Rozšíření nástroje DIVINE pro verifikaci vstupně-výstupně otevřených programů

Henrich Lauko Jan Mrázek Vladimír Štill
garant: Jiří Barnat

ParaDiSe
Parallel & Distributed Systems Laboratory

Masarykova univerzita
Brno, Česká republika

27. dubna 2017

Cíle projektu

- nástroj DIVINE pro ověřování vlastností vstupně-výstupně uzavřených programů
- nástroj SymDIVINE – prototypový nástroj pro ověřování vlastností vstupně-výstupně otevřených programů

Cíle projektu

- nástroj DIVINE pro ověřování vlastností vstupně-výstupně uzavřených programů
- nástroj SymDIVINE – prototypový nástroj pro ověřování vlastností vstupně-výstupně otevřených programů

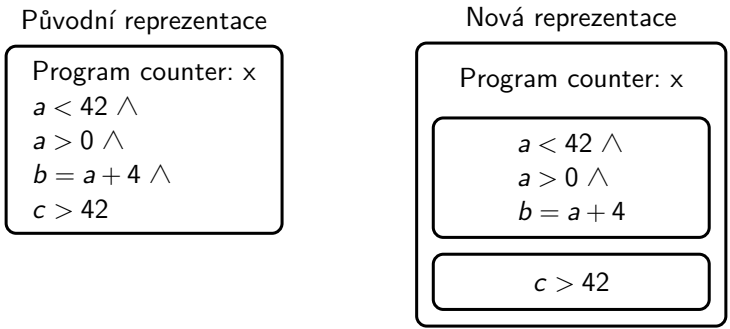
- cíle projektu jsou
  1. navržení techniky pro efektivnější využívání SMT solveru v dotazech v nástroji SymDIVINE za pomoci cache dotazů
  2. integrace DIVINE a SymDIVINE

Cíl 1: Cache dotazů pro SymDIVINE

- SymDIVINE generuje symbolické stavy programu
  - kvantifikované dotazy na SMT solver pro rovnost stavů
  - nekvantifikované dotazy na SMT solver pro prázdnost stavů
- inkrementální tvorba stavů $\rightarrow$ podobné dotazy
- naivnímu využití cache dotazů brání kvantifikátory

Cíl 1: Cache dotazů pro SymDIVINE

- technika rozdělení stavů na datově nezávislé komponenty



- reprezentace pomocí datově nezávislých komponent
- porovnání po komponentách
  - umožňuje cachovat dotazy na nezměněné komponenty
  - komponenty je třeba spojovat podle struktury obou stavů

Cíl 1: Cache dotazů pro SymDIVINE

	Bez cache		S cache	
Kategorie	Čas[s]	Vyřešeno	Čas[s]	Vyřešeno
Concurrency	1828	40	**1506**	**42**
DeviceDrivers	12156	241	**763**	**298**
ECA	**20794**	**230**	21606	211
ProductLines	19571	276	**11995**	**293**
Sequentialized	3710	44	**1735**	**47**
Celkem	58061	831	**37607**	**891**

- prezentace na MEMICS
- prezentace v rámci soutěže SV-COMP
- publikace v přípravě

Cíl 2: Integrace DIVINE a SymDIVINE

- SymDIVINE přináší podporu pro vstupně-výstupně otevřené programy v C/C++
  - jedná se ale o prototypový nástroj sloužící primárně k demonstraci použitelnosti techniky Control-Explicit Data-Symbolic model checkingu
  - z dlouhodobého hlediska je nepraktické udržovat DIVINE i SymDIVINE

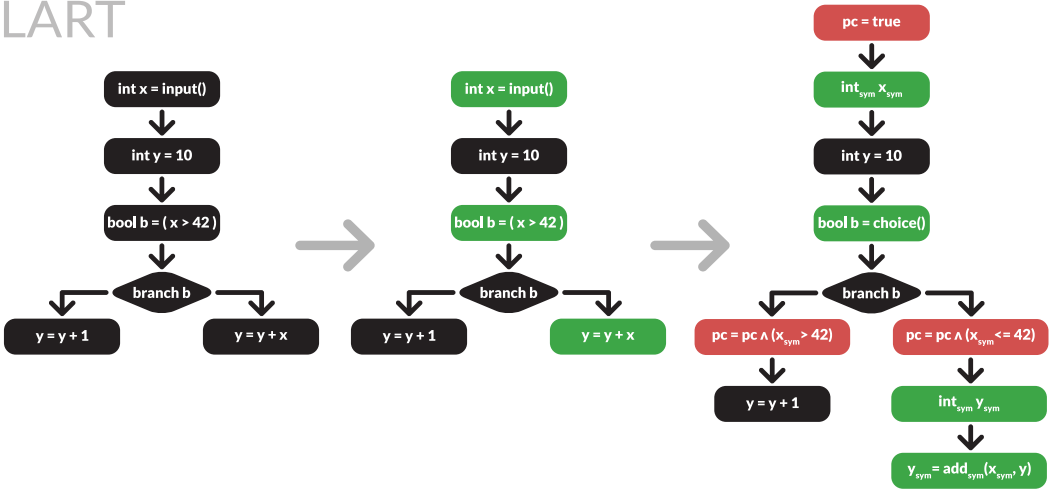
Cíl 2: Integrace DIVINE a SymDIVINE

- SymDIVINE přináší podporu pro vstupně-výstupně otevřené programy v C/C++
  - jedná se ale o prototypový nástroj sloužící primárně k demonstraci použitelnosti techniky Control-Explicit Data-Symbolic model checkingu
  - z dlouhodobého hlediska je nepraktické udržovat DIVINE i SymDIVINE
- potřeba integrovat SymDIVINE do DIVINE bez zásadního zkomplikování jádra DIVINE

Cíl 2: Integrace DIVINE a SymDIVINE

- myšlenka: zakódování manipulací se symbolickými daty do vstupního programu + algoritmus schopný procházet symbolický stavový prostor

LART



Cíl 2: Integrace DIVINE a SymDIVINE

- symbolická data = vstupy
  - lze reprezentovat pomocí formulí v bitvektorové logice
  - v SymDIVINE reprezentaci vytváří model checker
  - DIVINE instrumentuje program tak, aby formuli vytvářel sám

Cíl 2: Integrace DIVINE a SymDIVINE

- symbolická data = vstupy
  - lze reprezentovat pomocí formulí v bitvektorové logice
  - v SymDIVINE reprezentaci vytváří model checker
  - DIVINE instrumentuje program tak, aby formuli vytvářel sám
- do DIVINE přidány:
  - podpora pro označování části paměti za symbolickou
    - symbolická paměť porovnávána pomocí SMT solveru
    - využívá nového způsobu reprezentace haldy v DIVINE 4
  - interní formát reprezentace formulí
  - základní verze instrumentace/symbolizace programu
  - podpora pro volání SMT solveru