

SAT s diferenciálními rovnicemi

Diplomová práce

Kolárik Tomáš

kolarto5@fit.cvut.cz

Vedoucí práce: doc. Dipl.-Ing. Dr. techn. Stefan Ratschan

13. června 2018

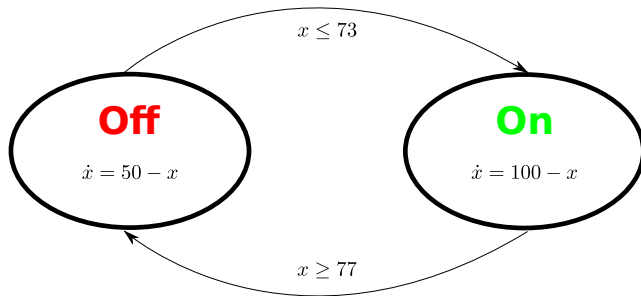
10 slajdů, 10 minut

Obsah prezentace

- 1 Motivace.
- 2 Cíle práce.
- 3 Ukázkové úlohy.
- 4 Model komponent.

Motivační příklad

- Ke *spolehlivé* činnosti přístroje je nutné dodržet provozní teplotu.
- Řešení: použití *termostatu*.



- Termostat popíšeme *automatem*.
 - ▶ Rozsah povolené teploty: $70 \leq x \leq 80$.
- Jak **verifikovat správnou funkci** termostatu?
 - ▶ Tj. dodržení rozsahu teploty.

Motivace

- Okolí *vestavných systémů* typicky vyžaduje popis pomocí **diferenciálních rovnic (ODE)**.
 - ▶ ODE: Ordinary Differential Equation.
- Samotný problém SAT *neovládá* ODE.
 - ▶ Ani jeho aritmetická rozšíření je neovládají.

Současný stav

- Řešiče kombinující SAT a soustavy ODE již existují.
- K řešení ODE ale používají *intervalovou aritmetiku*.
- Důsledky:
 - ▶ Umožňují intervalové počáteční podmínky ODE.
 - ▶ Garantují maximální dosaženou chybu.
 - ▶ Ale jsou **pomalé**.

Cíl práce

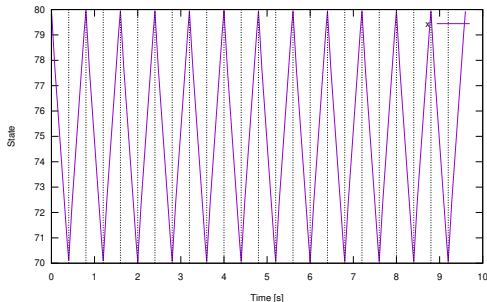
K řešení ODE použít **klasické numerické metody**.

- ① Ověřit koncept, který k řešení ODE používá **klasické numerické metody**.
 - ▶ Vyžadují jednoznačné počáteční podmínky.
 - ▶ Mohou být méně přesné, ale jsou **rychlejší**.
- ② Použít zvolené řešení ODE pro účely **formální verifikace**.
 - ▶ Kombinovat ODE a problém SMT.
 - ★ SMT: Satisfiability Modulo Theories.
 - ★ SMT rozšiřuje SAT o aritmetické teorie.
- ③ **Srovnat výkonnost** prototypu se stávajícím řešičem dReal.
 - ▶ dReal pochází z disertační práce na Carnegie Mellon University.

Ukázkové úlohy (1)

Termostat

- x ... provozní teplota.
- Nutné dodržet meze teploty: $70 \leq x \leq 80$.
- Systém je řízen časem:
 - ▶ předem dané časové okamžiky,
 - ▶ v nichž dochází k přechodům a kontrole specifikací.
- Srovnání délky výpočtu dReal a našeho prototypu: **46** a **0,5** s.
 - ▶ Tj. téměř **stonásobné zrychlení**.

