State Space Compression for the DiVinE Model Checker

Vladimír Štill

17. června 2013

Explicitní model checking

- **verifikace vícevláknových aplikací**
  - nedeterminizmus $\rightarrow$ testování je těžké
- **automatická formální verifikace** platnosti konkrétní vlastnosti
- poskytnutí **protipříkladu**

Co je možné verifikovat?

- **nepřítomnost deadlocků**
- **neporušení assertions**
- vlastnosti v **temporální logice** (LTL, CTL, . . .)

DiVinE

- explicitní LTL model-checker vyvíjený v laboratoři ParaDiSe
- paralelní a distribuovaná verifikace
- rozmanité vstupní formáty
  - LLVM (C & C++), UppAal časové automaty, DVE, CoIn, Cesmi

Stavový prostor

- stav programu = **obsah paměti programu**
- tvoří orientovaný graf
  - stavy = vrcholy
  - hrany = relace následnictví
- automat
- nedeterminismus
- iniciální stavy, akceptující stavy
- zadán implicitně programem

Exploze stavového prostoru

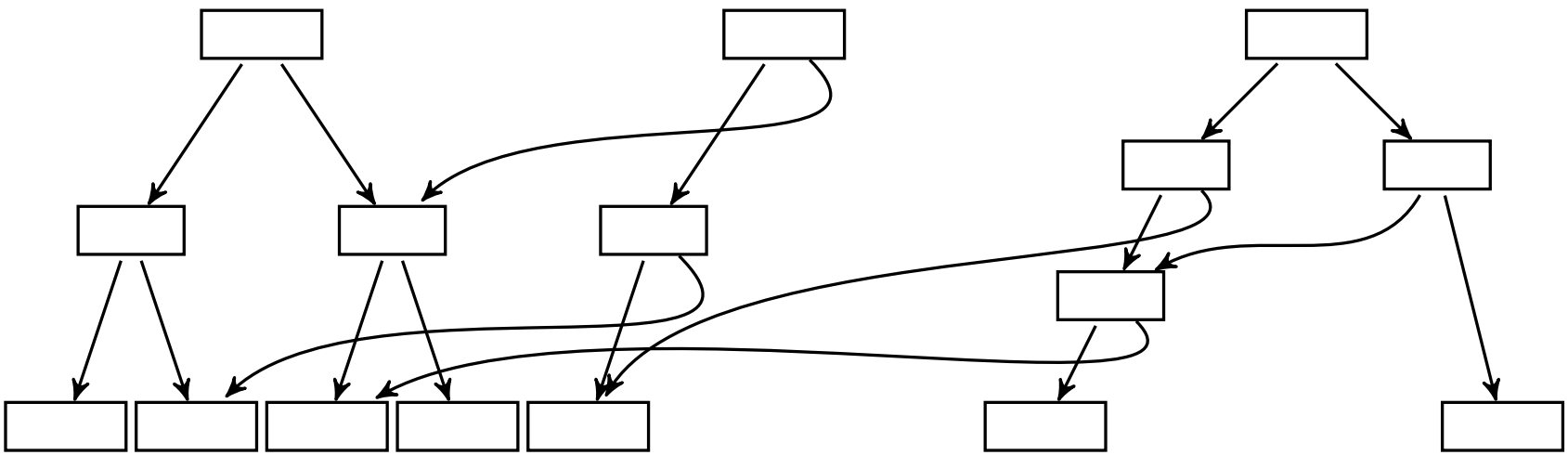
- stavů ve stavovém prostoru je mnohem více než v konkrétním běhu programu
- explicitní model checking – prohledávání stavového prostoru
- třeba udržovat množinu navštívených vrcholů

  ↓

  paměťově náročné

Stromová komprese stavového prostoru

- malé změny paměti mezi stavy
  - zbytečná redundance
- stav rekurzivně rozdělen – stromová struktura
  - podstromy jsou znovu využívány
  - list stromu reprezentuje nedělitenou část stavu
  - kořen jednoznačně reprezentuje stav
  - vnitřní uzel jednoznačně reprezentuje část stavu



