove that satisfiability of Krom formulae over Σ (which is assumed to be a normal K-structure) is solvable in polynomial time if and only if Σ is Krom, and the problem is NP-complete otherwise. The proof reveals that if Σ is not Krom, then the 3-COLOURABILITY problem can be reduced to the satisfiability problem for Krom formulae over Σ in polynomial time.

Theorem 3.7 *Let $\Sigma = (P, \mathcal{S})$ be a normal K-structure. Satisfiability of Krom formulae over Σ is solvable in polynomial time if and only if Σ is Krom. Otherwise, the problem is NP-complete.*

Proof: The if-direction is a consequence of Theorem 3.6. To prove the opposite direction, we assume to the contrary that Σ is not Krom. The normality of Σ implies that there exists an infinite set $\{P'_1, P'_2, \dots\}$ where each $P'_i \subseteq P$ is finite and the following holds:

1. for all i, j, $P'_i \cap P'_j = \varnothing$; and
2. for all i, $\mathcal{S}(P'_i) = 0$ but $\mathcal{S}(X) = 1$ for every two-element subset $X \subseteq P'_i$

For each P'_i, we assume that $|P'_i|$ is as small as possible (and we note that $|P'_i| \geq 3$). This can be done without loss of generality since if P'_i is not of minimal size, then we can remove at least one element from the set without violating properties 1) and 2). The minimality of P'_i implies that $\mathcal{S}(X) = 1$ for all $X \subset P'_i$. Choose a $P'_i = \{p_1, \dots, p_n\}$, consider the formula

$$F = \{\{p_1, p_2\}, \{p_1, p_3\}, \{p_2, p_3\}, p_4, \dots, p_n\}$$

and note that in any support set N for F, exactly one of p_1, p_2, p_3 will *not* be a member of N. We will now use the formula F in a polynomial-time reduction from 3-COLOURABILITY to the satisfiability problem of Krom formulae over Σ; this proves NP-completeness when combined with the fact that the problem is in NP.

Arbitrarily choose an undirected graph $G = (V, E)$ such that $V = \{v_1, \dots, v_k\}$. We will construct a Krom formula over Σ that has a support set iff G is colourable with three colours. For each vertex v_i, introduce a fresh copy of the formula F (i.e., for v_1 use P'_1 as the basis for F and so on) where we denote the 'important' elements p_1, p_2, p_3 as p^i_1, p^i_2, p^i_3, respectively. As we have already noted, this will force exactly one of p^i_1, p^i_2, p^i_3 not to be a member in any given support set. We interpret that p^i_j is not a member as 'vertex v_i has colour j'.

For each edge $(v_i, v_j) \in E$, we add the clauses $\{p^i_1, p^j_1\}$, $\{p^i_2, p^j_2\}$, and $\{p^i_3, p^j_3\}$ which ensures that v_i and v_j are not assigned the same colour. The resulting formula is a Krom formula over Σ, it can be computed in polynomial time and it is satisfiable iff G is 3-colourable. □

3.2 Horn formulae

Necessary and sufficient conditions for the tractability of deciding satisfiability of Horn formulae are presented in this subsection. We begin by providing a tractable algorithm HornSAT (see Figure 2) for deciding satisfiability of

Horn formulae over H-structures satisfying the Horn property. This algorithm recursively simplifies the input formula in such a way that satisfiability is preserved, and eventually satisfiability can be decided easily. The correctness of the algorithm is proved in Theorem 3.13 and the main result can be found in Theorem 3.14. The normality condition is not required for proving the correctness of HornSAT.

Throughout the correctness proofs of HornSAT, we let $\Sigma_H = (P, H, \mathcal{S})$ be a H-structure that is Horn but not necessarily normal. For the sake of brevity, we simply refer to a 'horn clause' when we actually mean a 'horn clause over Σ_H'.

Definition 3.8 *Let C be a Horn clause. Let* $\mathrm{Pos}(C) = C \cap (P - H)$ *and* $\mathrm{Neg}(C) = C \cap H$. *We say that C is* positive *iff* $|\mathrm{Neg}(C)| = 0$ *and that C is* negative *iff* $|\mathrm{Pos}(C)| = 0$. *If C is positive or negative, we say that C is* homogeneous *and otherwise* heterogeneous. *We extend these definitions to formulae in the obvious way. For example, if F is a Horn formula and every $C \in F$ is positive, then F is positive.*

Example 3.9 *We consider the H-structure* $\Sigma = (P_{\text{prop}}, H, \mathcal{S}_{\text{prop}})$ *where* $H = \{\neg p \mid p \text{ is a variable}\}$. *Let* $C = \{p, \neg q, \neg r\}$. *We see that* $\mathrm{Pos}(C) = \{p\}$, $\mathrm{Neg}(F) = \{\neg q, \neg r\}$ *and that C is heterogeneous. Similarly,* $C' = \{\neg q, \neg r\}$ *is a negative clause and, consequently, homogeneous.*

Note that if C is a positive Horn clause, then C is *unary*, that is, $|C| = 1$. The concept of *blocking* is defined as follows.

