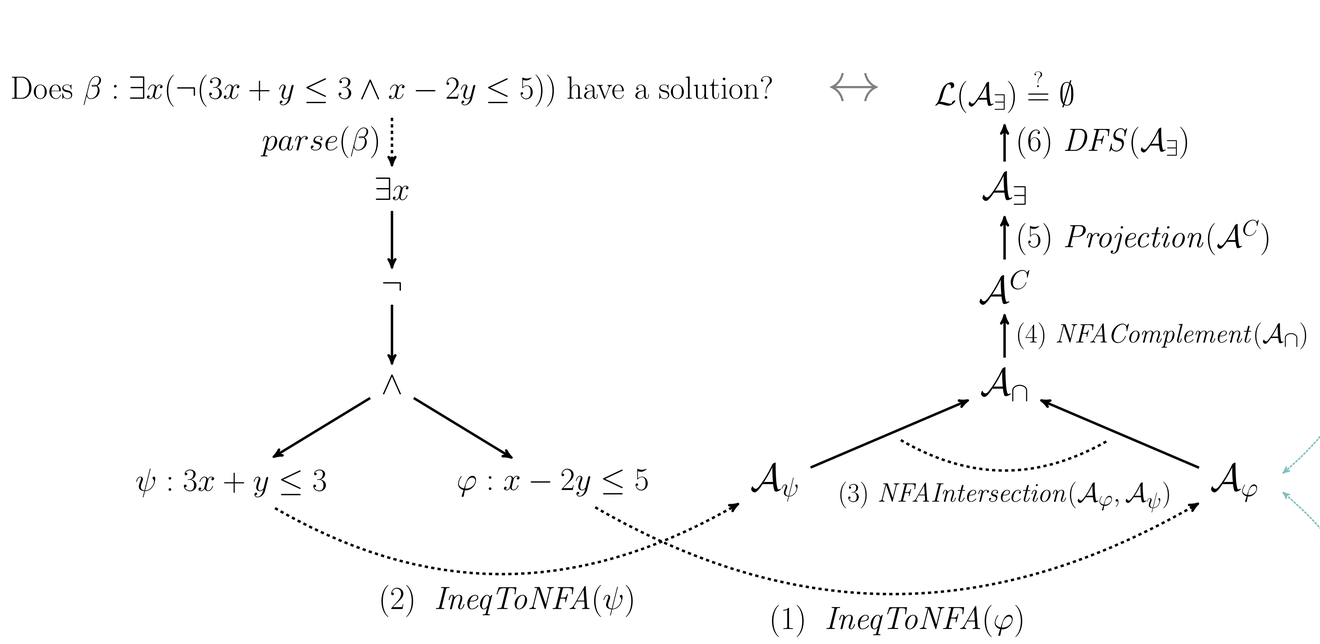
An Automata-Based Decision Procedure for Presburger Arithmetic

