

# 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. Michal Kopecký, Ph.D.      **Role** Oponent  
**Pracoviště** KSI MFF UK

## Text posudku:

Cílem práce bylo navrhnout, a implementovat nástroj pro detekci duplikovaných fragmentů ve zdrojovém kódu, schopný detekovat i mírně upravené duplikáty a zohledňující syntaktická pravidla daného jazyka. Jednak pro hodnocení udržovatelnosti kódu, ale hlavně pro odhalování plagiarismu v rámci odevzdávaných studentských prací.

Na práci musím kladně hodnotit řadu jejích rysů.

- Problém možných duplicit je na začátku práce pěkně popsán a vysvětlen na příkladech
- Stejně tak jso v práci přehledně popsány metriky na výpočet podobnosti stromů a podstromů

Celkově je práce psaná v pěkné angličtině a je dobře strukturovaná, takže se poměrně dobře čte. Zahrnuje vše podstatné od analýzy problému až po implementační detaily a návod k použití. Potřebné algoritmy jsou zároveň srozumitelně popsány formálně.

Výsledkem je aplikace PyPlug, která umí porovnávat zdrojové kódy jazyka Python 3.

Domnívám se, že práce po všech stránkách splnila zadání a požadavky na práce diplomové, a proto ji doporučuji k obhajobě. K předloženému řešení bych měl pár dotazů:

- Podle popisu se vždy porovnává právě načtený zdrojový kód, který prošel parserem a hashovacím modulem s uloženými předchozími zdrojovými kódy. Neuvažovalo se o možnosti nejprve zdrojový kód načíst a uložit, a teprve potom jej porovnat s ostatními uloženými kódy? Bylo by tak m.j. možné zpětně spouštět porovnávání již uložených zdrojových kódů.

- Práce popisuje (a zdůvodňuje), proč byl Hasher (hašovací modul) zahrnutý do modulů závislých na jazyce. Nešlo by - za cenu asi komplikovanějšího návrhu nějakého obecného AST - zahrnout hasher do modulů, nezávislých na jazyce. Výhodou by byla jen jedna implementace hashování a možnost hledat duplikáty i napříč jazyky, pokud by byla doimplementována podpora i pro ně.

Součástí předkládaného řešení jsou i experimenty. V experimentu 5.2 jsou pro rozhodování o shodném / různém autorovi použity parametry získané v experimentu 5.1, i když jsou zdrojové kódy více než třikrát delší. Nebyly by výsledky lepší, pokud by se použil jiný treshold pro počet shodných uzlů? Byly vyzkoušeny i jiné parametry?

**Práci doporučuji k obhajobě.**

**Práci nenavrhuji na zvláštní ocenění.**

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

**Datum** 3. září 2020

**Podpis**