

MASTER THESIS: OVERAPPROXIMATING THE COST OF WHILE, FOR, AND FOREACH LOOPS

INTRODUCTION

TouchDevelop (<http://research.microsoft.com/en-us/projects/touchdevelop/>) is a novel programming language introduced by Microsoft to develop scripts on mobile devices (e.g., smartphones). Its language is particularly simple, since it has to allow one to develop applications on mobile devices with limited screen and input devices.

Nevertheless, it provides three different forms of loops: i) `while`, ii) `for (a<=v<=b)` which iterates from `a` to `b`, and iii) `foreach(x in Collection)` which iterates over all elements of a collection.

Because of the limited hardware resources of mobile devices, a cost analysis of a TouchDevelop script could provide some useful information to optimize its execution. In particular, we could attach cost information to the script and use this information at runtime to decide if the application should be executed on the mobile device, or in the cloud. Developing a precise overapproximation of the cost of a loop is a complex problem that has been (partially) explored in the last few years.

During the last three years, at the Chair of Programming Methodology we have developed the static analyzer `Sample`, which is based on the abstract interpretation theory. This analyzer has been already applied to a wide range of properties (for instance, types, string values, integers, heap structures) and to different programming languages (Scala and Java bytecode). Therefore, it seems flexible and generic enough to be applied to the cost analysis of TouchDevelop programs. The representation of a program in `Sample` relies on control flow graphs, and at this level the information about the specific type of loop (namely, `while`, `for`, and `foreach`) represented by the control flow graph is already abstracted away.

GOAL

The goal of this master thesis is to develop a cost analysis that overapproximates the cost of a loop, that is, the number of times a loop is iterated. Since the representation of a program adopted by `Sample` is based on control flow graphs, we cannot distinguish between `while`, `for`, and `foreach` loops, and we have to treat them in a uniform way. The first step will be to precisely overapproximate the cost of loops obtained from `for` and `foreach` statements. We expect that the treatment of loops obtained from these constructs will be easier than the ones obtained from `while` loops, since they iterate in a more predictable way. In particular, we will study the patterns of the `cfg` structure and of the statements contained in the loop body to identify the parts of the `cfg` obtained compiling `for` or `foreach` statements. In this way, we will be able to precisely compute the costs of these parts of code. The second step of the thesis will be the extension of this approach to loops obtained from arbitrary `while` statements. We expect that we will be able to precisely abstract the cost of only some of these loops, while in the other cases the analysis will not yield a result. This cannot be avoided since we cannot precisely compute the cost of *any* loops statically. The whole approach will be implemented in `Sample` to study the results of the proposed analysis in practice.

Note that this project is focused on loops. Therefore, the treatment of other statements is outside the scope of this master thesis, and it will be studied separately. In the context of this thesis, we will assume to know the cost of the body of a loop, focusing the work only on the statements that affect the number of iterations of the loop.

This thesis is in the context of the project “TouchBoost – Cloud-based Static Analyses to Improve Lay Programming on Mobile Devices” funded by Microsoft.

RECOMMENDED BACKGROUND

- Formal methods
- Static program analysis