SAT with differential equations

Author: Tomáš Kolárik (tomaqa@gmail.com)
Supervisor: Stefan Ratschan (stefan.ratschan@cs.cas.cz)

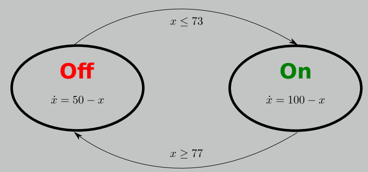
Faculty of Information Technology
CTU in Prague

Introduction

- Many systems, especially **embedded systems**, must nowadays satisfy high specification requirements, which often depend on physical features of the real world. **Formal verification** showed to be convenient method to guarantee specifications fulfillment in complex systems.
- A widely used approach of formal verification is **SAT**. Here, a problem arises when one needs to use another means of modeling—**differential equations** (**ODEs**), which describe **physical features** natively.
  - Neither SAT nor its arithmetic extensions handle ODEs.
  - Embedded systems are typically set into a physical environment.

Example: thermostat

- Thermostat is modeled by a simple state machine.
- Variable x stands for temperature.
  - Temperature must remain in given interval: $70 \leq x \leq 80$.
- State machine can be encoded into Boolean formula; additionally, each state describes different ODE.
- Proposed thermostat is only a demo example, SAT and its derivatives are aimed to much more complex tasks.



State of the art

- Solvers combining SAT with ODEs **already exist**. All state of the art solvers handle **ODEs** using **interval arithmetic**, which **prefers accuracy over speed**. This causes its usage to be limited in industry (practical instances might be huge).
  - It allows interval initial values of ODEs and guarantees maximal approximation error.
  - But it is **slow**.
- One of these solvers is e.g. **dReal** [1], which comes from a dissertation at Carnegie Mellon University, supervised by Turing award recipient, Edmund Clarke.

Objectives

1. **Prove the concept**, that handles ODEs with **classic numerical methods**.
   - These methods require exact initial values (initial value problems, IVPs), and might be less precise than interval arithmetic, but are **faster**.
   - Intervals can be efficiently approximated by value enumerations in logical sum.
   - Mostly known methods: Euler, Runge–Kutta, linear multistep.
2. Utilize these methods for the purposes of **formal verification**.
   - I.e. to **combine** selected ODE solvers **with SAT** solvers. Instead of pure SAT, **SMT** problem (Satisfiability Modulo Theories) was chosen, which extends SAT of arithmetic theories. SMT problem is still under active research.
3. **Compare performance** of our prototype implementation with dReal.

Components model

- The tool is named SOS (SMT+ODE Solver). It consists of four main parts: input processing, SMT solver, ODE solver and central component.
- Input language is similar to **SMT-LIB**, which is being standard among SMT solvers. Input is forwarded to SMT solver directly with minor modifications. Preprocessor allows **generic** code constructions (conditional, iterative or recursive) via **macros**, which can be useful for e.g. BMC (bounded model checking).
- Both ODE and SMT solvers are modeled as **standalone and exchangeable** components. Entire communication with SMT solver is based on SMT-LIB standard, thus provides high **flexibility**.