

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
CPACHECKER	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			

Bemenet

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

Kimenet

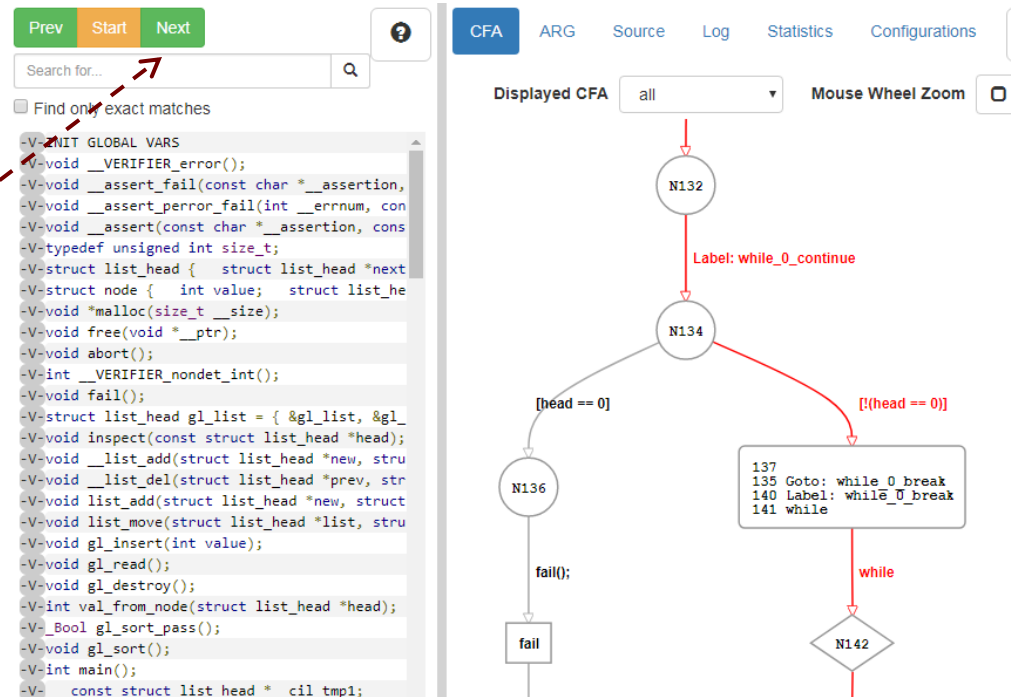
- HTML jelentés (interaktív)

- CFA: Control Flow Automaton
- ARG: Abstract Reachability Graph
- Forráskód (utasításokra ugrás a CFA éleiről)

- Ellenpélda:

- Programút bemutatása a CFA-n illetve az ARG-n
- Lépésenként bejárható végrehajtás (CFA és forráskód követés)
- Változóértékek követése

- Statisztikák



Konfigurálhatóság: Specifikációk

- Parancssori interfész
 - `cpa.bat -config <config> -spec <spec> <prog.c>`
- Specifikációk és konfigurációk
- **Specifikációk:**
 - `Assertion.spc`: assertion a kódban
 - `ErrorLabel.spc`: ERROR címke a kódban
 - `TerminatingFunctions.spc`: termináló függvény (pl. `exit`, `abort`)
 - `Deadlock.spc`: holtpont konkurens programban
 - `Null-deref.spc`: null-pointer dereferencia
 - `UninitializedVariables.spc`: Nem inicializált 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

- ERROR címke ellenőrzése
 - Kód módosítás
 - Ellenpélda bemutatása: CFA, ARG
- Assertion ellenőrzése
 - Kód módosítás
 - Ellenpélda bemutatása: CFA
- Bubblesort algoritmus verifikálása
- Linux Driver Verification Project
 - Problems in Linux Kernel Found by CPAchecker