Definition 3.10 *Let A be a satisfiable set of Horn clauses and let C be a Horn clause. We say that C* blocks *A iff $A \cup \{c\}$ is not satisfiable for any* $c \in \mathrm{Neg}(C)$,

Example 3.11 *(continuation of Example 3.9) If* $A = \{\{\neg p, \neg q\}, \{q\}, \{r\}\}$, *then the clause C blocks A since $A \cup \{c\}$ is not satisfiable for any* $c \in \mathrm{Neg}(C) = \{\neg q, \neg r\}$. *However, C does not block the formula* $A' = \{\{\neg p, q\}, \{r\}\}$ *since $A' \cup \{\{\neg q\}\}$ is satisfiable.*

Observe that if $A \cup \{C\}$ is satisfiable and C blocks A, then there must exist an $c \in \mathrm{Pos}(C)$ such that $A \cup \{c\}$ is satisfiable. This observation will be of great importance later on. Another important property of blocking is that it can be decided in polynomial time, provided that A consists of unary Horn formulae only (this follows immediately from the fact that $\mathcal{S}$ is computable in polynomial-time). Now, before proving the correctness of HornSAT, we need to prove a number of minor results concering the satisfiability of Horn formulae.

Lemma 3.12 *Let F be a Horn formula and $A \subseteq F$ the set of positive clauses in F. Then, the following holds:*

1. *Let $D = \{D_1, \ldots, D_k\} \subseteq F$ be the set of formulae that are not positive. Under the condition that A is satisfiable, F is satisfiable if D_i does not block A for any $1 \leq i \leq k$.*

2. *If there exists a heterogeneous Horn clause $C \in F$ such that C blocks A, then F is satisfiable iff $(T - \{C\}) \cup \mathrm{Pos}(C)$ is satisfiable.*

3. *If there exists a negative clause $C \in F$ that blocks A, then F is not satisfiable.*

Proof:

1. Pick one member d_i out of every D_i such that $A \cup \{d_i\}$ is satisfiable. This is possible since no D_i blocks A. Now, Σ_H is Horn so $\mathcal{S}(A \cup \{d_1, \ldots, d_k\}) = 1$ and F is satisfiable.

2. The if-direction is trivial. To show the other direction, we see that if F is satisfiable, then C has to be satisfiable. Since C blocks A, $\mathrm{Pos}(C)$ must be a member of every support set for F.

3. If F is satisfiable, then $A \cup \{C\}$ must be satisfiable. Since C is negative and blocks A, this is not the case.

$\square$

We have now all the machinery needed for proving that the satisfiability problem for Horn formulae is tractable.

Theorem 3.13 *Deciding whether a Horn formula is satisfiable or not is a polynomial time task.*

Proof: We begin by showing that algorithm HornSAT (Figure 2) runs in polynomial time. We have already noted that the blocking property can be decided in polynomial-time so the tests in lines 3–5 can be performed in polynomial time. Furthermore, the number of recursive calls is bounded by the number of heterogeneous clauses in the given input which implies that the running time of HornSAT is bounded from above by some polynomial.

Next, we prove that HornSAT(F) accepts if and only if F is satisfiable.

only-if: Induction over n, the number of heterogeneous clauses in F.

Basis step: If $n = 0$ and HornSAT(F) accepts, then the formulae in A are satisfiable and there does not exist any $C \in F$ that blocks A. Consequently, F is satisfiable by Lemma 3.12(1).

Induction hypothesis: Assume the claim holds for $n = k$, $k \geq 0$.

Induction step: F contains $k + 1$ heterogeneous formulae. If HornSAT accepts in line 5, then $(F - \{C\}) \cup Pos(C)$, which contains k heterogeneous clauses, is satisfiable by the induction hypothesis. By Lemma 3.12(2), this is equivalent to F being satisfiable. If HornSAT accepts in line 6, then there

1 **algorithm** HornSAT(F)
2 $A \leftarrow \{C \mid C \in F \text{ is positive}\}$
3 **if** $\mathcal{S}(A) = 0$ **then return** 'no'
4 **if** $\exists C \in F$ that blocks A and is negative **then return** 'no'
5 **if** $\exists C \in F$ that blocks A and is heterogeneous
  then return HornSAT($(F - \{C\}) \cup \mathrm{Pos}(C)$)
6 **else return** 'yes'

Figure 2: Algorithm for deciding satisfiability of Horn theories.

does not exist any negative or heterogeneous $C \in F$ which blocks A. By Lemma 3.12(1), this implies that F is satisfiable.

if: We show the contrapositive by induction over n, the number of heterogeneous formulae in F.

