1. Motivation

The problem of checking language inclusion/equivalence of Buchi automata is a fundamental problem in formal language theory and has important applications in various fields such as software verification, model checking, and synthesis. The RABIT tool is currently the best performing tool for this problem, but its parallelization is incomplete. In this project, we aim to improve the performance of the RABIT tool by re-implementing some of its core algorithms in a concurrent way.

The motivation behind this project is twofold. Firstly, parallel computing has become increasingly important due to the proliferation of multi-core processors and the need for faster and more efficient computation. Secondly, the problem of language inclusion/equivalence of Buchi automata is computationally intensive and can take a long time to solve, especially for large automata. Therefore, parallelization can significantly improve the performance of the RABIT tool, making it more efficient and scalable.

This project is of both academic and industrial significance. Academically, it will advance the state-of-the-art in the area of formal language theory and concurrency. Industrially, it will have practical applications in software verification, model checking, and synthesis, where the performance of the RABIT tool can directly impact the efficiency and effectiveness of these applications.

1.1 Problem Statement

The problem we are addressing is the incomplete parallelization of the RABIT tool for checking language inclusion/equivalence of Buchi automata. While some of the core algorithms in the tool have parallel versions, others do not. This limits the performance and scalability of the tool, especially for large automata. Therefore, we aim to re-implement some of the core algorithms in a concurrent way to improve the performance and scalability of the tool.

1.2 Research Hypothesis and Objectives

Our research hypothesis is that re-implementing the core algorithms of the RABIT tool in a concurrent way will significantly improve its performance and scalability. Specifically, we conjecture that the parallel version will scale well with the number of available cores and that the optimal size of sub-tasks for single-threaded computation can be determined through empirical evaluation.

The objectives of this project are as follows:

* To re-implement the core algorithms of the RABIT tool in a concurrent way, specifically the BLAFairSimRelNBW and JumpingBLAFairSimRelNBW algorithms.
* To evaluate the performance of the parallel versions of these algorithms on large test cases and compare them against the sequential version.
* To present the results in a clear and concise manner, using tables, graphs, and other visual aids as necessary.
* To determine the optimal size of sub-tasks for single-threaded computation through empirical evaluation.

To contribute to the advancement of the state-of-the-art in the area of formal language theory and concurrency.

* To provide a practical solution for checking language inclusion/equivalence of Buchi automata, which has important applications in software verification, model checking, and synthesis.

In summary, this project aims to improve the performance and scalability of the RABIT tool for checking language inclusion/equivalence of Buchi automata by re-implementing some of its core algorithms in a concurrent way. We believe that this will have important academic and industrial implications and contribute to the advancement of the state-of-the-art in the field of formal language theory and concurrency.