ABSTRACT

In recent decades, we have seen a number of satisfiability solvers emerge in the world of fuzzy propositional logics. However, among the three fundamental fuzzy logics—Łukasiewicz, Gödel, and product—only a few of them can solve problems with product semantics. Existing solvers may be categorized into those based on (1) translations into instances of other systems, such as satisfiability modulo theories, (2) black-box optimization algorithms, and (3) fuzzy generalizations of classical-logic procedures, such as hyperresolution or the Davis-Putnam-Logemann-Loveland (DPLL) procedure.

This thesis is focused on solving satisfiability (SAT), validity (VAL), and deducibility (DED) of finite propositional theories in product fuzzy logic extended with the Monteiro-Baaz Δ connective, the equality connective $=$, and the strict order connective $\prec$. We have proposed a deterministic algorithm performing a product-logic generalization of the DPLL procedure. The approach has been extended to also treat intermediate constants (constants between $\bar{0}$ and $\bar{1}$ representing partial truth values). In conjunction, we have proposed an algorithm for finding models of satisfiable theories.

These algorithms have been implemented in our solver prodfsat. To the best of our belief, this is the first publicly available solver capable of solving the SAT, VAL, and DED problems in Δ-extended propositional product logic (Π_Δ) that falls within the third category, i. e., is standalone instead of being based on translations or stochastic black-box optimization methods. We have conducted empirical tests and evaluated our solver on a series of experiments, comparing it to the mature SMT-based solver mNiBLoS, which we consider to be the state of the art solver supporting Π_Δ. The results of evaluation confirm the qualities of our solver, being superior in runtime at some of the experiments and inferior at others. One of these qualities is its self-contained nature, which allows one to easily extend it with heuristics that may improve its performance in general, or in specific scenarios given knowledge about the problem to be solved. With this in mind, we have demonstrated the extensibility of our solver by devising reduction rules that help constrain the solution search space, and empirically confirmed the improvement.