Extension and Equivalence Problems for Clause Minimal Formulas

Hans Kleine Büning
Department of Computer Science
Universität Paderborn
D-33095 Paderborn (Germany)
E-mail: kbcsl@uni-paderborn.de

Xishun Zhao
Institute of Logic and Cognition
Zhongshan University
510275, Guangzhou (P. R. China)
Email: lcxszhao@hotmail.com

Abstract

Inspired by the notion of minimal unsatisfiable formulas we first introduce and study the class of clause minimal formulas. A *CNF* formula F is said to be clause minimal if any proper subformula of F is not equivalent to F.

We investigate the equivalence and extension problems for clause minimal formulas. The extension problem is the question whether for two formulas F and H there is some formula G such that $F+G$ is equivalent to H. For various parameters and constraints we discuss the complexity of these problems.

1 Introduction

We are mainly interested in the problem whether for pairs of *CNF* formulas F and H there exists a formula G, such that $F + G$ and H are logically equivalent. This problem, called extension problem, often occurs during the development and verification of knowledge based systems. In the general case of arbitrary *CNF* formulas the question of deciding whether there is some G with $F + G \approx H$ is equivalent to the question of $H \models F$. In practice, several parameters and constraints play an important role: the number of added clauses, minimality, deficiency, fixed goal formula, etc.

The term minimal has been used in the literature with different meanings. One definition reflects the length of the formulas, i.e. the number of occurring literals. Another one counts the number of different clauses. In order to make clear that we are dealing with numbers of clauses we use the term *clause minimal*. Similarly, to minimal unsatisfiable formulas, where removing a clause leads to a satisfiable formula, we say a formula F is *clause minimal* if removing a clause f from F results in a formula not equivalent to F, i.e. $F - \{f\} \not\approx F$.

An useful feature is the deficiency of a formula defined as the difference between the number of clauses and the number of variables. Let *MU* be the set of minimal unsatisfiable *CNF* formulas. Then *MU*(k) is the set of *MU* formulas with deficiency k. It is known that for fixed k the class *MU*(k) is solvable in polynomial time [6].

Our main goal is to see for which classes the equivalence and the extension problems are tractable. We use the following notation: A literal is a propositional variable or a negated propositional variable. A clause is a disjunction of some literals. A *CNF* formula is a conjunction of some clauses. We often regard formulas as sets of clauses. Thus, $F + G$ is used to denote the union of F and G. $F - \{f\}$ is the formula obtained by deleting the clause f from F. Given a formula F, we use $var(F)$ to denoted the set of variables occurring in F. $|F|$ is the number of clauses in F. $|var(F)|$ is the number of variables in $var(F)$. If G is a subformula of F, i.e. a subset of the clauses of F, then we write $G \subseteq F$. The deficiency of F, $d(F)$, is defined as $|F| - |var(F)|$; while the maximal deficiency $d^*(F)$ is defined as $d^*(F) = \max\{d(G) \mid G \subseteq F\}$.

Next we give a brief review of the relevant notions of complexity theory. For the classes Δ_k^P, Σ_k^P, and Π_k^P of the polynomial hierarchy please see [3]. The class $P^{NP[\log n]}$, consists of all problems solvable in polynomial time with $O(\log n)$ queries to an NP oracle [8]. It is known that $P^{NP[\log n]}$ coincides with the class of all problems solvable with parallel $O(n)$ queries to NP oracles. The class D^P consists of all problems which can be described as the intersection of a NP problem and a co-NP problem. The minimal unsatisfiability problem is D^P-complete[7]. The unique satisfiability problem is in D^P, but its D^P-completeness is open. The notion of completeness is based on many-one polynomial time transformations.

2 Clause Minimality and Equivalence

At first we introduce the class of clause minimal formulas without and with fixed deficiency.

