Probabilistic Inference of Data Abstractions in Machine Code

Aditya Thakur and Mark Chapman
May 16, 2011

1 Introduction

Our problem and motivation

2 Overview

Our solution in general terms

3 Related work

Related work

Early work that finds static library code using hash collisions [1].

Introduction to pointers, arrays, and recursive structures in binary [2].

Closest to our technique: pick fingerprints, calculate similarities [3].

Uses similar code region normalization; relies on approximate hashing instead of fingerprints for similarity [4]

Recent survey covering variety of methods to find similar code segments [5]

4 Current status

Current status of our project

5 Future work

Next steps in our project

References

[1] M. V. Emmerik, "Identifying library functions in executable files using patterns," in *Australian Software Engineering Conference (ASWEC '98)*, pp. 90–97, IEEE Computer Society, 1998.

[2] A. Mycroft, "Type-based decompilation (or program reconstruction via type reconstruction)," in *European Symposium on Programming (ESOP '99)* (S. D. Swierstra, ed.), vol. 1576 of *Lecture Notes in Computer Science*, pp. 208–223, Springer, 1999.

[3] R. Smith and S. Horwitz, "Detecting and measuring similarity in code clones," in *International Workshop on Software Clones (IWSC 2009)*, Computer, pp. 28–34, 2009.

[4] A. Sæbjœrnsen, J. Willcock, T. Panas, D. J. Quinlan, and Z. Su, "Detecting code clones in binary executables," in *International Symposium on Software Testing and Analysis (ISSTA 2009)* (G. Rothermel and L. K. Dillon, eds.), pp. 117–128, ACM, 2009.

[5] C. K. Roy, J. R. Cordy, and R. Koschke, "Comparison and evaluation of code clone detection techniques and tools: A qualitative approach," *Science of Computer Programming*, vol. 74, no. 7, pp. 470–495, 2009.