On the Complexity of Deciding the Derivation Length in Term Rewriting Systems

Michele Flammini, Paola Inverardi, Domenico Mango and Monica Nesi

Dipartimento di Informatica, Università di L'Aquila
via Vetoio, loc. Coppito, I-67100 L'Aquila Italy

email: {flammini,inverard,mango,monica}@di.univaq.it

Verification techniques based on rewriting can play an active role in many fields of software system analysis. To this respect, rewriting based tools can become complementary to model-checking or finite state based tools. However, in order to be practical these techniques have to exhibit a reasonable degree of efficiency, therefore, according to [5], a systematic study of the *quantitative* aspects of term rewriting is in order.

Complexity issues on term rewriting systems have been variously studied in the literature. The time complexity of determining the derivation length in a term rewriting system (TRS for short) is a well-known and investigated issue. From a computational complexity point of view, in general this problem is intractable, since termination is even undecidable [6]. A series of papers have then shown that undecidability also holds if the TRS contains only three rules [8], only two rules [3], one rule [1] and one rule which is left-linear, non-overlapping and variable preserving [2].

In this paper we investigate a numerical refinement of the termination problem, from now on referred to as *Min-DL*, in which, given a term t, a TRS R and a numerical bound k, we are asked to determine if there exists a derivation of t to normal form of length at most k. This problem originated from the need of looking for criteria to suitably bound the length of a terminating derivation in rewriting based tools for the analysis of software systems. In particular, we got interested in this problem when dealing with a rewriting strategy [7] that simulates the completion process of an equational theory in a bottom-up manner. In that case, given a term, it is important to give a *measure* of the search space originated by that term, when rewritten in all possible ways through terminating derivations.

Like the classical proofs of undecidability of termination, some of our negative results are accomplished by simulating Turing machines, although there are several differences between NP-hardness and undecidability simulations. First, we can restrict to Turing machines whose running time is bounded by a suitable polynomial function evaluated on the input size. This allows us to get simpler simulations since we can bound the portion of tape scanned during the computation. On the other hand, we are forced to use non-deterministic Turing machines and, under the restrictions imposed on the structure of the considered TRS, we must guarantee that the length of the computations in the Turing machines and of the corresponding maximal derivations in the TRS are polynomially related. In fact, even if non-determinism does not influence the decidability of a problem, if $P \neq NP$ it affects the tractability in terms of polynomial solvability (see [4] for examples of polynomial simulations).

The problem *Min-DL* is NP-hard, as it does not even belong to the class NP. In fact, k can be exponential in $|t|+|R|$ and derivations of polynomial length can generate terms of exponential size with respect to $|t|+|R|$. Therefore, if polynomial time solvability is concerned, suitable restrictions on the set of allowable instances must be defined.

Definition 1. *Given a term t, a TRS R and a numerical bound k, the triple $\langle t,R,k\rangle$ is p-feasible for a given polynomial p if the following conditions are satisfied:*

1. $k \leq p(|t|+|R|)$;
2. *if* $t \xrightarrow{j} t'$, *then* $|t'| \leq p(|t|+|R|+j)$.

Hence, if $\langle t,R,k\rangle$ is a p-feasible triple, then k and the size of any term derived from t in at most k steps are polynomially bounded (according to p) in the size of t and R.

Given any polynomial p, let *Min-DL*$_p$ be the restriction of *Min-DL* on the set of the instances such that $\langle t,R,k\rangle$ is a p-feasible triple. Then clearly *Min-DL*$_p$ is in NP. In fact, by the p-feasible property, every derivation of length at most k can be described in polynomial space with respect to the size of t and R, and it is possible to check in polynomial time if it is a derivation in R from t to normal form of at most k steps.

Unfortunately, *Min-DL*$_p$ is NP-complete in most cases, even under significant restrictions on the structure of the TRS. In fact, following the line of the results on the indecidability of termination, if one is interested in having a low number of rules the following theorem can be proved.

Theorem 1. *The* Min-DL$_p$ *problem is NP-complete for a suitable polynomial p under the restriction that the TRS R has only one rule.*

The proof is inspired to Dauchet's ideas in [1, 2], namely we provide a general transformation from a non-deterministic Turing machine NT. More precisely, we give a TRS R and a term t such that NT has an accepting computation on a given input x if and only if there is a derivation from t to normal form of at most k steps. By exploiting similar arguments, it is possible to prove the following theorem.

Theorem 2. *The* Min-DL$_p$ *problem is NP-complete for a suitable polynomial p under the restriction that the term t has a unique normal form and the TRS R has only two rules.*

Even if we do not claim it explicitly, the same negative results also hold for the analogous *Max-DL* decision problem in which we are asked if there is a derivation of t to normal form having length more than k.

In conclusion, we have shown that even the non-determinism induced by a single rewriting rule is sufficient to make the problem intractable. Similar negative results also hold if we consider the maximum derivation length.

Many questions remain open. First of all, are the two problems polynomially solvable when the TRS has only one rule and the term a unique normal form?

All the TRSs in the transformations given in the proofs of the above theorems contain overlapping or self-overlapping rules: what about non-overlapping TRSs? As it can be easily checked, under this assumption, if the TRS is right-linear or more in general in each rule the number of occurrences of each variable in the left-hand side is at least equal to the number of occurrences in the right-hand side, a minimum or maximum length derivation can be determined in polynomial time. In fact, if there is a term t that matches the left-hand side of a rule r_1 and a subterm t' of t that matches a rule r_2, since the TRS is non-overlapping, t' is a subterm of another subterm t'' of t that in the application of r_1 matches a variable of its left-hand side. Then, since the number occurrences of t'' cannot be increased by applying r_1, if we apply r_2 after r_1 we cannot get a longer derivation to normal form. On the contrary, if in each rule of the TRS the number of occurrences of each variable in the left-hand side is at most equal to the number of occurrences in the right-hand side, if we apply r_1 after r_2 we cannot get a longer derivation to normal form.

Although the results shown in the paper exhibit a high negative valence, we believe there can still be room for searching TRS characterizations (beyond the ones described above) which permit to suitably constrain the *bad* non-determinism in a TRS and allow polynomial solvability.

References

1. M. Dauchet. Termination of rewriting is undecidable in the one-rule case. In *13th Symposium on Mathematical Foundation of Computer Science (MFCS)*, volume 324 of *Lecture Notes in Computer Science*, pages 262–270. Springer-Verlag, 1988.
2. M. Dauchet. Simulation of Turing machines by a left-linear rewrite rule. In *3rd International Conference on Rewriting Techniques and Applications (RTA)*, volume 355 of *Lecture Notes in Computer Science*, pages 109–120. Springer-Verlag, 1989.
3. N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3:69–116, 1987.
4. M.R. Garey and D.S. Johnson. *Computers and Intractability*. W.H. Freeman, 1979.
5. D. Hofbauer and C. Lautemann. Termination proofs and the length of derivations (Preliminary version). In *3rd International Conference on Rewriting Techniques and Applications (RTA)*, volume 355 of *Lecture Notes in Computer Science*, pages 167–177. Springer-Verlag, 1989.
6. G. Huet and D.S. Lankfork. On the uniform halting problem for term rewriting systems. Technical Report 282, INRIA, Le Chesnay, France, 1978.
7. P. Inverardi and M. Nesi. A Strategy to Deal with Divergent Rewrite Systems. In *Proceedings of CTRS '92*, volume 656 of *Lecture Notes in Computer Science*, pages 458–467. Springer-Verlag, 1992.
8. R. Lipton and L. Snyder. On the halting of tree replacement systems. In *Proc. of the Conference on Theoretical Computer Science*, pages 43–46, Univ. of Waterloo, Waterloo, Canada, 1977.