The Complexity of Some Complementation Problems

David A. Plaisted

Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175
USA
e-mail: plaisted@cs.unc.edu

Gregory Kucherov

LORIA/INRIA-Lorraine
615, rue du Jardin Botanique
B.P. 101
54602 Villers-lès-Nancy France
e-mail: kucherov@loria.fr

Abstract

We study the computational complexity of the problem of computing a complement representation for a set of terms. Depending on the classe of sets considered, some sets are shown to have an exponential complement representation in the worst case, and some are shown to have a polynomial one. The complexity of deciding if a set has an empty complement is also studied.

Keywords: Term, ground term, ground instance, complement representation, computational complexity.

1 Introduction

In this paper we study the following natural problem. Given a set $S = \{t_1, \ldots, t_n\}$ of terms (or literals), find another set $c(S)$ with the set of instances consisting exactly of those ground terms which are not instances of terms of S. $c(S)$ is naturally called the *complement representation* of S.

Probably the first study of how to compute a complement representation has been done by Lassez and Marriot [LM87] in the context of learning a set specified by counter-examples. It can be easily observed that if terms of S contain only linear terms (i.e. terms without repetitive occurrences of variables) then the finite complement representation always exists. Lassez and Marriot proved that S has a finite complement representation if and only if S is "equivalent" to a set of linear terms, where "equivalence" means having the same set of instances. Moreover, this equivalence is decidable and such a set of linear terms can be always obtained by instantiating non-linear variables in S by some ground terms.

The notion of complement representation was (explicitely or implicitely) used in various application fields, such as functional and logic programming, automated deduction, machine learning. For example, in a logic program a clause L :- $L_1, L_2, \ldots, L_n$ will only be invoked if no earlier clauses succeed. Thus the set of goals for which this clause will be invoked is the intersection of the instances of L and the complement of the earlier instances. If we can compute a representation of the complement of the earlier instances, then we can compute the set of goals that will cause this clause to be invoked. This permits us to change the order of the clauses of the program without affecting its semantics. This could have applications in concurrent execution of logic programs, since it would be possible to attempt execution of all clauses at the same time without concern about altering the semantics of the program. As another exemple, work [FL96] discusses the complement problem and its application to model building, and gives an exponential algorithm. Computing a complement representation can be also viewed from a more general logic perspective as negation elimination in some class of equational formulae [Com91].

The notion of complement has particularly often occurred in the theory of term rewriting [DJ90]. For example, in [Kuc88] an algorithm has been proposed for proving inductive theorems in an equational theory, which is explicitely based on computing a complement. A generalization of the complementation for terms containing associative-commutative operations has also been studied [KLP91, LM93]. The notion of complement stands also behind more general notions studied in the term rewriting theory, such as sufficient completeness, ground reducibility, test set, the set of normal forms. The latter, for example, can be viewed as a generalization of the complement: a normal form is a ground term which not only is not an instance of a term from a given set S, but also does not contain a subterm which is such an instance.

There exists an extensive literature devoted to ground reducibility, the set of normal forms and related issues (some sample references are [KNZ86, JK89, BK89, Com91, HH94]). Related complexity questions have also been studied (see [Pla85, KT95, KNRZ91, KR95a, KR95b, CJ97]). However, to the best of our knowledge, no work has been done on analyzing complexity problems related to complement representation. The only result, trivially implied by the Lassez and Marriot's construction, is the exponential upper bound on the size of the complement representation. In this paper, we answer some further related complexity questions. We show that the size of the complement representation is necessarily exponential, in the worst case, for a set of linear terms and, even more restrictively, for a set of *disjoint* linear terms. On the other hand, we show that in case of ground terms, as well as in the special case of *hierarchical* set of linear terms, a polynomial complement representation exists. We also study the complexity of deciding if a set has an empty complement, and show that the problem is co-NP-complete for general sets of linear terms and polynomial (linear) for sets of disjoint linear terms.

2 Basic Definitions

$T(F, X)$ is the set of terms formed from function symbols in F and variables in X. $T(F)$ is the set of ground terms formed from function (and constant) symbols in F. A term $t \in T(F, X)$ is *linear* if no variable of X occurs more than once in t. We assume familiarity with basic notions of term rewriting [DJ90] (such as position in a term, substitution, matching, unification).

