

# Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

**Autor práce** Martin Zimen  
**Název práce** Statická analýza NumPy programů  
**Rok odevzdání** 2023  
**Studijní program** Informatika  
**Specializace** Systémové programování

**Autor posudku** Martin Blichá Oponent  
**Pracoviště** Katedra distribuovaných a spolehlivých systémů

## K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání		X	X	
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>		X	X	
<p>Cílem práce bylo vytvořit nástroj na detekci chyb plynoucích z nekompatibilních rozměrů matic v programech psaných v programovacím jazyce Python a využívajících knihovnu NumPy. Autor úspěšně navrhl a implementoval prototyp takového nástroje. Nicméně tento prototyp má značná omezení. Postrádá podporu jak pro některé konstrukce jazyka Python, tak pro funkce běžně využívané při práci s knihovnou NumPy. Vyhodnocení nástroje je provedeno pouze na malém množství jednoduchých testovacích příkladů a není jasné jak nástroj škáluje pro větší programy.</p> <p>Autor si zvolil náročné téma a bohužel celkově práce působí dojmem, že odevzdání bylo úspěšné. Text by hodně vylepšil formálnější přístup a samotná implementace by, dle mého názoru, měla podporovat víc konstruktů jak jazyka Python, tak především knihovny NumPy, na kterou se práce zaměřuje.</p>				

## Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>		X		
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>		X		
Analýza			X	
Vývojová dokumentace		X	X	
Uživatelská dokumentace			X	

Práce je psaná srozumitelnou angličtinou, což oceňuji. Struktura práce je standardní, ale jako vědecká práce obsahuje několik menších a větších problémů. Analýza problému popisuje několik alternativních přístupů, které autor zvažoval ale zamítnul. Avšak i zvolený přístup (kombinace data-flow analýzy a symbolické exekuce) trpí nevýhodami, které byly odhaleny až při pokusu o implementaci a pro které musel autor dodatečně zavést uměle působící úpravy algoritmu (především mám na mysli “general matrices”). V tradiční symbolické exekuci a podobných přístupech je chyba signalizována když může nastat pro nějaký vstup. V této práci naopak algoritmus identifikuje chybu jenom když nastane vždy, nezávisle na vstupech. Tato volba vede k výše uvedeným problémům, které si později uvědomil i autor (viz sekce 5.2.1 Validation vs Bug hunting).

Největší nedostatek spočívá v příliš neformálním popisu řešení. Tento popis často připomíná spíš tok myšlenek autora, což hodně stěžuje pochopení navrhovaného řešení. V textu úplně chybí formální definice používaných pojmů a pseudokód navrženého algoritmu, o který by se čtenář mohl opřít.

Mezi menší nedostatky patří například absence vysvětlení termínu “SMT řešič” nebo těžko čitelné obrázky, kde popisky hran a vrcholů grafu jsou překryty samotnými hranami.

I když je uživatelská dokumentace jednoduchá, což u nástroje příkazové řádky můžeme očekávat, postrádá základní informace jako výchozí hodnoty parametrů algoritmu určující počet přechodů různých fází nebo detailnější vysvětlení parametrů pro účely ladění.

### Implementační část práce

lepší    OK    horší    nevyhovuje

Kvalita návrhu    ... architektura, struktury a algoritmy, použité technologie		X		
Kvalita zpracování    ... jmenné konvence, formátování, komentáře, testování		X	X	
Stabilita implementace		X		

Implementace je s ohledem na navržený algoritmus přiměřená, program pracuje podle očekávání. S ohledem na složitost algoritmu bych očekával víc vysvětlujících a dokumentačních komentářů. Chybí jakákoliv podpora pro automatické testování nástroje. Jediné testy jsou ukázkové příklady připravené autorem, které ale vyžadují manuální spuštění nástroje a inspekci výsledku.

**Celkové hodnocení**    Dobře (spíše lepší)

**Práci navrhuji na zvláštní ocenění**    Ne

Datum

Podpis