CS221 Final Project Proposal

Peter Boennighausen and Mohammed Osman

October 2019

1 Problem Statement

1.1 Input and Output

- **Input**: A propositional formula composed of operations $\wedge, \neg, \vee, \rightarrow$, primitives $\top, \bot$, and particular variables p, q, etc.
- **Output**: A truth table that shows the truth value of the formula given particular inputs ($\top$ or $\bot$) for the variables.

1.2 Scope

We do not propose to create a Boolean satisfiability (SAT) problem solver that would determine if it is possible to make a certain formula true via the assignment of variables for an arbitrary number of variables. Instead, we merely want to generate the truth tables for relatively simple formulae. The existence of a $\top$ within the truth table indicates satisfiability, but our approach will limit ourselves to a reasonable number of variables.

2 Baseline and Oracle

2.1 Baseline

It is relatively difficult to come up with a baseline for generating truth tables, but such a baseline could include:

- the proper sized table, at 2^n for n variables
- a heuristic that assigns the formula the same truth value as the majority of the variables (i. e. true if most of the variables are true else false).

We implemented this baseline algorithm as well as classes for formulae and truth tables.

2.2 Oracle

We can generate labeled data via the CS103 truth table generator, which generates correct truth tables for a given formula.

3 Literature Review

- Deterministic truth table generators such as the CS103 generator exist and some are open source, allowing us to generate labeled data relatively easily
- Our review did not surface any publications that match our specific problem. However, SAT solvers offer interesting parallels for interpreting propositional formulae. Examples include Selsam et al. 2019, which uses a neural network to predict satisfiability of a particular problem, and the DPLL algorithm, which uses backtracking to determine satisfiability.

4 Next Steps

We need to generate data, which can be done relatively simply by stringing together variables and operators. In addition, we will set up a determinstic oracle that either draws from the CS103 truth table generator or provides its own approach.