Definition 1 *(Clause minimality)*
CL–MIN $:= \{F \in CNF \mid \forall f \in F : F - \{f\} \not\approx F\}$
CL–MIN$(k) := \{F \in$ *CL–MIN* $\mid d(F) = k\}$ *for* $k \in \mathbb{Z}$

The problem CL–MIN and the problem $SAT \cap CL$–MIN are known to be NP–complete [7]. Unsatisfiable CL–MIN formulas are exactly the minimal unsatisfiable formulas MU . Therefore, $UNSAT \cap CL$–MIN is D^P–complete. For fixed deficiency k the class $UNSAT \cap CL$–$MIN(k)$ is the set $MU(k)$, which is solvable in polynomial time [6]. But the class CL–$MIN(k)$ remains NP–complete. That can be seen as follows:
We define a polynomial transformation from CL–MIN to CL–$MIN(k)$. For a formula F with deficiency t, we proceed by case distinction.
Case 1: $t < k$: Let $\{g\} + G$ be a MU formula with deficiency $k - t + 1$ over new variables $y_1, \ldots, y_{k-t+1}$, where g is a clause and G is a formula. For an arbitrary but fixed clause f from F we construct the formula $F^* = (F - \{f\}) + \{f \vee g)\} + G$. Then F^* is in CL–$MIN(k)$ iff $F \in CL$–MIN.
Case 2: $t > k$: Add to the formula F a new clause $(y_1 \vee \ldots \vee y_r)$, where $y_1, \ldots, y_r$ are new variables and $r = t-k+1$. Then we obtain $F \in CL$–MIN iff $F + \{(y_1 \vee \ldots \vee y_r)\} \in CL$–$MIN(k)$.
Case 3: If $t = k$ then take the formula F.

Since $UNSAT \cap CL$–$MIN(k)$ is solvable in polynomial time and CL–$MIN(k)$ is NP–complete, we obtain the NP–completeness of $SAT \cap CL$–$MIN(k)$.

Similar to the case of unsatisfiable formulas, for uniquely satisfiable formulas the complexity for CL–MIN formulas increases (assuming *Unique*–SAT is not in NP), whereas for fixed deficiency k the problem is solvable in polynomial time. The proof is based on a transformation to a $MU(k)$ problem. A detailed proof of the following Theorem can be found in the appendix.

Theorem 1
1. *Unique–SAT $\cap$ CL–MIN is at least as hard as Unique–SAT.*
2. *For fixed k the problem Unique–SAT $\cap$ CL–MIN(k) is solvable in polynomial time.*

2.1 Equivalence

The equivalence problem, i.e. the problem whether formulas F and H are logically equivalent, can be considered as a special case of an extension problem, namely only the empty formula can be added. For clause minimal formulas we define

Definition 2 *For a fixed CNF formula α we define:*
1. $MIN\text{-}EQU(\alpha) := \{F \mid F \approx \alpha \text{ and } F \in CL\text{–}MIN\}$
2. $MIN\text{-}EQU := \{(F, G) \mid F \approx G \text{ and } F, G \in CL\text{–}MIN\}$

It is well–known that the problem whether two formulas are logically equivalent is coNP–complete. The D^P–hardness of *MIN-EQU* and *MIN-EQU*(α) for fixed α can be proved in a straight forward way by a reduction of the D^P–complete problem *MU*.
For *MIN-EQU* we choose the fixed formula $H = \{\sqcup\}$. Then $(F, H) \in$ *MIN-EQU* if and only if $F \in$ *MU*.
For *MIN-EQU*(α) w.l.o.g. we assume that α is clause minimal. If α is unsatisfiable then a formula F is in *MU* if and only if $F \in$ *MIN-EQU*(α). In case of a satisfiable formula we construct for every formula F a formula $F(\alpha)$ as follows: If F and α have common variables then we rename the variables of F, such that F and α have distinct variables. We define $F(\alpha) = \{f \vee g \mid f \in F, g \in \alpha\}$. Then we see that $F \in$ *MU* if and only if $F(\alpha) \in$ *MIN-EQU*(α). The direction from left to right is obvious. For the other direction suppose F is satisfiable. Since F and α have distinct variables, there is a satisfying truth assignment for $F(\alpha)$ for which α is false. Hence, F is unsatisfiable. Because of the clause minimality of $F(\alpha)$, the formula F must be clause minimal and therefore in *MU*.

For a fixed formula α and an arbitrary formula F we have: $\alpha \models F$ if and only if $\alpha \models \{f_1, \cdots, f_m\}$, where $F = \{f_1 \vee g_1, f_2 \vee g_2, \cdots, f_m \vee g_m\}$ and for each i, $var(f_i) \subseteq var(\alpha)$ whereas $var(g_i) \cap var(\alpha)$ is empty. Therefore, to verify $\alpha \models F$ it suffices to check $\alpha \models \{f_1, \cdots, f_m\}$.