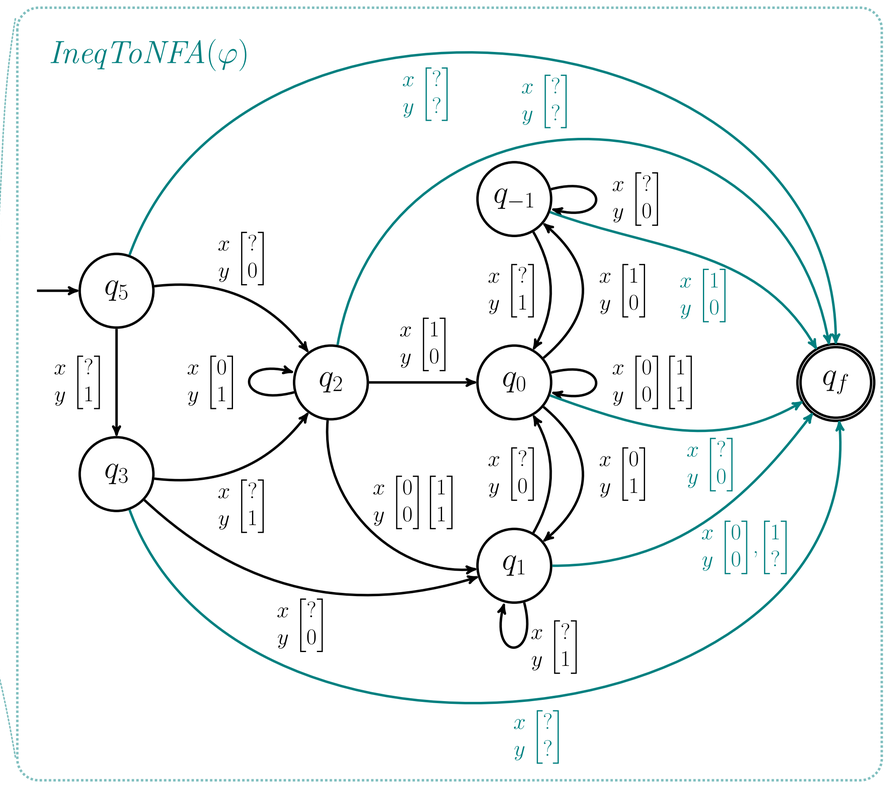
#16 Michal Hečko (xhecko02@stud.fit.vutbr.cz)

A decision procedure for Presburger arithmetics

Presburger arithmetics. Presburger arithmetic (PrA) is a decidable, first-order theory of the structure $(\mathbb{Z}, 0, 1, +, \leq)$. PrA allows describing a system using linear integer constraints (equations, inequations, congruences with a constant modulo). The decidability of PrA makes it possible to implement automatic tools — Satisfiability Modulo Theories (SMT) solvers — capable of deducing whether a given formula has a model.

Automata-based decision procedure. The idea of using finite automata as a basis for a decision procedure — an algorithm determining whether a formula has a model — dates back to the works of Büchi in 1960. This younger alternative to the well-known quantifier elimination works by constructing automata accepting solutions of atomic constraints and then combining these automata according to the structure of the input formula, mapping logical connectives to operations on automata.

