Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Bc. Jakub Saksa		
Název práce	Syntax-driven duplicate-code detection		
Rok odevzdání	2020		
Studijní program	Informatika	**Studijní obor**	Softwarové a datové inženýrství

Autor posudku	RNDr. David Bednárek, PhD	**Role**	Vedoucí
Pracoviště	KSI MFF UK		

Text posudku:

Hlavním výsledkem práce je mechanismus pro indexaci syntakticky uzavřených fragmentů zdrojového kódu a navazující systém pro detekci duplicit, použitelný především pro primární detekci plagiátů, ale i jako vodítko pro softwarově-inženýrské metriky a refaktorizaci rozsáhlého kódu. Vzhledem k závislosti na syntaxi je systém zatím implementován pro jediný jazyk - Python - případná adaptace pro další jazyky je možná, ikdyž způsob oddělení jazykově-závislých částí od zbytku systému není řešen nejlepším možným způsobem.
Součástí práce jsou experimenty, využívající zdrojové texty odevzdané studenty do systému ReCodEx - výsledky celkem jasně ukazují, že systém dokáže odlišit zdrojové texty vložené tímtéž autorem (jako postupně vytvářené verze téhož řešení) od zdrojových textů jiných autorů. Pravdou ovšem je, že obdobných výsledků by bylo možno dosáhnout i jednoduššími prostředky a skutečná aplikovatelnost implementovaného systému při detekci plagiátů zatím testována nebyla, především proto, že spojení mezi tímto systémem a ReCodExem je zatím poněkud krkolomné.
Samotná indexace syntakticky uzavřených fragmentů (tedy podstromů derivačního stromu) je provedena relativně jednoduchým způsobem, který nicméně dokáže překlenout přejmenování identifikátorů a funguje tedy jak v případě plagiátů maskovaných přejmenováním, tak v případě, kdy různí programátoři při nezávislém řešení téhož problému dospěli k logicky shodnému kódu, čímž je umožňeno nasazení systému i pro refaktorizaci (ačkoliv pro tento účel zatím neexistuje vhodné uživatelské rozhraní).
Práce tedy nepředstavuje žádný převrat v problematice, nicméně implementuje dosud nevyzkoušenou kombinaci několika přístupů, přičemž hlavní předností je úspornost a rychlost indexace, která umožňuje aplikaci na rozsáhlejší data.
Samotná implementace je pozoruhodný konglomerát jádra v C++, GUI v Javě a pomocných pythonovských modulů a svědčí o tom, že se autor orientuje v prostředí multiplatformního programování. Některé podstatné detaily implementace ovšem mohly být zvládnuty lépe - systém se např. nedokáže vyrovnat ze změnou některých atributů v mezikódu při změně instalované verze Pythonu z 3.7 na 3.8. Obdobná nadměrná citlivost se týká i rozdílu mezi verzemi Javy 10 a 11 - i v tomto případě je sice k citlivosti technický důvod, zároveň si však lze představit řešení, které by se s odlišností vyrovnalo.
Text práce v zásadě odpovídá charakteru díla, které kombinuje nepříliš komplikovanou teorii s praktickou implementací. Text je srozumitelný, je však zřejmé, že ve formálnějších částech autor nepostupuje s potřebnou suverenitou.
Výše kritizované nedostatky znamenají, že jak implementace, tak text práce nejsou ideální, nicméně stále jsou velmi dobře použitelné a mohou se stát základem jak pro praktickou implementaci systému pro detekci plagiátů, tak pro návrh nových algoritmů v této oblasti.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

Pokud práci navrhujete na zvláštní ocenění (cena děkana apod.), prosím uveďte zde stručné zdůvodnění (vzniklé publikace, významnost tématu, inovativnost práce apod.).

Datum 4.9.2020 **Podpis**