Based on this simple observation and because the satisfiability problem for *2CNF* is solvable in linear time one can show (see also [4]): For a fixed formula α, the class *MIN-EQU*$(\alpha) \cap (2 + t) -$ *CNF* is solvable in polynomial time, if t is the minimal length of clauses π with $\alpha \models \pi$.

With respect to tractability and for fixed deficiency k the problem *MIN-EQU*$(k)(\alpha)$ is equivalent to some problem *MU*$(k + r)$, where r depends on α.

Lemma 1 *Let α be a fixed satisfiable formula, where $var(\alpha) \subseteq var(\sigma)$ for any formula σ equivalent to α. Further, suppose β is a clause minimal CNF formula for which $var(\beta) \subseteq var(\alpha)$ and $\beta \approx \neg\alpha$.*
If F is a CNF formula with $\alpha \models F$ and deficiency $d(F) = k$. Then
$F \in$ MIN-EQU$(k)(\alpha)$ if and only if $F + \beta \in$ MU$(k + |\beta|)$.

Proof: We first prove the direction from right to left. $\beta \models \neg F$ follows from $F + \beta \in MU$. Since $\beta \approx \neg\alpha$ and $\alpha \models F$, we obtain $\alpha \approx F$. $F + \beta \in MU$ implies the clause minimality of F. Thus, we get $F \in \alpha$*MIN-EQU*$(k)(\alpha)$.

For the other direction we suppose $F \in$ *MIN-EQU*$(k)(\alpha)$. Since $F + \beta$ is unsatisfiable, it remains to show that each proper subformula of $F + \beta$ is satisfiable. Consider a clause $f \in F$. Since F is clause minimal, $F - \{f\}$ is strictly weaker than F, and hence than α. That means, $(F - \{f\}) + \beta$ is satisfiable. Next we consider any clause h in β. Since β is also clause minimal, by the same reason we get that $F + \beta - \{h\}$ is satisfiable. Altogether, we have shown that $F + \beta$ is minimal unsatisfiable. ∎

Since $\alpha \models F$ can be decided in time linear in the length of F, β can be constructed in constant time, and $MU(k + |\beta|)$ is solvable in polynomial time, we obtain the following corollary.

Corollary 1 *For any fixed CNF formula α and any fixed k, the problem MIN-$EQU(k)(\alpha)$ can be solved in polynomial time.*

The problem whether for a formula F and an integer t there is an equivalent formula G with at most t clauses, has been investigated intensively for definite Horn formulas. In terms of directed hypergraphs in [2] it has been shown that this problem is *NP*–complete. The restriction to clause minimal Horn formulas with or without fixed deficiency preserves the *NP*–completeness, since the clause minimality is decidable in quadratic time for Horn formulas. Now we fix the number of clauses and define

Definition 3 $EQU[t] := \{F \in CNF \mid \exists G : |G| \leq t \text{ and } G \approx F\}$

The proof of the following lemma can be found in the appendix, where the first three statements are based on simple reductions. The proof of the last statement is based on the calculation of prime implicants of the input Horn formula F.

Lemma 2 *For fixed $t \geq 1$*
1. *$EQU[t]$ is coNP-hard.*
2. *$EQU[t] \cap CL$–MIN is D^P–hard.*
3. *$EQU[t] \cap CL$–$MIN(k)$ is D^P–hard.*
4. *$EQU[t] \cap Horn$ is solvable in polynomial time.*

3 Extensions

The extension problem is the problem of determining whether for pairs of formulas (F, H) there is a formula G for which $F + G \approx H$ and $var(G) \subseteq var(H + F)$. Without any restriction this question is equivalent to the question whether $H \models F$. The latter problem is known to be $coNP$–complete.
We do not consider the problem for arbitrary but fixed goal formulas and look for extensions for *MU* formulas. Additionally, we demand that all formulas in consideration are clause minimal.

Definition 4
1. MIN-EXT $=$
$\{(F,H) \mid \exists G : F+G \approx H,\ var(G) \subseteq var(H+F),\ and\ F+G, H \in CL\text{–}MIN\}$.

2. MU–EXT $= \{F \mid \exists G:\ F+G \in MU\ and\ var(G) \subseteq var(F)\}$.

The next theorem shows that the problem *MU–EXT* is *NP*-complete.

Theorem 2 *MU–EXT* $=$ *CL–MIN.*

Proof: Let $F \in CNF$ be given. Suppose F can be extended to a minimal unsatisfiable formula. Then it is easy to see that F is clause minimal. For the converse direction we suppose $F \in CL\text{–}MIN$. We proceed by a case distinction:

Case 1: F is unsatisfiable. Then we have $F \in MU$ and therefore $F \in MU\text{–}EXT$.