If t is a term, then $||t||$ is the *symbol size* of t, which is the number of occurrences of function and constant symbols in t. Thus $||f(a, f(a, f(b, x)))|| = 6$. If T is a set of terms, then $||T||$ is the sum of the symbol sizes of the terms in T. The *depth* of a symbol in a term is its depth in the term tree. The depth of a term is the maximal depth of a symbol in it. For example, the depth of variable x in the term above is 3 which is the depth of the term itself. We consider the order on positions in a term corresponding to the pre-order in the tree. For example, the leftmost variable in a term means the variable occurring at the smallest position with respect to the pre-order of positions.

Definition 2.1 If T is a set of terms, $Gr(T)$ is the set of ground terms in $T(F)$ that are instances of terms in T.

3 The Complexity of Some Complement Problems

Definition 3.1 If T and U are sets of terms, we say that U *represents the complement of* T if $Gr(U) = T(F) \setminus Gr(T)$.

We first consider the version of the complement representation problem for sets of ground terms. In this case, $Gr(U) = T(F) \setminus T$.

Definition 3.2 If t is a term and $\{x \leftarrow f(x_1, \ldots, x_n)\}$ is a substitution and the x_i are distinct from x and do not appear in t, then $\{x \leftarrow f(x_1, \ldots, x_n)\}$ is an *elementary substitution for* t. $t\{x \leftarrow f(x_1, \ldots, x_n)\}$ denotes the result of applying $\{x \leftarrow f(x_1, \ldots, x_n)\}$ to t.

The following notion will be very useful for our constructions.

Definition 3.3 Suppose $T \subseteq T(F)$ is a finite set of ground terms. Let $T^{\downarrow}$, the set of *prefixes* of T be the set $U \subseteq T(F, X)$ such that

(i) x is in U for some variable x,

(ii) if t is in U and y is the leftmost variable (with respect to the pre-order of positions) in t, and $\{y \leftarrow f(x_1, \ldots, x_n)\}$ is an elementary substitution for t, and $t\{y \leftarrow f(x_1, \ldots, x_n)\}$ matches some term in T, then $t\{y \leftarrow f(x_1, \ldots, x_n)\}$ is in U as well,

(iii) no other term is contained in U.

For example, if T is $\{f(a, b, e), f(a, c, d)\}$ then $T^{\downarrow}$ is $\{x, f(x_1, x_2, x_3), f(a, x_2, x_3), f(a, b, x_3), f(a, c, x_3), f(a, b, e), f(a, c, d)\}$. Note that for every position α in any $t \in T$, there is a term $s \in T^{\downarrow}$ having a variable at position α.

Lemma 3.4 *For any finite $T \subseteq T(F)$, $T^{\downarrow}$ is finite and $||T^{\downarrow}|| = \mathcal{O}(||T||^2)$.*

Proof: The finiteness is obvious. The quadratic bound follows from the fact that the function symbol f that we add to a term in $T^{\downarrow}$ according to (ii) of Definition 3.3 can be mapped to a distinct symbol f of T. Thus, only a linear number of such symbols can be added. Each symbol addition comes with copying of an existing term. This implies the quadratic bound. $\square$

Using the set of prefixes $T^{\downarrow}$ we now define another set which we will prove to be a complement representation for T.

Definition 3.5 Suppose $T \subseteq T(F)$ is a finite set of ground terms. Let $c(T)$ be the set of terms $u\{x \leftarrow f(x_1, \dots, x_n)\}$ such that $u \in T^{\downarrow}$, x is the leftmost variable in u, $\{x \leftarrow f(x_1, \dots, x_n)\}$ is elementary for u, and $u\{x \leftarrow f(x_1, \dots, x_n)\}$ does not match any term of T.

Theorem 3.6 *If $T \subseteq T(F)$ is a finite set of ground terms then $c(T)$ represents the complement of T. Moreover, assuming a constant size of F, $c(T)$ is computable in polynomial (quadratic) time on $||T||$, and $||c(T)||$ is polynomial (quadratic) in $||T||$.*

Proof: It is clear that if $s \in T(F)$ is an instance of some element of $c(T)$, then s is not in $Gr(T)$, since elements of $c(T)$ do not match any term of T. Conversely, if $s \in T(F) \setminus T$, then consider the smallest position α of s (with respect to the pre-order) such that there exists a term in T which coincides with s at all positions strictly smaller than α, but none of such terms has the same symbol as s at position α. Note that position α exists in some term of T. Let t be an element of $T^{\downarrow}$ which has its leftmost variable x at position α. By the remark before Lemma 3.4, such a term t exists. Then $t\{x \leftarrow f(x_1, \dots, x_n)\}$ is an element of $c(T)$ which has s as an instance.

