Project Title

CMPUT 664: Final Project Report

FIRST STUDENT, University of Alberta, Canada
SECOND STUDENT, University of Alberta, Canada
THIRD STUDENT, University of Alberta, Canada

1 Introduction

State the problem or research question, explain why it matters, and summarize the project's goal and main contribution. Keep this section brief so that most of the report can focus on the technical work and evaluation.

2 Technical Approach

Explain what you implemented, formalized, extended, replicated, or investigated, and how it works. Introduce the necessary definitions and assumptions, then explain the main design choices, algorithms, encodings, or proof ideas. A small example, diagram, or code fragment can make the explanation more concrete. Clearly distinguish your own contribution from existing tools, libraries, and results. You may rename this section to suit your project.

3 Evaluation

Explain how you assessed the project, present the results, and discuss what they establish. Choose evidence appropriate to your question: experiments, comparisons, case studies, or an assessment of a formalization and its proofs. For experiments, describe the setup, inputs, baselines where relevant, and measurements. For proof-oriented work, explain which properties were verified, under what assumptions, and what remains unproved. Discuss limitations, unsuccessful attempts, and what you learned. The accompanying software artifact must include a README with Unix/Linux setup and run instructions (Docker is acceptable), so that the main experiments or proof checks can be rerun.

4 Related Work

Discuss the most relevant prior work and explain how your project builds on it or differs from it. Compare goals, approaches, assumptions, or results rather than listing paper summaries. Cite the sources of ideas, methods, tools, and results that you use.

For example, a project using Hoare logic could cite its foundational paper [Hoare 1969] and explain how that work relates to the project.

Division of Labor

Briefly describe each team member's specific contributions, such as design, implementation or formalization, evaluation, and writing. Identify shared work where appropriate. This note does not count toward the four-page limit.

References

C. A. R. Hoare. 1969. An Axiomatic Basis for Computer Programming. *Commun. ACM* 12, 10 (1969), 576–580. doi:10.1145/363235.363259