Case 2: F is satisfiable. We have to show $F \in MU\text{–}EXT$. We will construct a formula G, such that $F+G \in MU$.
Let $\{t_1, \cdots, t_r\}$ be the set of satisfying truth assignments for F. And let $G = \{C_{t_1}, \cdots, C_{t_r}\}$, where $C_{t_i} = \{\neg L \mid t_i(L) = 1, L \text{ literal }\}$. Then $G \approx \neg F$ and hence, $F+G$ is unsatisfiable. Clearly, deleting any C_{t_i} form $F+G$ leads to satisfiable formula. For any $f \in F$, since F is clause minimal, we know that $F - \{f\}$ is not equivalent to F. Therefore, $(F-\{f\})+G$ is satisfiable. Consequently, $F+G \in MU$. Hence $F \in MU\text{–}EXT$. ∎

Since the clause minimality of Horn–formulas can be decided in quadratic time we immediately obtain the following corollary.

Corollary 2 *Horn* $\cap$ *MU–EXT is solvable in polynomial time.*

There are clause minimal Horn formulas for which any extension to a *MU* formula is a non–Horn formula. Take for example the Horn formula $F = (\neg a \vee b) \wedge (a \vee \neg b)$. Any extension G of F with $F+G \in MU$ is a non-Horn–formula. A possible extension is $G = (\neg a \vee \neg b) \wedge (a \vee b)$. Here the reason is that the clauses of F form a cycle $(a \to b), (b \to a)$. Please note, that $F+G$ has deficiency 2.
But cycles are not the only reason. Consider the following formula:
$F = (a \vee \neg x) \wedge (a \vee \neg y)$. Any extension G of F leads to a non–Horn formula, take for example $G = (\neg a) \wedge (x \vee y)$. Please note, that $F+G$ has deficiency 1.

If we demand that additionally the extension must be also a Horn formula, then we can show

Lemma 3 *$\{F \in Horn \mid \exists\, G \in Horn : F + G \in MU\}$ can be solved in polynomial time.*

Proof: Since unsatisfiable Horn formulas are refutable by unit resolution, any Horn MU-formula consists of $n+1$ clauses and is therefore in $MU(1)$. Furthermore, any variable occurs at most once positively in the formula and there exists exactly one negative clause. Thus, each Horn MU-formulas must have the following form

$$\begin{pmatrix} - & + & & & & \\ * & * & + & & & \\ * & * & * & + & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ * & * & * & * & \cdots & + \end{pmatrix}$$

where each row represents a variable, each column represents a clause, and entries "*" are wildcards for "$-$" or "0". From this structural property the following statement can easily be shown by induction: For any Horn-formula F which is clause minimal over variables $x_1, .., x_n$, there is a Horn-formula G such that $F + G \in MU$ if and only if F^* in MU, where

$$F^* = \begin{cases} F + Hypo(F), & \text{if } F \text{ contains a negative clause} \\ F + Hypo(F) + \{\neg x_1 \vee \cdots \vee \neg x_n)\}, & \text{if } F \text{ contains no negative clause,} \end{cases}$$

and $Hypo(F)$ is the conjunction of all x_i occurring only negatively in F. That implies the solvability of the MU-extension problem for Horn-formulas in quadratic time. ■

The complexity of the question whether for fixed k a Horn formula has a extension to a $MU(k)$ formula is open. More generally, we can consider MU–extensions for which the number of clauses and the deficiency of the goal formula is fixed. We therefore define

Definition 5 *For fixed $k, t \geq 1$:*
1. $MU\text{–}EXT[\texttt{ext} = t] = \{F \mid \exists G : F + G \in MU, var(G) \subseteq var(F), |G| = t\}$
2. $MU\text{–}EXT[\texttt{ext} = t, \texttt{defi} = k] =$
$$\{F \mid \exists G :\ F + G \in MU(k), var(G) \subseteq var(F), |G| = t\}$$

For a fixed number of clauses t the problem remains intractable, whereas for fixed deficiency k and here adding one clause the problem is solvable in polynomial time. The proof of the next theorem can be found in the appendix.

