# *Tutorial on Analysis and Verification of Imperative Programs through CLP*[*]

John Gallagher

*Roskilde University, Denmark and IMDEA Software Institute, Spain*
(*e-mail:* `jpg@ruc.dk`)

## Abstract

In this tutorial we show how constraint logic programs (CLP) provide a flexible framework for analysis and verification of other languages. Here the focus is on analysing imperative programming languages. It is first necessary to translate a given imperative program into CLP clauses. Different approaches to automatic translation based on small-step or big-step semantics will be shown, along with their advantages and disadvantages. The role of query-answer transforms in the translation process is also presented. The problem of analysing or verifying properties of an imperative program is thus translated into a CLP analysis or verification problem. The main CLP analysis technique covered in this tutorial is the computation of approximate models of CLP clauses using abstract interpretation over numeric or symbolic abstract domains. The tutorial contains a survey of analysis and verification problems for imperative programs that have been successfully tackled in this way, including verification of safety properties in sequential and concurrent programs, termination, resource analysis and shape analysis.

[*] Some of the contents of this tutorial resulted from support from EU FP7 project 318337, *ENTRA - Whole-Systems Energy Transparency* and Danish Research Council grant FNU-10-084290.