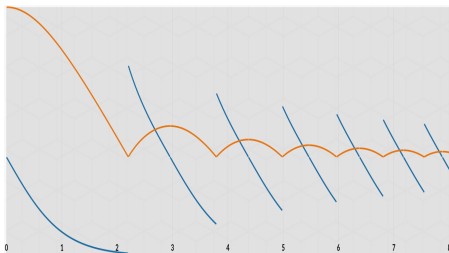


Ukázkové úlohy (2)

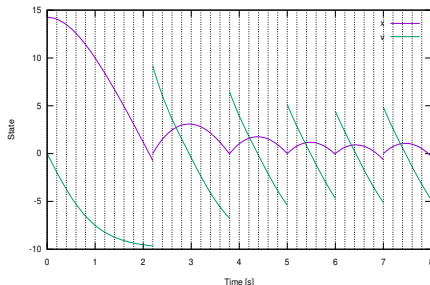
Skákající míč

- x ... výška míče, v ... rychlost.
- Míček musí setrvat nad podložkou: $x \geq 0$.
- Systém *není* řízen časem:
 - ▶ nutno často kontrolovat přechody a specifikace.
- Srovnání délky výpočtu dReal a našeho prototypu: 0,1 a 0,5 s.
 - ▶ Naše *implementace* je zatím pro tyto úlohy neefektivní.

(a) dReal



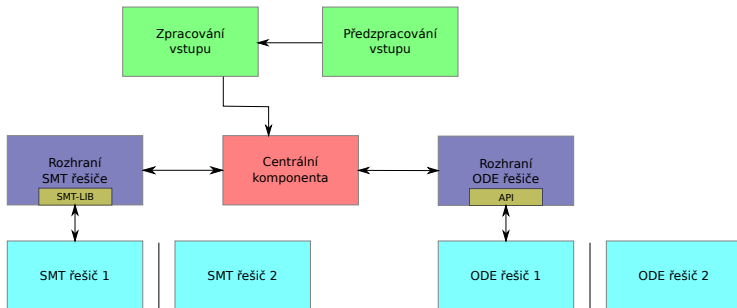
(b) Náš koncept



- Náš *prototyp* si v některých úlohách počíná mnohem **rychleji** než dReal.
 - ▶ Zejména v úlohách řízených časem.
- Tj. podařilo se mi **potvrdit**, že zvolený **koncept** je **nadějný** pro lepší použití **v praxi**.
 - ▶ Průmyslové instance mohou být velmi rozsáhlé.

Model komponent

- Vstupní jazyk je podobný standardu SMT-LIB.
- SMT i ODE řešič realizovány jako výměnné samostatné komponenty.
 - ▶ Lze použít libovolný SMT řešič konformní s SMT-LIB standardem.



- Cíl: aplikovat **odlišný přístup** v integraci ODE.
 - ▶ Méně přesné, ale **rychlejší** metody.
- Prototyp srovnán s řešičem dReal.
 - ▶ Naše řešení je výrazně **rychlejší** v některých úlohách.
- Zvolený **koncept** se mi podařilo **potvrdit**.
- Na práci budu dále pokračovat.

Zvýšení výkonu řešiče

- Algoritmické nedostatky převažují nad implementačními.
 - ▶ Chybí sofistikovanější prohledávání stavového prostoru.
 - ▶ SMT řešič počítá exaktně (zbytečně).
 - ▶ Srovnání délky výpočtu SMT a ODE řešiče: 95 % a 5 %.
 - Zajistit efektivnější rozdělení zátěže (s ohledem na prohledávaný prostor).
 - ▶ Úlohy, které nejsou řízeny časem, jsou počítány neefektivně.
 - ▶ Není využit paralelismus.
- Nejprve řešit algoritmickou stránku, až poté efektivní implementaci.
- Možnosti vyšší efektivity implementace:
 - ▶ (Jiné než procesorové architektury nepřichází v úvahu.)
 - ▶ Přizpůsobení kódu optimalizacím překladače (odstranění rekurze, vektorizace ...).
 - ▶ Efektivní implementace paralelismu.
 - ★ Využití cache pamětí.