The facts that $c(T)$ is computable in quadratic time and of quadratic size follow from Lemma 3.4. $\square$

Note that the quadratic upper bound of Theorem 3.6 is tight, as any complement representation of the set $T = \{h^n(a)\}$ has at least quadratic symbol size.

We now consider the extension of this problem to linear terms with variables.

Theorem 3.7 *There is a set T of linear terms such that any set U of terms representing the complement of T is of exponential size.*

Proof: Suppose the signature contains constant symbols a, b and c, and an n-ary symbol f. Let T be the terms $\{a, b, c, f(c, x_2, \dots, x_n), f(x_1, c, x_3, \dots, x_n), \dots, f(x_1, x_2, \dots, c)\}$. Suppose U represents the complement of T. Let u be a term in U. Then u is of the form $f(t_1, \dots, t_n)$ where the t_i are variables, constant symbols, or terms of the form $f(y_1, \dots, y_n)$. But no t_i can be a variable, since then U would not represent the complement of T. (Such a term u is unifiable with a term of T having c as the i-th argument.) Therefore, all the t_i must be a or b or of the form $f(y_1, \ldots, y_n)$. All combinations must be present in order to represent the complement of T, and this requires at least 3^n terms in U. $\square$

It is still possible that by placing more conditions on T, we could obtain sets of terms for which the complement problem is polynomially solvable. One can easily note that the proof above essentially uses the property that terms of T have common instances. Therefore, forbidding this would be a further natural restriction.

Definition 3.8 We say a set T of terms is *disjoint* if no two distinct terms in T are unifiable.

In other words, T is disjoint if for any $t_1, t_2 \in T$, $Gr(\{t_1\}) \cap Gr(\{t_2\}) = \emptyset$. It turns out that this condition is not enough to guarantee that the complement problem is polynomially solvable.

Theorem 3.9 *There are disjoint sets T of linear terms such that any set U of terms representing the complement of T is of exponential size on $||T||$.*

Proof: Let ϕ be a mapping from $\{(i,j) : 1 \leq i,j \leq n, i \neq j\}$ to $\{1, 2, ..., n(n-1)/2\}$ such that $\phi(i,j) = \phi(k,l)$ iff $\{i,j\} = \{k,l\}$. Let $m = n(n-1)/2$. Let T be a set of n linear terms $\{r_1, \ldots, r_n\}$ where r_i is of the form $f(t_1, \ldots, t_m)$ and $t_{\phi(j,k)} = a_i$ if $j = i$ or $k = i$, and $t_{\phi(j,k)}$ is a distinct variable otherwise.

The idea is that the arguments of f represent edges of an undirected complete graph on n vertices, with $t_{\phi(j,k)}$ representing the edge between vertices j and k, and r_i has a_i at all the positions corresponding to edges incident to the vertex i, and variables elsewhere.

Now, this is a disjoint set of terms, because r_i and r_j have a_i and a_j, respectively, as the $\phi(i,j)$ argument of f.

Suppose that U represents the complement of T. We claim that U has an exponential number of elements.

First we observe that every term u in U which is not a constant is of the form $f(t_1, \ldots, t_m)$, and at least $\lceil n/2 \rceil$ of the terms t_i are not variables. Otherwise there would exist j such that $t_{\phi(j,k)}$ is a variable for all $1 \leq k \leq n, k \neq j$, in which case u would be an instance of r_j.

Let us consider all terms of the form $f(t_1, \ldots, t_m)$ in which the t_i are in the set $\{a_1, a_2, \ldots, a_n\}$. There are n^m such terms altogether. By the above remark, each element u of U matches at most $n^{m - \lceil n/2 \rceil}$ of them, since at least $\lceil n/2 \rceil$ of the arguments of f are constants in u. How many terms are not instances of terms of T? It is $n^m - n \cdot n^{m-(n-1)}$ since T is disjoint and each term r_i in T has $n-1$ arguments non-variable. If U contains l terms $f(t_1, \ldots, t_m)$, we obtain the inequation $n^m - n \cdot n^{m-(n-1)} \geq l \cdot n^{m - \lceil n/2 \rceil}$, which implies the exponential lower bound $\Omega(n^{n/2})$ for l. $\square$

Finally, we consider more restrictive sets of linear terms in which the complement can be represented by a set of polynomial size.

Definition 3.10 We say that a set T of terms is *hierarchical* if T has at most one element, or there is some position α such that all terms t in T have a non-variable symbol at position α and coincide at all positions ancestor to α, there are at least two terms t_1, t_2 in T that have different symbols at position α, and the subsets of T having a given symbol at position α are also hierarchical.