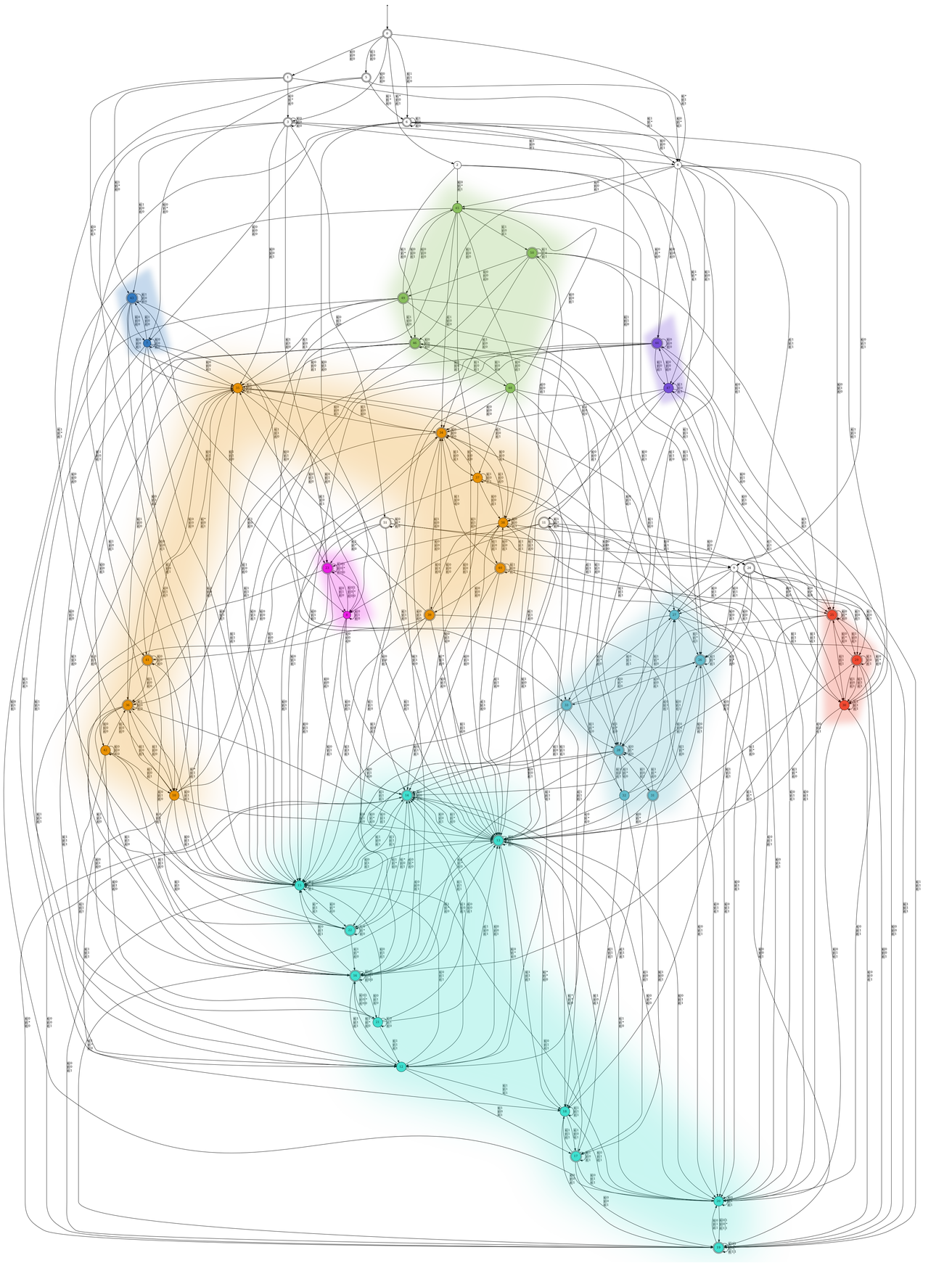




Amaya

Amaya is a novel, experimental SMT solver for PrA engineered from scratch implementing an automata-based decision procedure — an approach no other SMT solver employs. It presents a foundation for future research on the practical applications of automata in the context of deciding PrA. Amaya supports a subset of SMT-LIB — a standardized input language allowing easy comparison to the state-of-the-art solvers, allowing identification of areas where automata offer a more performant solution to the approaches used by the state-of-the-art-solvers. Designed with experimentation in mind, Amaya provides full introspection into the decision procedure by exporting intermediate automata in various formats (e.g. the DOT language). Our solver provides two execution backends the user can choose from: a native one representing transition symbols explicitly, enabling easy experimentation, and an optimized, performance-oriented backend using Multi-Terminal Binary Decision Diagrams (MTBDDs) to represent automaton transition relations, addressing the exponential time complexity of the classical automata algorithms wrt. the number of variables used in the input formula.

Example of an intermediate automaton constructed during the automata-based decision procedure



Besting the state-of-the-art on deciding the Frobenius coin problem

Frobenius coin problem is a famous mathematical problem that can be formulated as follows: Given coins of certain denominations, what is the highest possible amount that cannot be obtained? The precise formulation is given by the following formula where $\vec{w}$ is the vector of coin denominations, m is the number of coins and f is the solution:

$$Frob(f) \colon \forall \vec{n} \in \mathbb{N}^m (f \neq \vec{w} \cdot \vec{n}) \ \wedge (\forall f' \in \mathbb{N}[(\forall \vec{n'} \in \mathbb{N}^m (f' \neq \vec{n'} \cdot \vec{w})) \rightarrow f' \leq f])$$

Our solver is able to vastly outperform the state-of-the-art solvers Z3 and CVC4.

Excel@FIT 2022

LaTeX TikZposter