



# Posudek oponenta závěrečné práce

Oponent práce: Ing. Jiří Daněček  
Student: Bc. Martin Bedrna  
Název práce: Interaktivní dokazovací asistent pro výuku formálních metod  
Obor / specializace: Softwarové inženýrství  
Vytvořeno dne: 6. února 2024

## Hodnotící kritéria

### 1. Splnění zadání

- [1] zadání splněno
- ▶ [2] zadání splněno s menšími výhradami
- [3] zadání splněno s většími výhradami
- [4] zadání nesplněno

Viz závěrečné zhodnocení

### 2. Písemná část práce 85 /100 (B)

Viz závěrečné zhodnocení.

### 3. Nepísemná část, přílohy 85 /100 (B)

Viz závěrečné zhodnocení.

### 4. Hodnocení výsledků, jejich využitelnost 95 /100 (A)

Vytvořená pomůcka bude jistě dobře využitelná při výuce.

## Celkové hodnocení 85 /100 (B)

Zadání předložené práce, která se zabývá návrhem a implementací výukového dotazovacího systému bylo splněno. V první části práce uvádí bakalář základní teoretické informace o teorii dokazování. Zde jsem postrádal nějaké novější poznatky, např. jak bylo matematické dokazování ovlivněno možnostmi automatického dokazování, nemluvě o možnostech AI v této oblasti. Teoretická část je zakončena velmi stručným popisem dokazovacího systému použitého v aplikaci, aniž by bylo vysvětleno, proč bylo vybráno právě toto schéma. Tato část měla být důkladně vyargumentována a podle mého názoru

uvedena až v sekci Analýza.

V analytické části práce bakalář podrobně a vyčerpávajícím způsobem zhodnotil dostupné aplikace s obdobným zaměřením. Výsledkem této části je dobře zdůvodněné rozhodnutí, které vlastnosti stávajících aplikací uplatnit ve vlastním návrhu.

V části Softwarový návrh použil bakalář klasické řešení popsáním funkčních a nefunkčních požadavků. Tato metodika byla vytvořena pro rozsáhlé softwarové systémy vyvíjené celými týmy. Popisovat každou operaci při dokazování jako samostatný funkční požadavek mi připadá pro tento druh aplikace formalistické a málo přínosné.

V další části bakalář dobře popisuje použité technologie a důvody proč si zvolil webový framework React. S čím se nedokážu ztotožnit je, že místo typově silného jazyka TypeScript použil JavaScript. I když chápu praktické důvody, které bakaláře k tomuto rozhodnutí vedly, tak argumentace, že „slabé typování umožňuje vynechat definici typů u neprimitivních objektů a využívat je pro flexibilní obsah prvků jednotlivých objektů“ jde zřejmě proti požadavku zadání: „Udělejte návrh celého softwarového balíku tak, aby se po skončení projektu dal snadno udržovat a rozšířit“.

Oceňuji, že bakalář vytvořil prototyp uživatelského rozhraní pomocí wire modelu.

V sekci Implementace popisuje bakalář vkládání logických formulí do systému. Ze softwarového hlediska jsou formule doménově specifický jazyk (DSL), který je navíc velmi podobný podmnožině obecných programovacích jazyků (GPL). Proto nechápu, proč jeho syntaxi a sémantiku nepopsal již v analýze a to standardním aparátem, tedy bezkontextovou gramatikou a sémantickými pravidly.

V sekci Testování pak bakalář poukazuje na to, že vytvořil především funkční testy a nikoliv test uživatelského rozhraní, „jelikož se jedná o část kódu, která je obvykle často měněna“. To je ovšem paradox, protože právě ta část aplikace, která je nejméně stabilní potřebuje nejvíce aparát automatického testování. V sekci uživatelského testování jsem postrádal více formalizovaný přístup, tedy např. jaké znalosti byly u uživatelů předpokládány, podle jakého scénáře měli postupovat atp. Celkově hodnotím práci jak velmi dobrou.

## **Instrukce**

### **Splnění zadání**

Posudte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posudte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.

### **Písemná část práce**

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posudte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti.

Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 52/2021, článek 3.

Posudte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

### **Nepísemná část, přílohy**

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů.

### **Hodnocení výsledků, jejich využitelnost**

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

### **Celkové hodnocení**

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.