Basis step: If $n = 0$ then, HornSAT can reject in lines 3 and 4. If HornSAT rejects in line 3, then, trivially, F is not satisfiable. If HornSAT rejects in line 4, then there exists a negative $C \in F$ that blocks A. Hence, F is not satisfiable by Lemma 3.12(3).

Induction hypothesis: Assume the claim holds for $n = k$, $k \geq 0$.

Induction step: F contains $k+1$ heterogeneous formulae. If HornSAT rejects in line 3, then F is not satisfiable. If HornSAT rejects in line 4, then F is not satisfiable by Lemma 3.12(3). If HornSAT rejects in line 5, then $(F - \{C\}) \cup \mathrm{Pos}(C)$, which contains k heterogeneous formulae, is not satisfiable by the induction hypothesis. By Lemma 3.12(2), this is equivalent to F not being satisfiable. □

We can now state and prove the main result for Horn formulae. The proof is once again a reduction but, unlike the proof of Theorem 3.7, we will start from the 3-SAT problem instead.

Theorem 3.14 *Let $\Sigma = (P, H, \mathcal{S})$ be a normal H-structure. Satisfiability of Horn theories over Σ is solvable in polynomial time if and only if Σ is Horn. Otherwise, the problem is NP-complete.*

Proof: The if-direction is a consequence of Theorem 3.13. To prove the opposite direction, we assume to the contrary that Σ is not a Horn structure. Normality implies that there exists an infinite set $\{(P'_1, H'_1), (P'_2, H'_2), \ldots\}$ where all $P'_i \subseteq P$, $H'_i \subseteq H$ are finite and the following holds:

1. for all i, j, $(P'_i \cup H'_i) \cap (P'_j \cup H'_j) = \varnothing$;
2. for all i, $\mathcal{S}(P'_i \cup \{h_1\}) = 1, \ldots, \mathcal{S}(P'_i \cup \{h_k\}) = 1$ (where $H'_i = \{h_1, \ldots, h_k\}$) but $\mathcal{S}(P'_i \cup H'_i) = 0$.

For each pair (P'_i, H'_i) we additionally assume that $|H'_i|$ is as small as possible and we note that $|H'_i| \geq 2$. The minimality of H'_i implies that $\mathcal{S}(P'_i \cup X) = 1$ for all $X \subset H'_i$. Assume $H_1 = \{h_1, \ldots, h_k\}$ and consider the formula

$$F = P'_1 \cup \{\{h_1, h_2\}, \{h_3\}, \ldots, \{h_k\}\}.$$

Observe that in any support set N for F, either $h_1 \in N$ or $h_2 \in N$ but not both. We will now use the formula F in a polynomial-time reduction from 3SAT to the satisfiability problem of the Horn theory over Σ.

Arbitrarily choose a 3-SAT instance $\phi = c_1 \wedge \ldots \wedge c_n$ over the variables $p_1, \ldots, p_m$. We incrementally construct a Horn formula over Σ that is satisfiable iff ϕ is satisfiable – this proves that the problem is NP-complete.

For each variable p_i, introduce a fresh copy of the formula F (i.e., for p_1 use (P'_1, H'_1) as the basis for F and so on) where we denote the 'important' elements h_1 and h_2 as h^i_t and h^i_f, respectively. As we noted earlier, this will force either h^i_t or h^i_f to be a member in every support set but not both. We interpret $h^i_t \in \mathcal{M}$ as 'p_i is true' and $h^i_f \in \mathcal{M}$ as 'p_i is false'.

For each clause c_i in ϕ, it is now easy to add a disjunction corresponding to the clause: for instance, $(p_i \vee \neg p_j \vee p_k)$ is translated to $\{h^i_t, h^j_f, h^k_t\}$. Obviously, the resulting formula (which obviously can be computed in polynomial time) is a Horn formula over Σ and it is satisfiable iff ϕ is satisfiable. $\square$

4 Adding Clauses to Poor Man's Logics

We shall now apply the results from the previous section to poor man's logics. We present the main result in Subsection 4.1 and apply the result to modal logic KD in Subsection 4.2.

