Dynamic Session Types

Student Number: 170027195

June 7, 2018

1 Description

When communicating over a protocol, errors can occur throughout the session which corrupt the state of communication between the parties, this corruption can mean that further communication is impossible, and at worst lead to security vulnerabilites. To navigate around this, in part, session types have been developed - these lay the groundwork for "acceptable" communication over a protocol, allowing for the session to self check to ensure that the state is valid. A type relating to the respective protocol is produced, this is a programmatic representation of the expected inputs and outputs in all cases of the protocol's features, which is then checked against the "real world" implementation of the protocol. Two approaches can be used to perform this analysis; static checking, which checks the state of the communication is valid at compile time and dynamic checking, which performs the analysis at run time as the program is executing. The aim of this project is to produce a working library that implements dynamically checked session types in Java, allowing for the user to define acceptable protocol communication and have the state consistently checked as transaction of messages between parties occurs.

2 Objectives

2.1 Primary

- Produce a literature review which provides a strong introduction to session types - the problems that led to their creation, current examples of their application and language-specific implementation details.
- Produce a library for in, which provides a sound implementation of dynamically checked session types, in a well maintained, documented and extensible manner.
- Produce a number of "showcases", showing the application of the session types library in practice, how they change the execution of the program and what advantages and disadvantages this comes with.
- Implement the HTTP protocol, checked dynamically via the session types library as a test showcase.

2.2 Secondary

- Implement a heavily concurrent program that communicates between processes over a protocol as a test case.
- Integrate the library with modern Java package managers to allow for external users to download and use the library on their machine.
- Compare the performance of dynamic session type implementations to statically checked versions of the same program.
- Allow for the users to access the functionality of the library through custom Java annotations (@SessionType(x))

2.3 Tertiary

- Implement the Diffie-Helmann key exchange protocol using session types to ensure that its execution is secure.

3 Ethics

The ethical approval process for my thesis is simple due to its engineering focused nature. The ethical board is largely concerned with the use of human subjects - which my project will not use, and therefore the possibility of physical and psychological harm being caused is a non-factor. There are no potential conflicts of interest involved in the development of the library, and funding is also not a variable that needs considered during the approval process in this instance.

4 Resources

The resources required for the implementation of the session types library are mostly simple in a hardware sense, a modern computer capable of running Java programs will be required. However, from a software perspective there may be a requirement for the computer to have build tools such as Gradle installed, these will be outlined in the project requirements. Valid licenses will be required for the IDE and other development software used, this will likely by IDEA IntelliJ Ultimate edition - which has a special status for academic use to allow for it to be used free of charge.