Výuková aplikace pro automatické dokazování ve výrokové logice

Bakalářská práce

Vedoucí práce:

Ing. Pavel Haluza, Ph.D. Dominik Farlík

Poděkování

Zde je možné psát text poděkování. Pokud chcete poděkování vynechat, smažte tento text a u dvoustranného dokumentu i následující stránku.

Čestné prohlášení

Prohlašuji, že jsem práci Výuková aplikace pro automatické dokazování ve výrokové logice vypracoval/a samostatně a veškeré použité prameny a informace uvádím v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a v sou­ladu s platnou Směrnicí o zveřejňování závěrečných prací.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 12. května 2025

Abstract

Rybička, J. The document template for bachelor/diploma thesis. Bachelor thesis. Brno: Mendel University, 2019.

Example of document template for bachelor/diploma thesis is presented here. An abstract is in (British) English.

Keywords

Thesis, template, document, thesis example. Here are key words in (British) English.

Abstrakt

Rybička, J. Šablona pro závěrečné práce. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2019.

V textu je popsána šablona pro závěrečné práce a jednotlivé styly jsou aplikovány na příkladech.

Klíčová slova

Závěrečná práce, šablona, dokument, příklad závěrečné práce.

Obsah

1 Úvod a cíl práce 11

1.1 Úvod 11

1.2 Cíl práce 11

2 Výroková logika 12

2.1 Základní pojmy 12

2.1.1 Výrok 12

2.1.2 Logické spojky 12

2.1.3 Složené výroky 12

2.1.4 Tautologie a kontradikce 12

2.2 Konjuktivní normální forma 12

3 Automatické dokazování 13

3.1 Metody dokazování (rezoluční metoda, pravdivostní tabulky, …) 13

3.2 Algoritmus rezoluční metody 13

3.3 Použití v informatice a umělé inteligenci 13

4 Návrh a implementace aplikace 14

4.1 Backend 14

4.1.1 Tvorba API 14

4.1.2 Struktura přihlašování uživatelů 14

4.1.3 Algoritmus pro převod od KNF 14

4.1.4 Algoritmus rezoluční metody 14

4.2 Frontend 14

4.2.1 Uživatelské rozhraní 14

5 Diskuse a zhodnocení 15

5.1 Přínos aplikace 15

5.2 Omezení a možná vylepšení 15

5.3 Možnosti dalšího rozvoje 15

6 Závěr 16

6.1 Shrnutí výsledků 16

6.2 Možnost praktického využití 16

7 Literatura 17

7.1 Seznam bibliografických citací 17

Seznam obrázků

Seznam obrázků a případně seznam tabulek umisťujeme do dokumentu v případě, že to má pro čtenáře nějaký zvláštní význam. V tomto dokumentu nemá význam „seznam“ obrázků ani tabulek, protože v celém dokumentu je obrázek a tabulka jen jedna a takový seznam je spíše komickou vložkou než nějakou pomocí pro čtenáře.

Obr. 1 Náčrt lokomotivy řady 310.0 (příklad obrázku a jeho popisku) Zdroj: Atlas parních lokomotiv, 2013. 14

Seznam tabulek

[Tab. 1 Toto je popisek pro tabulku 13](#_Toc4176461)

1. Úvod a cíl práce
   1. Úvod
   2. Cíl práce
2. Analýza
   1. Aktuální stav

Důvod vzniku této aplikace je především to, že žádná taková s takovými parametry nebyla dosud vytvořena.

Z hlediska funkcionality, žádná aplikace nenabízí komplexní řešení, aby byla schopná, jak množiny výrokových klauzulí řešit, tak i uživateli poskytnout konkrétní postup jakým způsobem se k výsledku došlo.

* 1. Požadavky aplikace

V této sekci budou popsány základní dva druhy požadavků, které by měla aplikace splňovat. U funkčních požadavků se bude jednat zejména o to, jakým způsobem a jaké operace mohou uživatelé provádět a u nefunkčních spíše pozadí aplikace, jako je například spolehlivost nebo škálovatelnost aplikace.

* + 1. Funkční požadavky

Aplikace poskytne webové rozhraní, které umožní uživatelům zadávat logické formule v přehledné podobě, včetně intuitivního editoru a možnosti načíst předdefinované příklady. Rozhraní zobrazí průběh řešení krok za krokem. Vstupní data lze zadávat v infixové notaci, aplikace je automaticky převede do konjunktivní normální formy (KNF) a umožní jejich nahrání ze souboru. Implementace rezoluční metody zajistí rozhodování o splnitelnosti množiny formulí, zaznamenávání rezolučních kroků a jasné oznámení výsledku. Vizualizace bude zahrnovat stromový diagram rezolučních kroků, zpětné procházení jednotlivých fází řešení a možnost stažení protokolu ve formátu PDF nebo JSON. Aplikace bude dostupná online bez nutnosti instalace, kompatibilní s moderními prohlížeči a navržena s ohledem na budoucí rozšíření. Kód bude dobře dokumentovaný a otevřený pro další vývoj.

