Title: Product abstraction-based strategies for efficient software model checking

Keywords: formal methods, software verification, abstraction

Software systems are controlling devices that surround us in our everyday life. Many of these systems are safety-critical (e.g., autonomous vehicles, power plants), thus ensuring their correct operation is gaining increasing importance. Formal verification techniques can both reveal errors and give guarantees on correctness with a sound mathematical basis. One of the most widely used formal verification approaches is model checking, which systematically examines all possible states and transitions (i.e., the state space) of the software. However, a major drawback of model checking is its high computational complexity, often preventing its application on real-life software.

Counterexample-guided abstraction refinement (CEGAR) is a supplementary technique, making model checking more efficient in practice. CEGAR works by iteratively constructing and refining abstractions in a given abstract domain. There are several existing domains, such as explicit-values, which only track a relevant subset of program variables and predicates, which use logical formulas instead of concrete values. Observations show that different abstract domains are more suitable for different kinds of software systems. Therefore, so-called product domains have also emerged that combine different domains into a single algorithm.

In this work, we develop and examine various strategies to combine the explicit-value domain with predicates. Our approaches use different information from the already explored abstract state space to guide further exploration more efficiently. We implement our new strategies on top of Theta, an open source verification framework. This allows us to perform an experiment with a wide range of software systems including industrial PLC codes. We evaluate the strengths and weaknesses of the different approaches and we also compare them to existing methods. Our experiment shows that the new strategies can form efficient combinations of the existing algorithms.

Cím: Szorzat absztrakció-alapú stratégiák hatékony szoftver-modellellenőrzéshez

Kulcsszavak: formális módszerek, szoftver verifikáció, absztrakció

Mindennapi életünket egyre jobban meghatározzák a szoftverrendszerek. Ezek sokszor biztonságkritikusak (pl. autonóm járművek, erőművek), tehát helyes működésük garantálása kiemelten fontos feladat. Ennek egyik eszköze a formális verifikáció, ami a hibák jelenlétét és a helyes működést is képes matematikailag precíz módon bizonyítani. Az egyik legelterjedtebb formális verifikációs módszer a modellellenőrzés, amely a program összes lehetséges állapotát és átmenetét (azaz állapotterét) szisztematikusan megvizsgálja. A módszer egyik hátránya viszont a nagy számítási igénye, ami gyakran megakadályozza használatát valós szoftvereken.

Az ellenpélda-alapú absztrakciófinomítás (angolul Counterexample-Guided Abstraction Refinement, CEGAR) egy olyan kiegészítő technika, melynek segítségével a modellellenőrzés hatékonyabbá tehető. Működése során a CEGAR iteratívan hozza létre és finomítja az ellenőrzendő probléma egy absztrakcióját. Az irodalomban több különböző absztrakciós megközelítés létezik, például az explicit változók módszere, illetve a predikátumabsztrakció. Előbbi a programnak csak a verifikáció céljából releváns változóit tartja nyilván, míg az utóbbi konkrét értékek helyett matematikai kifejezések teljesülését vizsgálja. Korábbi eredmények alapján megfigyelhető, hogy különböző absztrakciós

módszerek különböző típusú szoftvereken működnek hatékonyabban. Ebből kifolyólag létrejöttek úgynevezett szorzat absztrakciók, amik többféle módszert kombinálnak egy algoritmusban.

Munkám során eltérő stratégiák alapján kombináltuk az explicit változókat predikátumokkal. Megközelítésünk lényege, hogy a már felderített absztrakt állapottérből kinyert információk figyelembe vételével a további felderítést és ellenőrzést hatékonyabbá teszi. Ezeket az új stratégiákat a Theta nevű nyílt forráskódú verifikációs keretrendszerben implementáltuk. Ennek segítségével szoftverrendszerek széles skáláján tudtuk lefuttatni méréseinket, többek között ipari vezérlő (PLC) kódokon. Összevetettük a különböző stratégiák előnyeit és hátrányait, és a már létező módszerekkel is összehasonlítottuk őket. Az eredményeink azt mutatják, hogy az új módszereink hatékonyan tudják kombinálni a meglévő algoritmusok előnyeit.