Theorem 3 *For fixed k and t:*
1. MU–EXT$[\texttt{ext} = t]$ *is* D^P*-hard,*
2. MU–EXT$[\texttt{ext} = 1]$ *is in* $P^{NP[\log n]}$,
3. MU–EXT$[\texttt{ext} = 1, \texttt{defi} = k]$ *is solvable in polynomial time.*

4 Conclusion and Future Work

At least two questions remain open: How complex is the task to decide whether there is an extension of t clauses resulting in a formula in $MU(k)$. It is strongly conjectured that for fixed t and k the problem $MU\text{–}EXT[\texttt{ext} = t, \texttt{defi} = k]$ is solvable in polynomial time.
The other open problem is the complexity of the question whether for fixed k and for a Horn formula F there exists an extension $F + G \in MU(k)$.

References

[1] Aharoni, R., Linial, N.: Minimal Non-Two-Colorable Hypergraphs and Minimal Unsatisfiable Formulas. *Journal of Combinatorial Theory* **43** (1986), 196–204

[2] Ausiello, G., D'Atri, A., Sacca, D., Minimal representation of directed hypergraphs. *SIAM Journal on Computing* **2** (1986), 418–431

[3] Johnson, D. S., A Catalog of Complexity Classes, in *Handbook of Theoretical Computer Science*, Vol A, Elsevier, Amsterdam, 1990

[4] Kleine Büning, H., Lettmann, T.: *Propositional Logic: Deduction and Algorithm*, Cambridge University Press, 1999

[5] Kleine Büning, H., Zhao, X. S.: The Complexity of Some Subclasses of Minimal Unsatisfiable Formulas, 2000, submitted for publication

[6] H. Fleischner, O. Kullmann, S. Szeider, Polynomial-time recognition of minimal unsatisfiable formulas with fixed clause-variable difference, to appear in *Theoretical Computer Science.*

[7] Papadimitriou, C.H., Wolfe, D.: The Complexity of Facets Resolved, *Journal of Computer and System Science* **37** (1988) 2-13

[8] K. Wadner, Bounded Query Classes, *SIAM Journal on Computing*, **19**(1990), 833–846

5 Appendix

Theorem 1
1. Unique–SAT $\cap$ CL–MIN is at least as hard as Unique–SAT.
2. For fixed k the problem Unique–SAT $\cap$ CL–MIN(k) is solvable in polynomial time.

Proof: *Ad 1:* Let $F = \{C_1, \cdots, C_n\}$ be a *3CNF* formula with $C_i = (L_{i,1} \vee L_{i,2} \vee L_{i,3})$. For new variables $x_1, \cdots, x_n$ we use π_i to denote the clause $(x_1 \vee \cdots \vee x_{i-1} \vee x_{i+1} \vee \cdots \vee x_n)$. Now we define $\omega(F) = \sigma(F) \wedge (x_1 \vee \cdots \vee x_n)$, where $\sigma(F) = \bigwedge_{1 \leq i \leq n}(C_i \vee \pi_i) \wedge \bigwedge_{1 \leq i \leq n, 1 \leq j \leq 3}(\neg L_{i,j} \vee \pi_i \vee \neg x_i) \wedge \bigwedge_{1 \leq i < j \leq n}(\neg x_i \vee \neg x_j)$

It is known [5] that F is satisfiable if and only if $\omega(F) \in MU$, and F is unsatisfiable if and only if $\sigma(F) \in MU$.

Since $\sigma(F)$ is a subformula of $\omega(F)$, we see that $\sigma(F) \in$ *CL–MIN* in both cases.

Now we prove that F is uniquely satisfiable if and only if $\sigma(F)$ is uniquely satisfiable. For the *only if* part we suppose $F \in$ *Unique–SAT*. Let t_1 and t_2 be two satisfying truth assignments of $\sigma(F)$. Since $\sigma(F) \wedge (x_1 \vee \cdots \vee x_n)$ is unsatisfiable, it follows that $t_1(x_i) = t_2(x_i) = 0$. From the definition of $\sigma(F)$ we know that $t_1(F) = 1$ and $t_2(F) = 1$. Then $t_1 = t_2$ because of the unique satisfiability of F. For the other direction please note that every satisfying truth assignment of F can be extended to a truth assignment satisfying $\sigma(F)$. Thus, if $\sigma(F)$ is uniquely satisfiable then $F \in$ *Unique–SAT*.

Altogether we have shown, $F \in$ *Unique–SAT* if and only if $\sigma(F) \in$ *Unique–SAT* $\cap$ *CL–MIN*. That means, *Unique–SAT* $\cap$ *CL–MIN* is at least as hard as the unique satisfiability problem.

