Chapter 36
Using Kolmogorov Complexity to Study the Coevolution of Header Files and Source Files of C-Alike Programs

Liguo Yu
Indiana University South Bend, Computer Science Department, South Bend, USA

ABSTRACT

In C-alike programs, the source code is separated into header files and source files. During the software evolution process, both these two kinds of files need to adapt to changing requirement and changing environment. This paper studies the coevolution of header files and source files of C-alike programs. Using normalized compression distance that is derived from Kolmogorov complexity, we measure the header file difference and source file difference between versions of an evolving software product. Header files distance and source files distance are compared to understand their difference in pace of evolution. Mantel tests are performed to investigate the correlation of header file evolution and source file evolution. The study is performed on the source code of Apache HTTP web server.

INTRODUCTION

In C, C++, or Objective-C programs, there are two types of code files: header files and source files. Header files contain the declarations of classes, functions, global variables, and identifiers. Source files contain the implementations of classes and functions and the usages of variables and identifiers. The purpose of this organization is to separate program structure and interface from specific implementations (Stroustrup, 2013). On the other hand, software evolution is inevitable. Software products need to adapt to new requirements and new environments (Bendifallah, 1987; Gall et al., 1997; Kemerer & Slaughter, 1999; Lehman, 1980; Yu & Ramaswamy, 2009b; Yu & Ramaswamy, 2006). In this evolution process, both the program structure (interface) and the program implementation will need to be changed.

DOI: 10.4018/978-1-7998-3016-0.ch036

Because header files contain the program structure and interface, they are expected to be more robust than source files to requirement changes and environment changes. However, there has been no empirical research to validate this speculation. The objective of this study is to understand and compare the evolution of header files and source files. Using normalized compression distance that is derived from Kolmogorov complexity, we measure the difference of versions of header files and difference of versions of source files. Our study is performed on the source code of Apache HTTP, a web server written in C. Through this study, our objective is to answer the following two questions:

1. In the source code of Apache HTTP, do header files and source files have similar paces of evolution?
2. In the source code of Apache HTTP, is the evolution of header files correlated with the evolution of source files?

To answer these two questions, the remainder of the paper is organized as follows. Section 2 reviews Kolmogorov complexity and normalized compression distance. Section 3 describes our research method and research data. Section 4 presents the results of this study. Conclusions appear in Section 5.

KOLMOGOROV COMPLEXITY AND NORMALIZED COMPRESSION DISTANCE

In algorithmic information theory, the Kolmogorov complexity of an object, such as a string (a piece of text), is the length of the string's shortest description in some fixed universal description language (Li & Vitányi, 2009). The Kolmogorov complexity of a binary string x can be represented by $K(x)$ denoting the shortest length of string x described using a universal language.

Based on Kolmogorov complexity, Bennett et al. (1998) defined information distance between two binary strings x and y as the following.

$$E(x,y) = \max\{K(x|y), K(y|x)\} \tag{1}$$

In Equation 1, $K(x|y)$ denotes the conditional Kolmogorov complexity of string x given string y. Information distance $E(x,y)$ measures the absolute distance between two strings x and y and thus does not reflect the relative difference between the two strings. Li et al. [10] then defined normalized information distance between two binary strings x and y.

$$d(x,y) = \frac{\max\left\{\left(K(x|y), K(y|x)\right\}\right.}{\max\left\{K(x), k(y)\right\}} \tag{2}$$

Equation 2 is based on Kolmogorov complexity, which is non-computable in practice. To make the information distance more practical to use, Li et al. (2004) further substituted Kolmogorov complexity $K(x)$ of a string x with the optimal compressed length $C(x)$ of string x. The normalized compression distance between two binary strings x and y is defined below (Li et al., 2004).