# **COMP60332 Mini-Project**

# Task 1:

The videos in blackboard were studied and understood.

Takeaways from the videos –

1. Backjumping is superior to backtracking, as it reduces the search space by skipping search paths which results in a conflict.
2. Lemma learning is useful to identify the conflict clause, that will prevent conflict-causing search paths.
3. Most state-of-the-art SAT solvers at present use some form of conflict-driven clause learning to reduce search space.

# Task 2:

From the different sources provided for reading, there are several methods of constructing the conflict clause identified. Notably,

1. Tri-asserting clause, where the immediate literals which implied the conflict literal are chosen for the conflict clause.
2. First unique implication point, where the first implication point, that is, the first node from the conflict literal from where all the paths from the decision pass through.
3. Last unique implication point – this constructs a conflict clause using the decision literals chosen which results in the conflict.

In my implementation, I have chosen to use “last unique implication point” for lemma learning.

# Task 3:

The CDCL algorithm is programmed in C++14. Makefile is present in the root directory of the project, which has been compiled and tested on Apple M2, and Intel x86\_64 architectures.

After compiling the program using the makefile, the solver can be run using a single command –

**cat problem.cnf | ./main**

# Task 4:

The program has been evaluated against the test environment provided, with the following output –

1. g-test **simple** ./optdir ../../main

A black background with white text

Description automatically generated

1. g-test **medium** ./optdir ../../main

A black background with numbers and text

Description automatically generated

1. g-test **hard** ./optdir ../../main

A black background with white text and numbers

Description automatically generated

# Task 5:

1. Methods studied:

The main challenge for implementation of the algorithm was the lemma-learning feature, which investigated from the following sources [1, 3, 4, 5], with the most implementation-specific information provided in [5]

1. Implementation

First draft of the algorithm involved using backtracking without lemma learning, which was performing well with “simple” test-set, however it was not performing well with “medium” and “hard” test-sets.

Second draft of the algorithm was implemented with tri-asserting clause method of lemma learning, which was not successful due to implementation shortcomings.

Third version involved using “last unique implication point”, which fared well with “simple” testset and “medium” test-set, however, there were some inconsistencies in the “medium” test set due to implementation issues.

1. The analysis of the test-results is provided under Task 4 heading of this document.

# References

1. <http://www.cs.cmu.edu/~mheule/15816-f20/slides/CDCL.pdf>
2. <http://minisat.se/downloads/MiniSat.pdf>
3. <http://homepage.cs.uiowa.edu/~tinelli/papers/NieOT-JACM-06.pdf>
4. <https://sat.inesc-id.pt/~ines/publications/phd.pdf>
5. <https://www.cs.princeton.edu/courses/archive/fall13/cos402/readings/SAT_learning_clauses.pdf>