For the proof of part 2 dealing with uniquely satisfiable formulas with fixed deficiency we need the following lemma.

Lemma 4 *Suppose I is a satisfying truth assignment for a formula F. Let C_I be the disjunction of literals L for which $I(L) = 0$. Then $F \in$ Unique–SAT $\cap$ CL–MIN(k) if and only if $F + \{C_I\} \in MU(k+1)$.*

Proof: ($\Rightarrow$) Suppose $F \in$ *Unique–SAT* $\cap$ *CL–MIN*(k). Since F is uniquely satisfiable, it follows that $F^* := F + \{C_I\}$ is unsatisfiable. Because F is in *CL–MIN*, $F - \{f\}$ is not equivalent to F for any $f \in F$. That means $(F - \{f\}) + \{C_I\}$ is satisfiable for any $f \in F$. Therefore, removing any clause from F^* leads to a satisfiable formula. Hence, F^* is minimal unsatisfiable consisting of $n + k + 1$ clauses.

($\Leftarrow$) Suppose $F+\{C_I\} \in MU(k+1)$. Then $F \models \neg C_I$ implies the unique satisfiability of F. Since $F + \{C_I\}$ is minimal unsatisfiable, we obtain $F \in$ *Unique–SAT* $\cap$ *CL–MIN*(k). ■

Now we proceed with the proof of part 2 of Theorem 1.
The deficiency $d(F)$ of a formula F is the difference between the number of clauses and the number of variables. It is known that for fixed k the satisfiability of formulas with $d^*(F) = max\{d(F') \mid F' \subseteq F\} \leq k$ can be solved in polynomial time [6]. Please note that if $G \in MU$ then the deficiency of any proper subformula of G is less than $d(G)$. By Lemma 4 we know that if $F \in$ *Unique–SAT* $\cap$ *CL–MIN*(k), then $d^*(F) \leq k+1$.
Let $\mathcal{A}_{k+1}$ be a satisfiability algorithm deciding in polynomial time whether a formula H with $d^*(H) \leq k+1$ is satisfiable. At first we apply $\mathcal{A}_{k+1}$ to F. We do not know whether F has the property $d^*(F) \leq k+1$. If the algorithm $\mathcal{A}_{k+1}$ returns 'unsatisfiable' or runs longer than expected then F is not in *Unique–SAT* $\cap$ *CL–MIN*(k), because of Lemma 4. Suppose $\mathcal{A}_{k+1}$ returns a truth assignment I. Then we check whether I satisfies F. If this is not the case then F is not in *Unique–SAT* $\cap$ *CL–MIN*(k). Otherwise we test whether $F + \{C_I\} \in MU(k+1)$ by means of the well-known algorithm for $MU(k+1)$. If this is the case then we know F is in *Unique–SAT* $\cap$ *CL–MIN*(k). ■

Lemma 2 *For fixed* $t \geq 1$
1. $EQU[t]$ *is coNP-hard.*
2. $EQU[t] \cap$ *CL–MIN is* D^P*–hard.*
3. $EQU[t] \cap$ *CL–MIN*(k) *is* D^P*–hard.*
4. $EQU[t] \cap$ *Horn is solvable in polynomial time.*

Proof: Ad 1 and 2: For any formula F and $t \geq 1$ we construct the formula $\sigma(F,t) := \bigwedge_{1 \leq i \leq t+1}(F_i \vee x)$, where x is a new variable and F_i is a copy of F with different variables. Then F is unsatisfiable resp. in *MU* iff $\sigma(F,t)$ in $EQU[t]$ resp. $\sigma(F,t) \in EQU[t] \cap$ *CL–MIN*.
If F is unsatisfiable resp. in *MU* then $\sigma(F,t) \approx x$ and therefore in $EQU[t]$ resp. in $EQU[t] \cap$ *CL–MIN*. Suppose $\sigma(F,t)$ in $EQU[t]$ resp. $\sigma(F,t) \in EQU[t] \cap$ *CL–MIN*. Since we have $t+1$ copies of F with distinct variables, we get $\sigma(F,t) \approx x$. From the D^P–completeness of *MU* we obtain our desired result.