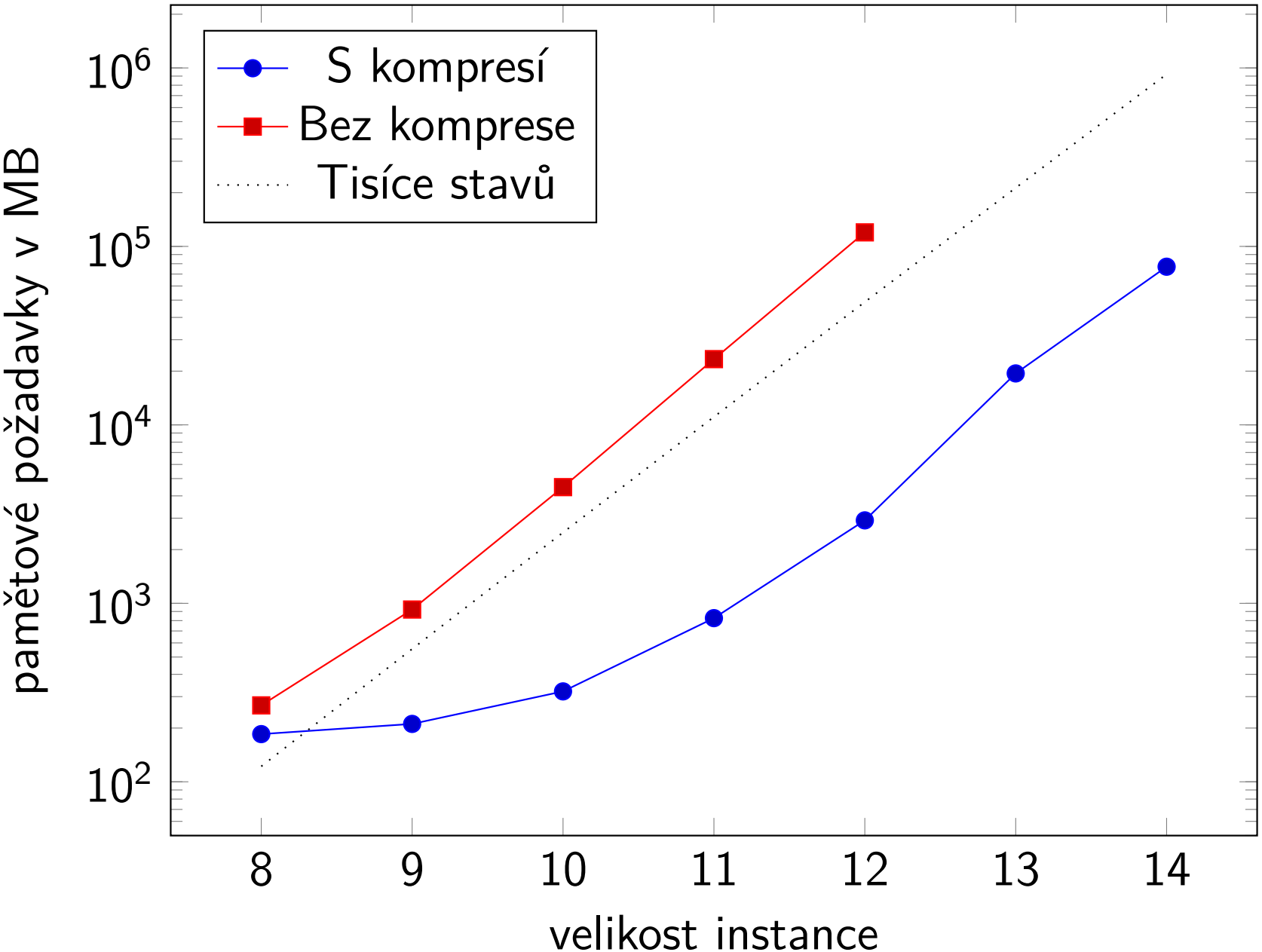
Stromová komprese – přínosy mé práce

- implementována v DiVinE a integrována s ostatními funkcionalitami
- nezávislá na algoritmu
- nezávislá na vstupním formalismu
- funguje s redukcemi zaměřenými na množství stavů
- komprese pomocných datových struktur

Stromová komprese – přínosy mé práce

- snadno použitelná pro uživatele
- velká úspora paměti
  - až 42x v testech
- malý vliv na rychlost verifikace

Příklad stromové komprese



Děkuji za pozornost

Otázky. . .