

Chapter 1

Table of contents

- 1 Introduction
 - 1.1 Motivation
 - * 1.1.1 Problem Definition
 - 1.2 Research Questions
 - 1.3 Non-Goals
 - 1.4 Research Methodology
 - 1.5 Structure of the thesis
- 2 Background
 - 2.1 Language Server Protocol
 - * 2.1.1 JSON-RPC
 - * 2.1.2 Commands and Notifications
 - 2.1.2.1 File Notification
 - 2.1.2.2 Hover
 - 2.1.2.3 Completion
 - 2.1.2.4 Go-To-*
 - 2.1.2.5 Symbols
 - 2.1.2.6 code lenses
 - * 2.1.3 Shortcomings
 - 2.2 Configuration programming languages
 - * 2.2.1 Infrastructure as Code
 - * 2.2.2 Nickel
 - 2.2.2.1 Gradual typing
 - 2.2.2.2 Contracts
 - 2.2.2.3 Nickel AST
- 3 Related work
 - 3.1 Language Servers
 - * 3.1.1 Considerable dimensions
 - 3.1.1.1 Language Complexity
 - * 3.1.2 Representative LSP Projects
 - 3.1.2.1 LSP as standard tooling
 - 3.1.2.2 Rapid implementation of LSP features
 - 3.1.2.3 Language Independence
 - 3.1.2.4 Language Servers for Domain Specific Languages

- * 3.1.3 Honorable mentions
 - 3.2 Alternative approaches
 - * 3.2.1 Platform plugins
 - * 3.2.2 Legacy protocols
 - * 3.2.3 LSP Extensions
 - * 3.2.4 LSIF
- 4 Design implementation of NLS
 - 4.1 Illustrative example
 - 4.2 Linearization
 - * 4.2.1 States
 - * 4.2.2 Transfer from AST
 - 4.2.2.1 Usage Graph
 - 4.2.2.2 Scopes
 - 4.2.2.3 Linearizer
 - 4.2.2.4 Linearization Process
 - * 4.2.3 Post-Processing
 - 4.2.3.1 Sorting
 - 4.2.3.2 Resolving deferred access
 - 4.2.3.3 Resolving types
 - * 4.2.4 Resolving Elements
 - 4.2.4.1 Resolving by position
 - 4.2.4.2 Resolving by ID
 - 4.2.4.3 Resolving by scope
 - 4.3 LSP Server
 - * 4.3.1 Diagnostics and Caching
 - * 4.3.2 Commands
 - 4.3.2.1 Hover
 - 4.3.2.2 Jump to Definition and Show references
 - 4.3.2.3 Completion
 - 4.3.2.4 Document Symbols
- 5 Evaluation
 - 5.1 Methods
 - * 5.1.1 Qualitative
 - * 5.1.2 Quantitative
 - 5.2 Process
 - 5.3 Results
 - * 5.3.1 Qualitative
 - * 5.3.2 Quantitative
- 6 Discussion
 - 6.1 Project results
 - 6.2 Project shortcomings
 - 6.3 Future Work