Szoftver-modellellenőrző eszköz: CPAchecker

dr. Majzik István
BME Méréstechnika és Információs Rendszerek Tanszék

Szoftver-modellellenőrzés: CPAchecker

- The Configurable Software-Verification Platform
- Bemenet: C program + specifikáció
  - Ellenőrzések: Assertion, error címke, deadlock, null dereference, …
- Nagymértékben konfigurálható
  - Többféle absztrakció (nem csak predikátumok)
- Célok integrálása
  - Program analízis: Nagy programokról, egyszerű tulajdonságok
  - Modellellenőrzés: Egyszerűbb programokról, teljes viselkedési analízissel felderíthető tulajdonságok
- Elérhető:
  - http://cpachecker.sosy-lab.org/ (Java alapú, letölthető)
  - http://cpachecker.appspot.com/ (online használatra)

Alkalmazott módszerek

- CEGAR: predikátum absztrakció, lusta absztrakciófinomítás, interpoláció
- Korlátos modellellenőrzés
- Szimbolikus végrehajtás

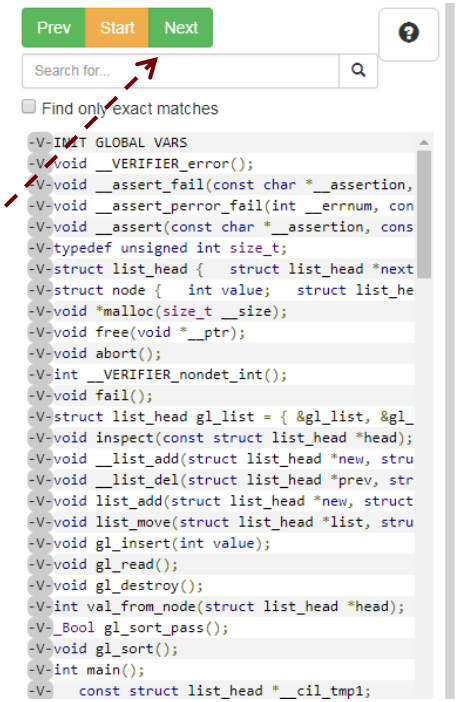
Verifier	CEGAR	Predicate Abstraction	Symbolic Execution	Bounded Model Checking	k-Induction	Property-Directed Reachability	Explicit-Value Analysis	Numerical Interval Analysis	Shape Analysis	Separation Logic	Bit-Precise Analysis	ARG-Based Analysis	Lazy Abstraction	Interpolation	Automata-Based Analysis	Concurrency Support	Ranking Functions
APInfVE			✓				✓	✓			✓						✓
Blocca	✓	✓		✓													
Blast	✓	✓					✓					✓	✓	✓			
Cascade			✓	✓							✓						
Cbmc				✓	✓						✓					✓	
CPAchecker	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓			

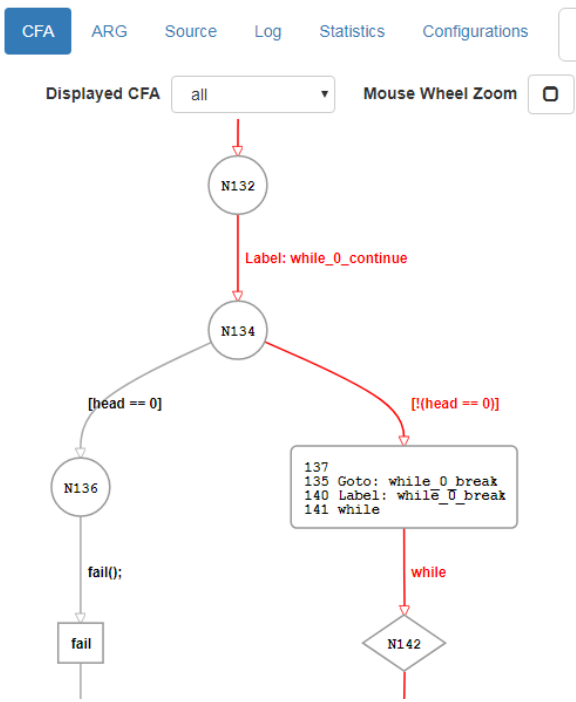
Bemenet

- C program előfeldolgozás után
  - Makrók, include, … feloldása
  - gcc –E program.c > program.i
- Támogatott nyelvi konstrukciók:
  Az alkalmazott SMT megoldótól függ
  (pl. Linux: MathSAT5, Windows: SMTInterpol)
  - Integer, float
  - Array, bitvector, list
  - Pointer, heap
  - Dinamikus memóriakezelés,
  - Rekurzió, többszálú konkurens működés, …
- Több fájl együttes kezelése

Kimenet

- HTML jelentés (interaktív)
  - CFA: Control Flow Automaton
  - ARG: Abstract Reachability Graph
  - Forráskód (utasításokra ugrás a CFA éleiről)
  - Ellenpélda:
    - Programút bemutatása: forráskód lefutás, CFA-n illetve ARG-n
    - Lépésenként bejárható végrehajtás (CFA és forráskód követés)
    - Változóértékek követése (ld. -V-)
  - Statisztikák





Konfigurálhatóság: Specifikációk

- Parancssori interfész
  - cpa.bat -config <config> -spec <spec> <prog.c>
- Specifikációk és konfigurációk
- Specifikációk:
  - Assertion.spc: assertion megsértése a kódban
  - ErrorLabel.spc: ERROR címke elérése a kódban
  - TerminatingFunctions.spc: terminál függvény (pl. exit, abort)
  - Deadlock.spc: holtpont konkurens programban
  - Null-deref.spc: null-pointer dereferencia
  - UninitilizedVariables.spc: Nem initializált változó, return érték
  - Memóriakezelés: MemorySafety-deref (invalid derefencing of pointers), MemorySafety-free (invalid freeing of allocations), MemorySafety-memtrack (memory leaks)

Konfigurálhatóság: Konfigurációk

- Parancssori interfész
  - cpa.bat -config <config> -spec <spec> <prog.c>
- Specifikációk és konfigurációk
- Konfigurációk (több, mint száz):
  - Absztrakciók (predicate, value, …), finomítások, ellenőrzések
  - Alapeset: default.properties, szekvenciális analízisek
    - Predikátum absztrakció túlcsordulás ellenőrzéshez
    - Lasso analízis terminálás ellenőrzéshez
    - BDD használata konkurens programokhoz
    - Predikátum absztrakció rekurzív programokhoz
    - k-indukció egyéb esetekre
  - Ajánlott konfiguráció:
    - predicateAnalysis.properties (CEGAR)
    - predicateAnalysis-bitprecise.properties
- Nincs „legjobb” specifikáció, illetve konfiguráció

Demo

- Assertion ellenőrzése
  - Sikeres és hibát jelző ellenőrzés (kód módosítás után)
  - Ellenpélda bemutatása: Forráskód, változók, CFA, ARG
- ERROR címke ellenőrzése
  - Ellenpélda bemutatása: Forráskód, CFA, ARG
- Bubbleshort algoritmus verifikálása
  - Több CFA (függvényenként)
- Linux Driver Verification Project
  - Problems in Linux Kernel Found by CPAchecker