Ad 3: (Reduction from $EQU[t] \cap$ *CL–MIN*)
Let k be fixed. To any F and t we associate the formula $\omega(F)$ as follows:
Suppose F has deficiency r. If $r \geq k$, then we define $\omega(F) := F \vee (x_1 \vee \ldots \vee x_s)$, where x_i are new variables and $s = r - k$. Then we obtain $F \in EQU[t] \cap$ *CL–MIN* iff $\omega(F) \in EQU[t] \cap$ *CL–MIN*(k).

If $r < k$, let f be a clause in F and H a minimal unsatisfiable formula with deficiency $s = k - r + 1$. We define $\omega(F) := (F - \{f\}) + (f \vee H)$. Then we obtain $F \in EQU[t] \cap CL\text{–}MIN$ iff $\omega(F) \in EQU[t] \cap CL\text{–}MIN(k)$. Using part 2 of the lemma we obtain the D^P–hardness of $EQU[t] \cap CL\text{–}MIN(k)$.

Ad 4: If a Horn formula F implies a non-Horn clause g then there is some Horn clause $g' \subseteq g$, such that $F \models g'$. Therefore, if a Horn formula F has an equivalent formula with t clauses, then there is Horn formula consisting of at most t prime implicants equivalent to F.
The idea of the proof of the extension problem is as follows:
1.) At first we show: There is a polynomial p, such that for any Horn formula G consisting of t clauses the number of prime implicants of G is less than $2^{p(t)}$.
2.) In a second step we prove: There is a polynomial q, such that the prime implicants of a Horn formula F can be computed iteratively, where the calculation of m prime implicants costs less than $q(lenght(F), n^m)$ time.

Using these facts we can decide the extension problem in polynomial time:
For a fixed t let F be a Horn formula. If F is unsatisfiable then the shortest equivalent formula has the form $x \wedge \neg x$. That means the formula consists of two clauses. Now we assume that F is satisfiable. If there is an equivalent formula G with $\mid G \mid \leq t$ then there is an equivalent Horn-formula G' with $\mid G' \mid \leq t$ and G' consists of prime implicants.
Then G' has at most $2^{p(t)}$ prime implicants. Therefore, F has at most $2^{p(t)}$ prime implicants. Now we calculate iteratively the prime implicants of F. We have to compute at most $2^{p(t)}$ prime implicants. That costs at most $q(length(F), n^{2^{p(t)}})$ time. Since t is fixed, the running time is bound by a polynomial in the length of F. If the number of the prime implicants exceeds $2^{p(t)}$ then there is no desired formula. Otherwise, we check for all subsets of the prime implicants with t clauses whether the subset is equivalent to F. If there is no equivalent subset then we know that $F \notin EQU[t] \cap Horn$.

Now it remains to prove the statements 1 and 2:
Suppose a Horn formula G consists of t clauses. The variables of $G = \{g_1, \cdots, g_t\}$ can be divided into a set of variables V^+ which occur positively in G and the set of variables V^- which occur only negatively in G. For every g_i let g'_i be the maximal subclause of g_i containing only literals over V^-. Any prime implicant of G may contain combinations of g'_i. But there are only 2^t different combinations. If we remove the subclauses g'_i from g_i then the resulting formula G^* has at most t variables (at most t different variables occur positively in G). But there are only $2^{p_1(t)}$ different formulas with t clauses over at most t variables. p_1 is some polynomial. If we combine both subclauses and count the different clauses then we immediately obtain our desired estimation.

Finally, it remains to show that the prime implicants of F can be calculated iteratively in polynomial time depending on the number of prime implicants and the length of the formula.
It suffices to show that for a Horn formula $F = \{f_1, \cdots, f_m\}$ consisting of prime implicants a further prime implicant can be computed in polynomial time in the length of F and m. We choose some pair of clauses f_i and f_j which can be resolved and have not resolved so far. Let g be the resolvent. Now we decide, whether g contains a new prime implicant. Suppose the clause f_r $(1 \leq r \leq m)$ consists of n_r literals. If f_1 is a subclause of g, then we remove the j_1^{th} literal of f_1 from g for $(1 \leq j_1 \leq n_1)$. The resulting clause is denoted by $g[j_1]$. If f_2 is subclause of $g[j_1]$ then we perform the same steps obtaining the clause $g[j_1, j_2]$. Finally, in worst case we obtain the clause $g[j_1, \cdots, j_m]$. Please note, that there are at most $n_1 \times \cdots \times n_m$ different clauses. For these clauses we test whether they contain a new prime implicant of F. Altogether, the calculation costs at most $O(n^m n^3 length(F)^3)$ time, where n is the number of variables of F. If no new prime implicant has been found, then we continue with the next pair of resolvable clause, not resolved previously. ■

Theorem 3 *For fixed k and t:*
1. MU–EXT[$\texttt{ext} = t$] is D^P-hard,
2. MU–EXT[$\texttt{ext} = 1$] is in $P^{NP[\log n]}$,
3. MU–EXT[$\texttt{ext} = 1, \texttt{defi} = k$] is solvable in polynomial time.