* + 1. Nefunkční požadavky

Aplikace by měla být optimalizovaná pro rychlé zpracování logických formulí a vizualizaci kroků algoritmu, aby poskytovala uživatelům přehledný a srozumitelný výstup. Důležitým aspektem je také přehledné uživatelské rozhraní, které umožní intuitivní ovládání a snadnou interakci se vstupními daty. Aplikace by měla být navržena modulárně a s ohledem na budoucí rozšíření, například podporu dalších metod dokazování. Dále je kladen důraz na multiplatformní přístup, což umožní nasazení nejen jako webovou aplikaci, ale potenciálně i jako mobilní verzi.

* 1. Podobné aplikace

Existují různé aplikace, které dokážou řešit splnitelnost logických formulí, dokonce i s možností výběru metody, která bude použita pro řešení, ale zatím neexistuje žádná, která by nabízela i vizualizaci postupu řešení.

<https://www.erpelstolz.at/gateway/formular-uk-zentral.html>

<https://www.emathhelp.net/calculators/discrete-mathematics/truth-table-calculator/>

<https://atozmath.com/MathLogic.aspx>

1. Automatické dokazování
   1. Metody dokazování (rezoluční metoda, pravdivostní tabulky, …)
   2. Algoritmus rezoluční metody
   3. Použití v informatice a umělé inteligenci
2. Návrh aplikace

V dnešní době je nespočet možností, jak navrhnout a implementovat aplikaci. Konkrétně třeba u výběru typu aplikace, vývojovém prostředí, ve kterém bude tvořena, programovacím jazyce, ve kterém bude napsána nebo různé technologie, které mohou být použity. Tato kapitola bude tedy zaměřena právě na výběr konkrétních možností.

* 1. Typ aplikace

U typu aplikace záleží hlavně na tom, jak a co má splňovat. Při tvorbě této aplikace, by se dalo zvolit více možností, třeba pro mobilní telefony, desktop nebo web. V mém případě jsem zvolil aplikaci pro web. Toto rozhodnutí jsem učinil z více důvodů, jako například lepší zobrazení pro uživatele PC z důvodu velkého množství zobrazovaných informací, využití pro uživatele mobilních zařízení i PC nebo výpočetní výkon na straně serveru.

* 1. Backend
     1. Flask framework
     2. Python
     3. Algoritmus pro převod od KNF
     4. Algoritmus rezoluční metody
  2. Frontend
     1. React
     2. Axios?
     3. Uživatelské rozhraní
  3. Databáze

1. Implementace

Architektura = obrazek = client side, server side

Struktura api callu - /api/solve, /login, …(tabulka)

Stejne jako navrh, ale s praktickyma ukazkama kodu, diagramu, templatama stranek, …

Api planning

Db = diagram, kus kodu s pozitim databaze

Struktura client side callu = -app/solve, app/login, …

1. Diskuse a zhodnocení
   1. Přínos aplikace
   2. Omezení a možná vylepšení
   3. Možnosti dalšího rozvoje
2. Závěr
   1. Shrnutí výsledků
   2. Možnost praktického využití
3. Literatura

Pro vytváření seznamu literatury podle normy ČSN ISO 690 (2011) se v této šabloně předpokládá použití nejjednoduššího systému odkazování – tzv. harvardského systému. V textu se uvede autor, rok a případná lokace v dokumentu, seznam literatury je seřazený podle abecedy (seřadíte ručně při vkládání jednotlivých položek). Použití různých automatismů vede obvykle k tomu, že dostaneme sice vygenerovaný seznam literatury, ale vůbec to není bez práce. Spíše naopak: Nejenže musíme stejně jednotlivé zdroje zapisovat do vstupních formulářů, ale navíc nelze takto vygenerovaný seznam jakkoliv ovládat a stejně tak nelze dostatečně ovládat i odkazy v textu. Vzhledem k tomu, že harvardský systém odkazování žádnou automatiku téměř nepotřebuje a je jak pro autora, tak i pro čtenáře nejpříjemnější variantou, byl tento systém zcela zřejmou a jasnou volbou pro tuto šablonu.

Po technické stránce jsou položky literatury formátovány stylem ZP: Literatura. V citačních informacích je jméno autora formátováno znakovým stylem ZP: Literatura – jméno autora a název zdrojového dokumentu znakovým stylem ZP: Literatura – název díla. Jméno autora nikdy nevkládejte velkými písmeny. Pokud jej chcete zobrazovat velkými písmeny, upravte styl ZP: Literatura – jméno autora tak, aby zobrazoval všechna písmena velká. V přednastaveném stylu jsou pro jméno autora zvoleny kapitálky, což je pravděpodobně optimální volba. Pro ni však někteří lidé z neznámých důvodů nemají pochopení a domnívají se, že autory je třeba psát velkými písmeny.

Více podrobností o zpracování textů, a tedy i odborných a závěrečných prací lze nalézt v učebnici Rybičky, Čačkové a Přichystala (2011).

* 1. Seznam bibliografických citací