For example, the following set of terms is hierarchical:

$$\{f(a,b,x), f(a,c,y), f(b,x,c), f(b,y,a), g(x,y,z)\}$$

This set is hierarchical because all terms have a non-variable at the top position, and those terms having a g at this position form a set of one element, which is hierarchical. Those terms having an f at the top position form a set of four elements that is also hierarchical, since the first arguments of all terms in this set $\{f(a,b,x), f(a,c,y), f(b,x,c), f(b,y,a)\}$ are all non-variable symbols. Those having an a in the first argument ($\{f(a,b,x), f(a,c,y)\}$) are also hierarchical, based on the second argument, and those having a b in the first argument ($\{f(b,x,c), f(b,y,a)\}$) are hierarchical, based on the third argument. Note that hierarchical sets are useful in automated deduction (cf [BR93]), as they allow a natural application of case analysis in deduction procedures.

Theorem 3.11 *Suppose T is a hierarchical set of linear terms. Then T is disjoint, and moreover, the complement of T has a polynomial representation.*

Proof: The fact that T is disjoint follows readily from the definition of hierarchical. The fact that it has a polynomial size representation of the complement follows by an algorithm similar to that for Theorem 3.6 above, but instead of considering the pre-order of positions in T, we consider them in the order given by the hierarchy. $\square$

However, even if a set is hierarchical, it may take some work to detect this fact. Note that the union of hierarchical sets is not hierarchical, since any singleton set is hierarchical, and any set can be expressed as a union of singleton sets.

We finish with some results about how hard it is to test if the complement is empty. The following theorem is equivalent to a result proved in Section 3 of [KNRZ91] in the context of testing sufficient completeness of a rewriting system over free constructors.

Theorem 3.12 ([KNRZ91]) *Given a set T of terms, it is co-NP-complete to determine if $Gr(T) = T(F)$.*

Note that Theorem 3.12 was proved in [KNRZ91] for any set of terms, containing possibly non-linear terms. However, the co-NP-hardness was proved for sets of linear terms, which implies that the problem is still co-NP-complete for this restricted case.

However, this problem becomes easier if the set T is required to be disjoint.

Theorem 3.13 *Given a disjoint set T of linear terms, there is a polynomial (linear) time algorithm to test if $Gr(T) = T(F)$.*

Proof: For any natural n, define $Q(n) \subseteq T(F, X)$ as follows. $Q(n)$ consists of all those linear terms which have each non-variable symbol at the depth at most n, and each variable, if any, at the depth exactly $(n+1)$. Note that for any n, $Gr(Q(n)) = T(F)$. On the other hand, it can be easily seen that any two distinct terms in $Q(n)$ are disjoint.

Let d be the maximal depth of terms in T, and consider $Q(d)$. For any $t_1 \in T$, $t_2 \in Q(d)$, either t_1 and t_2 are not unifiable, or t_2 is an instance of t_1. This is implied by the linearity of t_1 and by the fact that any variable in t_2 is deeper than the depth of t_1. This observation further implies that $Gr(T) = T(F)$ if and only if every term of $Q(d)$ is matched by a term from T. Since T is also disjoint, no term of $Q(d)$ can be matched by distinct terms of T. Therefore, it is sufficient to count the number of terms of $Q(d)$ matched by each term of T and to check if these numbers sum up to $|Q(d)|$.

$|Q(n)|$ can be computed by induction. Note that $|Q(0)| = |F|$. Let F_k, $k \geq 0$, be the symbols of F of arity k, and $|F_k|$ their number. Then $|Q(n)| = |F_0| + \sum_{k \geq 1} |F_k| \cdot (|Q(n-1)|)^k$. Assuming the signature of constant size and a constant time of arithmetic operations, values $|Q(n)|$ for $n = 1..d$ can be computed in time $\mathcal{O}(d)$.

Let $t \in T$, and assume t has l variables occurrences at the depths $d_1, \ldots, d_l$. Then the number of terms of $Q(d)$ matched by t is $M(t) = \prod_{i=1}^{l} |Q(d - d_i)|$. To check that $Gr(T) = T(F)$ we have to check that $\sum_{t \in T} M(t) = |Q(d)|$. This test takes $\mathcal{O}(||T||)$ operations altogether. □

4 Discussion

We have studied the complexity of some complement problems involving finite sets of terms. It would be interesting to extend this work in two ways. One area of interest is to find more applications where these results would be useful. Another possibility is to find new classes of terms for which one can prove lower or upper bounds on the complexity of the complementation problem. It would also be possible to investigate the complexity of the complementation problem relative to various equational theories. In general, the study of disunification to date has emphasized completeness results rather than results about the size of the representation of the complement set. Perhaps other disunification problems could be studied to determine bounds on the sizes of the sets of disunifiers returned.

