Program termination: from well-founded orderings to logical models and back*

Salvador Lucas

DSIC, Universitat Politècnica de València, Spain
http://users.dsic.upv.es/~slucas/

Abstract. Termination is a classical problem in Computer Science, Programming, and Software Engineering. In the late 40's, Alan Turing considered termination analysis and proposed the idea of *mapping computations* into *decreasing quantities* which "*vanish when the machine stops*" in order to *verify* program termination. In the 60's, Floyd championed the use of *well-ordered sets* (or *well-founded orderings*) rather than numeric quantities; and Manna modeled computations as logic formulas and drew a connection between termination and the absence of *logical models* for such formulas. Indeed, disregarding programming languages and paradigms, computations are often viewed as *proofs* of specific sentences in a *computational logic* describing the operational semantics of the programming language or computational system. And in *declarative programming*, programs *are* theories of a logic. Thus, logic provides a unifying framework to reason about programs, also regarding termination. We discuss the analysis of program termination with a special focus on declarative languages, also considering implementation issues as required in the development of tools for automatically proving program termination. In this setting, logic and logic-based techniques are essential at different levels; and *well-foundedness* often recasts into a side requirement for the logical models at stake.

Keywords: Abstraction, Logical models, Program analysis, Termination.

* Partially supported by the EU (FEDER), Spanish MINECO projects TIN 2013-45732-C4-1-P and TIN2015-69175-C4-1-R, and GV project PROMETEOII/2015/013.