

Kivonat

Életünk közel minden pontján megtalálhatók a szoftverek. Legyen szó akár a zsebünkben lapuló mobiltelefonról, akár a háztartásokban megtalálható újszerű mosógépekről, de gondolhatunk például az üzleti célú szoftverekre (banki szolgáltatások), illetve ma már az autóipar is erősen szoftverorientáltnak mondható.

Napjainkban szerencsére egyre több emberben tudatosul, hogy mekkora jelentősége is van a forráskódok ellenőrzésének, főleg biztonságkritikus rendszerek esetében. Egy lehetőség az ellenőrzésre maga a tesztelés, vagyis a kódunk futtatás során történő vizsgálata. Ilyen tesztek elvégzésének megtervezése sokféleképpen történhet, és az egyik ilyen módszer a specifikációra alapszik. Ennek során megpróbáljuk olyan lefutásokra bontani az elkészült programot, amelyek hasonló működést eredményeznek, és ezeket az útvonalakat próbáljuk minél széleskörűbben letesztelni. Egy ilyen értékkészletet, amelynek értékeire hasonlóan működik a program, egy ekvivalenciapartíciónak nevezzük. Azonban ezeket a partíciókat meghatározni nagyon bonyolult is lehet, így előfordulhat, hogy nem sikerül teljeskörűen letesztelnünk ilyen módon az elkészült alkalmazást.

Az Önálló laboratórium 1. tantárgyban a szakdolgozatom folytatásaként egy olyan alkalmazás elkészítésén dolgoztam, amelynek célja az előbb ismertetett teszttervezési módszer támogatása. Munkám során továbbra is azt vizsgáltam, hogyan lehet automatizáltan, pusztán a forráskód alapján meghatározni a kód alapján adódó ekvivalenciapartíciókat, hogy ezt később össze lehessen vetni a specifikáció alapján, kézzel meghatározott partíciókkal.

A szakdolgozatom során elkészült prototípust a nagyobb megbízhatóság és jobb felhasználhatóság érdekében új alapokra helyeztem. Ebben a dokumentumban egyrészt a munkám megértéséhez szolgáló háttérismeretek bemutatása a célom, másrészt az újonnan elkészítendő alkalmazás jelen pillanatban elkészült részét szeretném részletesen bemutatni. A dokumentum végén a további haladási irányról is szót ejtek.

Abstract

People interact with software in nearly every point of their lives. Just think about smartphones, recent washing machines in households, but business applications (e.g. banking) and even cars today fall under a heavily software-based category too.

Today, fortunately a lot of people realise how important it is to test, validate and verify the written code, especially in case of safety critical systems. An opportunity to check code is to test it, namely to check it while it's running. These kind of tests can be executed in a plethora of different ways, but one of these techniques are the ones that are based on specification. One of them is when we try to separate the written program into execution ways that are similar to each other. One codomain, from which the input parameters get their values from, and the written program behaves the same way, is called an equivalence partition. It can be really complex to determine these partitions. Eventually it might happen that the test design isn't sufficient enough, and the program won't be tested thoroughly.

In my current work as a sequel to my thesis my goal is to create an application, which aims to assist the method mentioned before. In my work I investigated further how determining these equivalence partitions can be done, based solely on the source code, so it can be compared to the ones defined by hand, based on the specification.

I've rewritten the prototype I made before in my thesis on a new base. In this document, I aim to present you the background knowledge needed to understand my work, and also the completed part of my current work. At the end of the document, I would like to mention some words about how I intend to continue my work.