4.1 Syntax and main result

Let V be a countably infinte set of variable symbols and let P denote the formulae of the poor man's logic (as defined in Section 2) based on V. Let $\Sigma_K = (P, \mathcal{S})$ be a K-structure such that $\mathcal{S}$ is the satisfiability function for some poor man's logic over P. Remember that $\mathcal{S}$ is polynomial-time computable, i.e., the satisfiability problem for the poor man's logic over P is assumed to be tractable. Also let $\Sigma_H = (P, H, \mathcal{S})$ be a H-structure such that $H \subseteq P$ and $\mathcal{S}$ the satisfiability function for some poor man's logic over P. We say that $H \subseteq P$ is *renamable* if for every $\phi \in H$ and every variable substitution θ relative to V, $\phi/\theta \in H$. For example, the set $\{\neg v \mid v \in V\}$ is renamable.

The set $\mathcal{L}_K$ of *Krom formulae* over Σ_K is the smallest set satisfying the following inductive definition:

- $P \subseteq \mathcal{L}_K$.
- if $\phi, \psi \in P$, then $(\phi \vee \psi) \in \mathcal{L}_K$
- if $\phi \in \mathcal{L}_K$ and $\psi \in \mathcal{L}_K$, then so is $\phi \wedge \psi$.

Similarly, the set $\mathcal{L}_H$ of *Horn formulae* over Σ_H is the smallest set satisfying the following inductive definition:

- $P \subseteq \mathcal{L}_H$.
- if $\phi_1, \ldots, \phi_k \in P$ and at most one $\phi_i \notin H$, then $(\phi_1 \vee \ldots \vee \phi_k) \in \mathcal{L}_H$
- if $\phi \in \mathcal{L}_H$ and $\psi \in \mathcal{L}_H$, then so is $\phi \wedge \psi$.

We refer to the logic whose language is restricted to $\mathcal{L}_K$ as the *Krom logic* over Σ_K, and the logic whose language is restricted to $\mathcal{L}_K$ as the *Horn logic* over Σ_H. Examples of formulae in $\mathcal{L}_K$ are $(\Box(p \wedge \Diamond q) \vee \Diamond(\Box p \wedge \Diamond q)) \wedge \Box r$ and $(p \vee \neg q) \wedge (\Box(p \wedge q) \vee \neg r)$.

Theorem 4.1 *The following holds:*

1. *the Krom logic over Σ_K is polynomial-time if and only if Σ_K is Krom;*
2. *if H is renamable, then the Horn logic over Σ_H is polynomial-time if and only if Σ_H is Horn.*

Proof:

1. We prove first that Σ_K is a normal K-structure. If Σ_K is Krom, then we are done. If Σ_K is not Krom, then there exists a finite set $P' \subseteq P$ that is not satisfiable but every two-element subset of P' is satisfiable. Assume that P' is built from the variable set $\{v_1, \ldots, v_k\} \subseteq V$. By renaming the variables, we can build an infinite sequence (since V is infinite) of pairwise disjoint sets $P'_1, P'_2, \ldots$ satisfying the same property as P'; in other words, Σ_K is normal. The result follows from Theorem 3.7.

2. We prove first that Σ_H is a normal H-structure. If Σ_H is Horn, then we are done. Otherwise, there exists a finite subset $H' = \{h_1, \ldots, h_n\} \subseteq H$ and finite subset $P' \subseteq P$ such that

$$\mathcal{S}(P' \cup \{h_1\}) = 1 \text{ and } \ldots \text{ and } \mathcal{S}(P' \cup \{h_n\}) = 1$$

but $\mathcal{S}(P' \cup H') = 0$. By renaming the variables (which is possible since H is renamable), we can build an infinite sequence $(P'_1, H'_1), (P'_2, H'_2), \ldots$ of pairs that shows that Σ_H is normal. The result follows from Theorem 3.14.

$\Box$

4.2 Modal logic KD

We will now illustrate the result presented in the previous subsection by applying it to a concrete modal logic. The modal logic KD (also known as the standard system for deontic logic) is based on frames in which every world has at least one successor, i.e., the accessibility relation R satisfies $\forall x \exists y : R(x, y)$. This logic has been one of the most influential modal logics, cf. [11]. The satisfiability problem for KD is PSPACE-complete and the poor man's version is tractable as shown by Hemaspaandra [10]. Note that this logic is denoted $\mathcal{F}_{\geq 1}$ in Hemaspaandra's paper. Hemaspaandra shows that a poor man's formula

$$\phi = \Box\psi_1 \wedge \ldots \wedge \Box\psi_k \wedge \Diamond\xi_1 \wedge \ldots \wedge \Diamond\xi_m \wedge \ell_1 \wedge \ldots \wedge \ell_s$$

(where ℓ_i is free from modal operators) is not satisfiable if and only if the following conditions hold: there exist i, j such that

1. $\ell_i \wedge \ell_j$ is not satisfiable;
2. $\Box\psi_i \wedge \Box\psi_j$ is not satisfiable;
3. $\Box\psi_i \wedge \Diamond\xi_j$ is not satisfiable; or
4. $\Diamond\xi_i \wedge \Diamond\xi_i$ is not satisfiable.

