

Dynamic programming languages allow us to write code without type information and types of variables can change during execution. Although easier to use and suitable for fast prototyping, dynamic typing can lead to error prone code and is challenging for the compilers or interpreters. Programmers often use documentation comments to provide the type information, but the correspondence of the documentation and the actual code is usually not checked by the tools. In this thesis, we focus on one of the most popular dynamic programming languages: PHP. We have developed a framework for static analysis of PHP code as a part of the Phalanger project -- the PHP to .NET compiler. The framework supports any kind of analysis, but in particular, we implemented type inference analysis with emphasis on discovery of possible type related errors and mismatches between documentation and the actual code. The implementation was evaluated on real PHP applications and discovered several real errors and documentation mismatches with a good ratio of false positives.