Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Adnrej Čižmárik
Název práce	Dynamic Analysis Framework for C#/.Net Programs
Rok odevzdání	2020
Studijní program	Informatika **Studijní obor** Softwarové systémy

Autor posudku	Lubomír Bulej **Role** oponent
Pracoviště	Katedra distribuovaných a spolehlivých systémů

Text posudku:

Nástroje pro dynamickou analýzu programů umožňují analyzovat chování programu za běhu a získat tak informace, které je obtížné, nebo nemožné získat analýzou zdrojového kódu. Takové informace mohou programátoři využít při hledání chyb, optimalizaci běhu, či údržbě programů. Tvorba nástrojů pro dynamickou analýzu programů je poměrně náročná a náchylná na chyby, protože vyžaduje poměrně hluboké znalosti o fungování běhového prostředí a chování systému.

V tomto kontextu bylo cílem práce vytvořit framework pro dynamickou analýzu programů napsaných v jazyce C# na platformě .NET, který by měl usnadnit vývoj různých analytických nástrojů. Aby byl framework použitelný pro různé typy analýz, měl by být konfigurovatelný a rozšiřitelný pomocí pluginů, které budou definovat způsob zpracování událostí o chování programu generovaných frameworkem.

Celkově je práce spíše implementačního charakteru (v celkovém rozsahu necelých 9 tisíc řádků kódu v jazyce C#), přičemž technická obtížnost práce je dána především nutností porozumět instrukční sadě *Common Intermediate Language* (CIL) a virtuálnímu stroji, který ji vykonává, na takové úrovni, aby bylo možné vytvořit program, který bezpečně modifikuje jiné programy tak, aby při běhu generovaly události potřebné pro analýzu jejich běhu. Přitom je nutné se vyhnout řadě nástrah spojených s během kódu dynamické analýzy v adresovém prostoru sdíleném s analyzovaným programem.

Tato problematika je popsána v textové části práce, která zdařile doplňuje implementační část. Textová část je přehledně členěna na logické celky odpovídající analýze, návrhu a implementaci, a vyhodnocení. V části věnované analýze autor poskytuje obecný úvod do problematiky dynamické analýzy a technický popis běhového prostředí *Common Language Runtime* (CLR). V detailu se věnuje překladu konstrukcí jazyka C#, pro které framework generuje události, do posloupnosti instrukcí CIL, což je nutné pro správnou instrumentaci. V části věnované implementaci pak autor popisuje návrh a implementaci nástroje *SharpDetect*, který sestává z offline části, která zajišťuje instrumentaci programu a z online části, která je přítomna za běhu programu a koordinuje distribuci událostí pluginům, které implementují konkrétní analýzu. V části věnované vyhodnocení popisuje vzorové implementace dvou typů analýz pro hledání časové závislých chyb a na příkladech demonstruje jejich správnou funkci.

Pokud bych měl práci něco vytknout, bylo by to zejména v analytické části, kde opomíjí podstatnou část related work související s instrumentačním frameworkem DiSL a frameworkem pro běh analýz v odděleném adresovém prostoru ShadowVM. Přestože DiSL a ShadowVM cílí primárně na platformu Java (resp. Android), ukazuje se, že technické problémy spojené s instrumentací a dynamickou analýzou sdílející adresový prostor s analyzovanou aplikací jsou přenositelné mezi platformami. A protože řada problémů souvisejících s platformou Java (resp. Android) byla v uvedeným nástrojích nějakým způsobem vyřešena, bylo by vhodné se jimi podrobněji zabývat již ve fázi analýzy a přizpůsobit tomu návrh řešení, případně některá rozhodnutí lépe zdůvodnit v kontextu podobných prací—např. proč nedělat instrumentaci online, případně nevyčlenit kód analýzy do odděleného adresového prostoru.

Celkově však práci považuji za úspěšnou a v souladu se zadáním, přičemž autor bezpochyby prokázal svou schopnost řešit netriviální technický problém, stejně jako schopnost toto řešení zpracovat na solidní technické úrovni.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

V Praze dne 27. 1. 2020

Podpis: