Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Electronic Notes in Theoretical Computer Science 127 (2005) 71–86
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Termination of High-Level Replacement Units with Application to Model Transformation*

Paolo Bottoni[a,1] Manuel Koch[b,2] Francesco Parisi-Presicce[a,d,3]
Gabriele Taentzer[c,4]

[a] *Università di Roma "La Sapienza" - Italy*
[b] *Freie Universität Berlin - Germany*
[c] *Technische Universität Berlin - Germany*
[d] *George Mason University - USA*

Abstract

Visual rewriting techniques, in particular graph transformations, are increasingly used to model transformations of systems specified through diagrammatic sentences. Several rewriting models have been proposed, differing in the expressivity of the types of rules and in the complexity of the rewriting mechanism; yet basic results concerning the formal properties of these models are still missing for many of them. In this paper, we propose a contribution towards solving the termination problem for rewriting systems with external control mechanisms. In particular, we obtain results of more general validity by extending the concept of transformation unit to high-level replacement systems, a generalization of graph transformation systems. For high-level replacement units, we state and prove several abstract properties based on termination criteria. Then, we instantiate the high-level replacement systems by attributed graph transformation systems and present concrete termination criteria. These are used to show the termination of some replacement units needed to express model transformations as a consequence of software refactoring.

Keywords: Transformation units, graph transformation, termination, refactoring.

* Research partially supported by the European Community under RTN Segravis
[1] Email: bottoni@di.uniroma1.it
[2] Email: mkoch@inf.fu-berlin.de
[3] Email: fparisi@ise.gmu.edu
[4] Email: gabi@cs.tu-berlin.de

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2004.08.048

1 Introduction

Visual rewriting techniques are increasingly used to model transformations of systems specified through diagrammatic sentences. Researchers are moving from the specification of static aspects of languages (defined through parsing processes) to the modelling of their dynamics. Graph transformations, in particular, are a widespread formalism with applications to parsing, model animation or transformation. Moreover, a whole new wealth of problems, such as software or model evolution [17,15,3,14] arises from the diffusion of UML as a tool for the specification of both software and general systems.

When specifying such transformations, it is hardly the case that a single, unstructured, diagram rewriting system is used to define complex transformations. A typical problem is to steer the progress of the transformation towards some well-defined configuration of the diagram, i.e. state of the system. This may involve the definition of some sequence of rule applications, as well as the prevention of repeated application of a same rule to the same match, or of cyclic repetitions of the same sequence of applications.

In general, guaranteeing such properties of the rewriting process is equivalent to proving its termination, an undecidable problem in its uniform version [16], but which can be studied for individual rewriting systems, following the classical approach of proving termination by constructing a monotone measure function on some multiset, and showing that the value of such a function decreases at each application, as introduced by Dershowitz and Manna in [6].

This problem is further complicated by the need for rule expressivity. Indeed, there is always a trade-off between the inherent expressivity (and complexity) of the rewriting relation for a single step and the availability of external control mechanisms steering the rewriting process. Although a number of rewriting models have been proposed, differing in the expressivity of the types of rules and in the complexity of the rewriting mechanism, basic results concerning the formal properties of these models are still missing for many of them. The combination of attributed rules and transformation units employing rule expressions seems to provide a transformation approach which can already be used practically, but which is simple enough for formal reasoning.

In this paper we first study a more abstract version of attributed rules, namely high-level replacement systems [9], to which we extend the notion of transformation unit [13]. We thus obtain an abstract property that a function has to satisfy in order to be used as a termination criterion for such units. In particular, Section 2 introduces related work, while Section 3 adapts the concept of transformation units to define high-level replacement units. In Section 4, a motivating example from software transformation, namely refactoring, is