



# Posudek oponenta závěrečné práce

**Student:** Irina Shushkova  
**Oponent práce:** Ing. Tomáš Pecka  
**Název práce:** Implementace automatových algoritmů na hledání pravidelností  
**Obor:** Teoretická informatika

**Datum vytvoření:** 6. 6. 2019

<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – následující škálou 1 až 4:</i>
<b>1. Splnění zadání</b>	<b><u>1=zadání splněno,</u> 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno</b>
<i>Popis kritéria:</i> Posuďte, 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, posuďte 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.	
<i>Komentář:</i> Cílem práce bylo představení a implementace algoritmů pro vyhledávání přesných a přibližných pravidelností.  Hodnotím práci jako průměrně náročnou, studentka musela sice nastudovat a naimplementovat netriviální algoritmy, ale v menším počtu.  Zadání je plně splněno.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):</i>
<b>2. Písemná část práce</b>	<b>80 (B)</b>
<i>Popis kritéria:</i> Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posuďte, 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. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3. Posuďte, 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.	
<i>Komentář:</i> Práce je na spodním limitu doporučeného rozsahu (31 stran). Kapitoly na sebe navazují a jsou věcné. Jazykově a typograficky je práce v pořádku, narazil jsem jen na občasné drobné nedostatky.  První kapitola je prakticky jen soupis definicí používaných dále v práci. Místo soupisu definicí bych občas raději viděl souvislý text (např. definice 1.3.20, 21 nebo u definic konečných automatů). Chybějící příklady nově definovaných pojmů také stěžují pochopení textu. V sekci 1.1 jsou zavedeny Levenshteinova a Damerau-Levenshteinova vzdálenost řetězců, přitom v sekci 2.2 se píše, že ačkoliv byly zavedeny, nebudeme se jimi zabývat. Přijde mi tedy zbytečné je definovat a zatěžovat tím čtenáře. Pokud je lze také použít pro algoritmy z kapitoly 2, pak by to stačilo jen zmínit. Definice 1.2.1 operuje s pojmy "vlastní prefix" a "vlastní suffix", které nejsou vysvětleny. V definici 1.3.4 trochu mate pojem stromový automat, ačkoliv práce je o řetězcích. V definici 1.3.5 je použita konfigurace automatu, která není předtím vysvětlena. Některé definice nad DKA a NKA jsou shodné a tak nevidím důvod je opakovat. Definice 1.3.18 by asi měla být zapsána jako algoritmus. Ve Značení 1.3.3 a 1.3.4 se vyskytují zbytečné dopředné reference.  V práci jsem nenašel uvedení asymptotické složitosti algoritmů z kapitoly 2.  Obrázek 3.2 znázorňuje implementované třídy v kompozici. V implementaci ale kompozice těchto tříd není.  V kapitole 2 jsou popisky obrázků nezvykle nad obrázkem. V kapitole 3 pak již pod obrázkem.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):</i>

### 3. Nepísemná část, přílohy

60 (D)

#### Popis kritéria:

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

#### Komentář:

Kód je otestovaný přiloženými unit testy a funkční.

K implementaci mám následující připomínky:

Některé parametry funkcí (hlavně v rekurzivní `ApproximateCoversComputation::processState`) jsou předávány kopií namísto referencí. Vytvoření kopie větších automatů, kterými automaty pro přibližné indexování mohou být, je náročné. Proto mají algoritmy značně větší dobu běhu. Dále, průchod `std::set` s mazáním je zvláštní, invalidace iterátoru po volání `erase` je vyřešena nastavením iterátoru na začátek namísto použití návratové hodnoty metody `erase`, což zbytečně přidává další kroky iterace. Použití `std::move` často nedává smysl v daném kontextu.

Zvláště první dva body hodnotím jako závažné, neboť mají velký vliv na dobu běhu algoritmu a i základní implementace by se měla vyvarovat takovýchto chyb. V práci je zmíněno, že doba běhu nebyla hlavní hledisko v práci, což je v souladu s filosofií ALT, neboť projekt se zaměřuje spíše na čitelnost algoritmů. Nicméně, po rychlé úpravě výtek (a použití efektivnějšího algoritmu na odstraňování epsilon přechodů) se čas běhu rapidně zmenšil.

Další poznámky:

- Některé fragmenty kódu jsou zbytečné, jako např. zjištění, zda je set prázdný (dokonce je namísto metody `empty` použito porovnání iterátorů začátku a konce) před tím, než se přidá do jiného setu. I kdyby byl prázdný, kód by fungoval.
- Code style není konzistentní. Zvláště v místech, kde se implementace kvůli možnostem jazyka liší od pseudokódu z textové části, by neškodilo několik komentářů.
- Některé průchody kontejnery pomocí `for-range` se dají nahradit funkcemi z moderního C++ (jako např. `std::any_of`, `std::accumulate`), což je dokonce připodobní jejich podobě v textové práci.

#### Hodnotící kritérium:

Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):

### 4. Hodnocení výsledků, jejich využitelnost

80 (B)

#### Popis kritéria:

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.

#### Komentář:

Kód bude po úpravách (viz předchozí sekce) začleněn do projektu ALT.

Naimplementované algoritmy jistě mohou být využity při výuce, obzvláště v předmětu MI-AVY.

#### Hodnotící kritérium:

Způsob hodnocení – nehodnotí se

### 5. Otázky k obhajobě

#### Popis kritéria:

Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

#### Otázky:

- Proč konstruujete nedeterministický k-přibližný suffixový automat pomocí epsilon přechodů, které poté odstraňujete? Online konstrukce NKA u těchto automatů není obtížná. Neušetřil by se tím čas?
- Jak pracné by bylo doimplementovat algoritmy pro přibližné pokrytí s použitím jiných vzdálenostních funkcí?

#### Hodnotící kritérium:

Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):

### 6. Celkové hodnocení

70 (C)

#### Popis kritéria:

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.

#### Text hodnocení:

Studentka nastudovala a naimplementovala tři algoritmy pro pokrývání řetězců do ALT. Textová část je celkem v pořádku. K implementaci mám výhrady, na druhou stranu, autorka se musela zorientovat v živě se vyvíjícím projektu. Vzhledem k tomu, že práce byla převážně implementační, ovlivnilo to mé hodnocení.

Práci doporučuji k obhajobě a hodnotím 70 body, tedy ještě známkou C (dobře).

Podpis oponenta práce: