Primitive Rewriting*

Nachum Dershowitz

School of Computer Science, Tel Aviv University,
Ramat Aviv 69978, Israel
nachum.dershowitz@cs.tau.ac.il

For Jan Willem with admiration

Abstract. Undecidability results in rewriting have usually been proved by reduction from undecidable problems of Turing machines or, more recently, from Post's Correspondence Problem. Another natural candidate for proofs regarding term rewriting is Recursion Theory, a direction we promote in this contribution.

We present some undecidability results for "primitive" term rewriting systems, which encode primitive-recursive definitions, in the manner suggested by Klop. We also reprove some undecidability results for orthogonal and non-orthogonal rewriting by applying standard results in recursion theory.

1 Introduction

Indeed, if general recursive function
is the formal equivalent of effective calculability,
its formulation may play a role
in the history of combinatory mathematics
second only to that of the formulation of natural number.

— Emil Post (1944)

A number of models of computation vie for the rôle of "most basic" mechanism for defining effective computations. These include: semi-Thue systems, Markov's normal algorithms, Church's lambda calculus, Schönfinkel's combinatory logic, Turing's "logical computing" machines, and Gödel's recursive functions. Although they operate over different domains (strings, terms, numbers), they are all of equivalent computational power.[1]

First-order term rewriting makes for a very natural symbolic programming paradigm based on subterm replacement, without bound variables or built-in operations. The two most basic properties a rewrite system may possess are termination (a.k.a. strong normalization) and confluence (the famous Church-Rosser property). Variations on these include (weak) normalization, unique normal forms, and ground confluence. For a comprehensive text on rewriting, see the recent volume by the "Terese" group, based in Amsterdam [38].

* This research was supported by the Israel Science Foundation (grant no. 250/05).

[1] See [4, 5] for a discussion of problems pertaining to comparisons of computational power of models operating over diverse domains.

A. Middeldorp et al. (Eds.): Processes... (Klop Festschrift), LNCS 3838, pp. 127–147, 2005.

It comes as no surprise that rewrite systems have the same computational power as the other basic models.[2] Moreover, rewrite systems may be restricted in various ways, including left-linearity, orthogonality, and constructor-basedness, without weakening the model from the point of view of computability.

To quote Klop [31–p. 356]: "As is to be expected, most of the properties of TRSs [term rewriting systems] are undecidable. Consider only TRSs with finite signature and finitely many reduction rules. Then it is undecidable whether confluence holds, and also whether termination holds." Early undecidability results in string and term rewriting were proved by reduction from undecidable problems of Turing machines (e.g. [9, 25]). More recently, Post's Correspondence Problem [47] has been used: for string rewriting by Book [6]; for term rewriting in [28, 36] (see also [14, 15] and [38–Sect. 5.3.3]). The most natural candidate for proofs regarding term rewriting, however, is recursion theory, a direction we promote here.

Recursive function theory is a uniquely suitable candidate for demonstrating, by means of suitable reductions, that various properties of members of classes of rewrite systems are undecidable. Standard works on recursion theory include [41, 45, 48, 53]. The encoding of recursive functions as term-rewriting systems is part of the field's age-old "folklore", and is mentioned by Klop as an exercise in his 1992 survey [32].

This paper present some undecidability results for "primitive" term rewriting systems, which encode primitive-recursive definitions. Primitive rewriting is defined in Sect. 3. Section 4 shows how to also faithfully encode partial-recursive functions. Kleene's computation predicate—which is central to the undecidability results—is coded as a primitive rewrite system in the Appendix, and its properties are discussed in Sect. 5. In Sects. 6 and 7, we reprove (and improve) some undecidability results for orthogonal and non-orthogonal rewriting (see [38–Chap. 5]) by applying standard results in recursion theory. The concluding section lists what we believe to be new by way of sufficient conditions for undecidability obtained in this way.

2 Background

A total function f over the natural numbers is *primitive recursive* if it is the constant $\lambda.0$, a projection function $\lambda x_1, \ldots, x_k.x_i$, the successor function $\lambda x.x{+}1$, the composition of other primitive-recursive functions, or else is itself definable by primitive recursion of the form:

$$f(n, \ldots, x_i, \ldots) \quad := \quad \begin{cases} g(\ldots, x_i, \ldots) & n = 0 \\ h(f(n-1, \ldots, x_i, \ldots), n-1, \ldots, x_i, \ldots) & \text{otherwise} \end{cases},$$

where g and h are already known to be primitive recursive.

A partial function f over the natural numbers is *partial recursive* if it is primitive recursive, or if it can be defined by composition or primitive recursion

[2] Of course, the classical Church-Turing Thesis asserts that these sets of functions are exactly what are mechanistically computable. See [3].