Having dealt with Boolean combinations of linear arithmetic constraints in the previous chapter, we now address the problem of solving Boolean combinations of nonlinear arithmetic constraints which may contain transcendental functions, like sine, cosine, and the exponential function. This gives rise to a plethora of problems, in particular (a) how to efficiently and sufficiently completely solve conjunctive combinations of constraints in the undecidable domain of nonlinear constraints involving transcendental functions and (b) how to efficiently maneuver the large search spaces arising from the potentially rich Boolean structure of the overall formula.

While promising solutions for these two individual sub-problems exist, it seems that their combination has hardly been attacked. Arithmetic constraint solving based on interval constraint propagation (e.g., [41, 17]), on the one hand, has proven to be an efficient means for solving conjunctions of nonlinear (in particular transcendental) constraints, provided the latter are robust\(^1\) in the sense that their truth value does not change under small perturbations of the occurring constants [104]. Modern SAT solvers, on the other hand, can efficiently find satisfying valuations of very large propositional formulae (cf. chapter 3), and, using the DPLL(\(T\)) framework, of complex propositional combinations of atoms from various decidable theories (cf. chapter 4).

In this chapter, we describe a tight integration of SAT-based proof search with interval-based arithmetic constraint propagation, thus providing an algorithm that reasons over the undecidable formula class of Boolean combinations of nonlinear constraints involving transcendental functions. Undecidability of the arithmetic base theory, however, precludes a DPLL(\(T\))-style integration. Instead, we exploit the algorithmic similarities between DPLL-based propositional SAT solving and constraint solving based on constraint propagation for a much tighter integration, where the

\(^1\)According to [104], robustness is a property which characterizes ‘exactly the problems that model real-life problems in a meaningful way’.
DPLL solver directly manipulates theory atoms instead of a propositional abstraction of the input formula. It has full introspection into and control over constraint propagation within the theory $T$, and it directly integrates any new theory atoms generated by the constraint propagation into the search space of the DPLL solver. This tight integration has a number of advantages. First, by sharing the common core of the search algorithms between the propositional and the theory-related, interval-constraint-propagation-based part of the solver, we are able to transfer algorithmic enhancements from one domain to the other: in particular, we thus equip interval-based constraint solving with all the algorithmic enhancements that were instrumental to the enormous performance gains recently achieved in propositional SAT solving, like watched-literal schemes or conflict-driven learning based on implication-graph analysis. Second, the introspection into the constraint propagation process allows fine-granular control over the necessarily incomplete arithmetic deduction process, thus enabling a stringent extension of SMT to an undecidable theory. Finally, due to the availability of learning, we are able to implement an almost lossless restart mechanism within an interval-based arithmetic constraint propagation framework.

In fact, our integration of DPLL and interval-based constraint solving gives a clean generalization of the DPLL routine to Boolean combinations of arithmetic constraints, which we refer to as ISAT algorithm. ISAT inherits the branch-and-deduce framework with all its recent enhancements from DPLL. For Boolean deduction it uses the unit-propagation rule from DPLL and adds deduction rules for arithmetic operators taken from interval constraint solving. Due to the undecidability of the arithmetic domain, ISAT is necessarily incomplete. This incompleteness manifests itself in the fact that, strictly speaking, only UNSAT results delivered by ISAT are reliable. In practice, however, this turns out to be no serious restriction, because if unsatisfiability cannot be established, then ISAT returns a small box in the search space which it cannot prune any further by deduction and which therefore is likely to either contain genuine solutions or at least *almost-solutions* which violate arithmetic constraints at most by a very small amount.

**Related Work**

In constrast to solvers adopting the DPLL($T$) approach to SMT, our algorithm does not feature the typical two-layered architecture of such tools, which consists of a DPLL-based SAT solver and a subordinated theory solver. Instead, both engines are inseparably interwoven in our algorithm and by this constitute a generalized form of the DPLL routine itself.