5 Acknowledgements

We are grateful to Miki Hermann and Laurent Juban for helpful discussions, and to Michaël Rusinowitch for his useful remarks. The first author was supported by Hélène Kirchner's group at INRIA-Lorraine which was invaluable for the completion of this research.

References

[BK89] K. Bundgen and W. Küchlin. Computing ground reducibility and inductively complete positions. In N. Dershowitz, editor, *Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, Lecture Notes in Computer Science, pages 59–75. Springer-Verlag, 1989.

[BR93] A. Bouhoula and M. Rusinowitch. Automatic case analysis in proof by induction. In Ruzena Bajcsy, editor, *Proceedings 13th International Joint Conference on Artificial Intelligence, Chambéry (France)*, volume 1, pages 88–94. Morgan Kaufmann, August 1993.

[CJ97] H. Comon and F. Jacquemard. Ground reducibility is exptime-complete. In *Proceedings of 12th IEEE Symposium on Logic in Computer Science, Warsaw (Poland)*. IEEE Computer Society Press, 1997.

[Com91] H. Comon. Disunification: a survey. In Jean-Louis Lassez and G. Plotkin, editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 9, pages 322–359. MIT Press, Cambridge (MA, USA), 1991.

[DJ90] N. Dershowitz and J.-P. Jouannaud. *Handbook of Theoretical Computer Science*, volume B, chapter 6: Rewrite Systems, pages 244–320. Elsevier Science Publishers, 1990.

[FL96] C. Fermuller and A. Leitsch. Hyperresolution and automated model building. *J. of Logic and Computation*, 6(2):173–203, 1996.

[HH94] D. Hofbauer and M. Huber. Linearizing term rewriting systems using test set. *Journal of Symbolic Computations*, 17(1):91–129, 1994.

[JK89] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82:1–33, 1989.

[KLP91] E. Kounalis, D. Lugiez, and L. Pottier. A solution of the complement problem in associative commutative theories. In A. Tarlecki, editor, *16th International Symposium Mathematical Foundation of Computer Sciences*, volume 520 of *Lecture Notes in Computer Science*, pages 287–297. Springer Verlag, 1991.

[KNRZ91] D. Kapur, P. Narendran, D. J. Rosenkrantz, and H. Zhang. Sufficient completeness, ground-reducibility and their complexity. *Acta Informatica*, 28:311–350, 1991.

[KNZ86] D. Kapur, P. Narendran, and H. Zhang. Proof by induction using test sets. In *Proceedings 8th International Conference on Automated Deduction, Oxford (UK)*, volume 230 of *Lecture Notes in Computer Science*, pages 99–117. Springer-Verlag, 1986.

[KR95a] G. Kucherov and M. Rusinowitch. Complexity of testing ground reducibility for linear word rewriting systems with variables. In *Proceedings 4th International Workshop on Conditional and Typed Term Rewriting Systems, Jerusalem (Israel)*, volume 968 of *Lecture Notes in Computer Science*, pages 262–275. Springer Verlag, 1995.

[KR95b] G. Kucherov and M. Rusinowitch. Undecidability of ground reducibility for word rewriting systems with variables. *Information Processing Letters*, 53:209–215, 1995.

[KT95] G. Kucherov and M. Tajine. Decidability of regularity and related properties of ground normal form languages. *Information and Computation*, 118(1):91–100, Apr 1995.

[Kuc88] G. Kucherov. A new quasi-reducibilty testing algorithm and its application to proofs by induction. In J. Grabowski, P. Lescanne, and W. Wechler, editors, *Proceedings of the 1st International Workshop on Algebraic and Logic Programming, Gaussig (GDR)*, volume 343 of *Lecture Notes in Computer Science*, pages 204–213. Springer Verlag, 1988.

[LM87] J-L. Lassez and K. Marriot. Explicit representation of terms defined by counter examples. *Journal of Automated Reasoning*, 3(3):301–318, 1987.

[LM93] D. Lugiez and J-L. Moysset. Complement problems and tree automata in AC-like theories. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *Proceedings STACS 93*, volume 665 of *Lecture Notes in Computer Science*, pages 515–524. Springer Verlag, Feb 1993.

[Pla85] D. Plaisted. Semantic confluence and completion method. *Information and Control*, 65(2/3):182–215, May 1985.