Proof: *Ad 1:* We will define a reduction of the D^P–complete MU-problem. For any formula F and new variables $x_1, \cdots, x_t$ we define

$$F^t = \{(f \vee x_1 \vee \cdots \vee x_t) \mid f \in F\}.$$

Next, we prove that for any $t \geq 1$, $F^t \in MU$–$EXT[\texttt{ext} = t]$ if and only if $F \in MU$.

Obviously, if $F \in MU$ then the extension $\neg x_1 \wedge \cdots \wedge \neg x_t$ leads to a MU formula. For the other direction, at first we consider the case $t = 1$. Suppose $F^1 \in MU$–$EXT[\texttt{ext} = 1]$. Since x_1 is a pure literal, the added clause must contain $\neg x_1$. If the added clause contains two or more literals, then the extended formula will be satisfiable. Hence, the added clause is $\neg x_1$. Now we can see that F is minimal unsatisfiable.
For $t > 1$ we suppose $F^t \in MU$–$EXT[\texttt{ext} = t]$. Now we split F^t on x_t. The one splitting formula F_{x_t} contains F^{t-1}. Since $\neg x_t$ must occur in some added clause, F_{x_t} contains at most $t-1$ additional clauses. That is to say, F^{t-1} can be extended to a minimal unsatisfiable formula by adding at most $t-1$ clauses. Please note that the shortest MU-formula containing the clause $(x_1 \vee \cdots \vee x_{t-1})$ has t clauses. Then F^{t-1} can not be extended to a MU-formula by adding less than $t-1$ clauses. Now, by the induction hypothesis, we get that F is in MU.

Ad 2: Let F be a formula over the variables $x_1, \dots, x_n$. For the clause $C = \{\neg L \mid F \models L\}$ we show that $F + C \in MU$ if and only if $F \in MU\text{–}EXT[\texttt{ext} = 1]$. The direction from left to right is trivial. The other direction can be seen as follows. Suppose $F \in MU\text{–}EXT[\texttt{ext} = 1]$. Then there is some clause f with $F + \{f\} \in MU$. For any literal L in f, we have $F \models \neg L$. Otherwise $F + \{f\}$ would be satisfiable. Consequently, f is a subclause of C. Since $F + C$ is unsatisfiable we obtain $F + C \in MU$.

Thus to test whether $F \in MU\text{–}EXT[\texttt{ext} = 1]$ we compute the clause C and then check whether $F + \{C\} \in MU$. To construct the clause C needs $2n$ many independent calls to a NP oracle for the satisfiability test. Checking if $F + \{C\} \in MU$ needs additionally 2 queries to the NP oracle. Since the satisfiability test can be performed with 2n parallel queries, the problem $MU\text{–}EXT[\texttt{ext} = 1]$ is in $P^{NP[\log n]}$.

Ad 3: Since for fixed k the satisfiability problem for formulas F with $d^*(F) \leq k$ can be solved in polynomial time, it follows that the problem $MU\text{–}EXT[\texttt{ext} = 1, \texttt{defi} = k]$ is solvable in polynomial time:
If there is a clause f with $F + \{f\} \in MU(k)$ then $d^*(F + \{L\}) \leq k + 1$ for any literal L. Let $\mathcal{A}_{k+1}$ be an pol–time algorithm deciding the satisfiability for formulas H with $d^*(H) \leq k + 1$. For every literal L over a variable of F we apply S_{k+1} to $F + \{\neg L\}$ in order to decide whether $F \models L$. If S_{k+1} returns 'unsatisfiable', then we know $F \models L$ or $d^*(F + \{L\}) \not\leq k + 1$. For the clause $f = \{\neg L \mid S_{k+1}$ applied to $F + \{\neg L\}$ returns 'unsatisfiable'$\}$ we test whether $F + \{f\} \in MU(k)$. If that is the case then we have found a desired extension. Otherwise there is no extension, since $d^*(F + \{L\}) \not\leq k + 1$. ■