These conjunctions contains at most two conjuncts which implies that $(P, \mathcal{S})$ is Krom so, by Theorem 4.1, the Krom logic for KD is tractable. One tractable Horn logic for KD is based on the H-structure $\Sigma = (P, \{\neg v \mid v \in V\}, \mathcal{S})$. The set H is renamable and to demonstrate that Σ is Horn, we assume that there exists a finite set $A \subseteq P$ and a finite set $B \subseteq H$ such that $A \cup \{b\}$ is satisfiable for all $b \in B$ but $A \cup B$ is not satisfiable. By using the conditions above and noting that A is satisfiable, there must exist a literal $a \in A$ which is unnegated and a literal $b \in B$ such that $a \wedge b$ is not satisfiable. This implies that $b \equiv \neg a$ and $A \cup \{b\}$ is not satisfiable which leads to a contradiction. Consequently, Σ is Horn and the Horn logic for KD based on Σ is tractable by Theorem 4.1.

5 Conclusions

We have studied the computational complexity of poor man's logics extended with a limited use of disjunctions (in the form of clauses). Given a poor man's logic with tractable satisfiability problem, we have shown necessary and sufficient conditions for when the addition of Horn and Krom clauses does not increase the complexity of the satisfiability problem. The methods used for proving this is not restricted to poor man's logics but can be applied to many different formalisms, for instance, these results subsumes earlier results on extending constraint satisfaction problems with disjunctions.

The results in this article rise a number of questions. The main question is two-fold:

1. which of the poor man's logics are tractable; and

2. which of them can be extended with Krom/Horn clauses with preserved tractability?

This question is probably very difficult to answer even if we restrict our attention to 'well-understood' logics such as those studied in Ladner's [14] article. The following may be a slightly simpler question: which are the poor man's logics that (like KD) have a very simple satisfiability-checking procedure that only consider two conjuncts at a time? In this case, we know that the Krom property holds unconditionally and it seems likely that the logic itself must satisfy certain strong conditions. Yet another interesting problem is to consider other restrictions on the set of operators. In a poor man's logic, we disallow the use of the disjunction operator. However, one can disallow many different sets of operators, study the complexity of the resulting logic and how the complexity is altered when clauses are added. Hemaspaandra [10] provides a number of results along these lines.

Acknowledgements

This research was partially supported by the *Swedish Research Council* (VR) under grant 221-2000-361.

References

[1] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear time algorithm for testing the truth of certain quantified Boolean formulas. *Information Processing Letters*, 8:121–123, 1979.

[2] M. Broxvall. Constraint satisfaction on infinite domains: Composing domains and decomposing constraints. In *Proceedings of the 8th International Conference on Principles on Knowledge Representation and Reasoning (KR-2002)*, pages 509–520, 2002.

[3] M. Broxvall, P. Jonsson, and J. Renz. Disjunctions, independence, refinements. *Artificial Intelligence*, 140(1/2):153–173, 2001.

[4] B. F. Chellas. *Modal Logic*. Cambridge University Press, Cambridge, 1980.

[5] D. Cohen, P. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2001.

[6] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. *Information and Computation*, 134:1–58, 1997.

[7] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.

[8] J. Y. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(2):361–372, 1995.

[9] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.

[10] E. Hemaspaandra. The complexity of poor man's logic. *Journal of Logic and Computation*, 11(4):609–622, 2001.

[11] R. Hilpinen. *Deontic logic: introductory and systematic readings.* Kluwer Academic Publishers, 1970.

[12] P. Jonsson and C. Bäckström. A unifying approach to temporal constraint reasoning. *Artificial Intelligence*, 102(1):143–155, 1998.

[13] M. Koubarakis. Tractable disjunctions of linear constraints: Basic results and applications to temporal reasoning. *Theoretical Computer Science*, 266(1–2):311–339, 2001.

[14] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

[15] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[16] K. Schild. A correspondence theory for terminological logics: preliminary report. In R. Reiter and J. Mylopoulos, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia, Aug. 1991. Morgan Kaufmann.

[17] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.