

# Kivonat

Napjaink komplex, elosztott és kritikus rendszerei új kihívások elé állítják a rendszermérnököket. Ez különösen igaz az utóbbi időben, amikor a gyorsan változó igények miatt a rendszerek újrakonfigurálása, áttervezése is a napi rutin részévé vált. Ez a gyors fejlődés azonban problémákat vet fel, fontos kérdéssé vált a rendszerek új állapotának az ellenőrzése: vajon a változtatások elrontották-e a rendszer viselkedését. Ilyenkor nyújt megoldást a regressziós tesztelés, ahol azt vizsgáljuk, hogy a rendszer változásai okoznak-e az eddigi viselkedéstől való nem kívánt eltérést.

A regressziós tesztelés támogatása fontos, de nehéz probléma. Sokat segít a regressziós tesztelésben, ha a rendszernek elkészül egy modellje, amely az elvárt viselkedéseket tartalmazza. Ezen modell elkészítésére azonban a fejlesztőknek gyakran nincs ideje, utólag pedig csak jóval költségesebben, a forráskódok és a konfigurációs fájlok visszafejtésével lehet előállítani.

Munkánk célja egy olyan módszer kidolgozása, amely képes a regressziós tesztelés támogatására specifikációs modellek automatikus szintézisével. A megközelítés során az egyes komponenseket külön-külön próbáljuk megtanulni és a komponensmodelleket előállítani. A tradicionális automatatanuló algoritmusok erre nem bizonyultak alkalmasnak, mivel nem kezelik a rendszer adatfüggő viselkedéseit, ezért kidolgoztunk egy absztrakciós keretrendszert, amely lehetővé teszi a komponens-bemenetek konfigurálható absztrakcióját és a kimeneti adatok szintézisét, ezáltal támogatva a rendszer komponenseinek tanulását. Az így előállított modellek alapján teszt szekvenciákat generálunk, amelyek a fejlesztés későbbi fázisaiban alkalmazhatóak regressziós tesztelésre. Amennyiben a regressziós tesztelés sikeres, a specifikációs automatát az új rendszerből megtanulva a két automata ekvivalenciáját az általunk fejlesztett keretrendszer ellenőrzi.

Dolgozatunkban bemutatjuk az általunk kidolgozott keretrendszert, amely nyelvi támogatást ad konfigurálható módon absztrakciók definiálására, majd ennek megfelelően megvalósítja a komponensek bemeneteinek absztrakcióját és a kimenetek konkretizációját. Emellett támogatja a megtanult modell alapján különböző feltételeknek megfelelő teszt szekvenciák generálását. A keretrendszer képes ellenőrizni a rendszer különböző fejlesztési fázisokban megtanult modelljeit ekvivalencia-ellenőrzéssel, így támogatva a különböző szoftververziók összehasonlítását és a változtatások ellenőrzését.

A módszer segítségével hatékonyan támogatjuk a regressziós tesztelést. Munkánk által lehetővé válik olyan szoftverkomponensek regressziós tesztelése, amelyekhez eddig nem állt rendelkezésre a működést leíró modell.

# Abstract

Nowadays, the development of complex, distributed and critical systems yield a huge challenge to systems engineers. Ensure the correct behaviour is especially difficult in evolving environments where the frequent changes in demands leads to the frequent reconfiguration and redesign of the systems. This rapid evolution raises additional problems. It has become important to verify the new state of the system, whether it satisfies the original specification at least as much as the previous version. Regression testing is a solution, as it searches for unintended divergence in the behaviour caused by the changes in the system.

Supporting regression testing is an important though difficult problem. Creating a model for the desired behaviour of the system contributes to regression testing. However developers usually do not have time to construct the specification model and it is costly to reconstruct afterwards from the source and configuration files.

The goal of our work is to develop a methodology to support regression testing with automatic synthesis of specification models. Our approach learns each component separately and produces behaviour models from them. Traditional automata learning algorithms proved to be insufficient for this task, as they do not deal with the data-dependent behaviour of the system. Therefore we developed a language framework to facilitate configurable abstractions of the inputs and the synthesis of the outputs for each component: our solution supports the learning of the model of even complex components. Based on these models we generate test sequences to be applied for regression testing in a later phase of development. In case the regression test succeeds, our framework could perform an equivalence test between the specification automata learnt from the new and the former system implementation.

In this report we introduce our framework which gives us a language to define and configure abstractions. Accordingly, it performs the abstraction of the inputs and concretization of the outputs in the components. Moreover it supports the generation of test sequences based on the learnt model and additional criteria. Also it is able to verify the learnt models from each development phase of the system with equivalence check by comparing the different versions of the software model and verifying the changes.

Our method provides an efficient support for regression testing of software components without the manual construction of specification models describing their behaviour.