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

Farlík, D. Educational Application for Automated Theorem Proving in Propositional Logic.  
Bachelor Thesis. Brno: Mendel University, 2025.

This bachelor's thesis focuses on the design and implementation of an educational application aimed at using the resolution method for proving the truth of propositional formulas. The application will help users better understand the process of formula decomposition through the visualization of individual steps.   
The theoretical part explains the fundamental concepts related to propositional logic and the use of the resolution method. It also includes an analysis of the application's requirements and currently available alternatives.   
The practical part is dedicated to the application's design and the selection of technologies used for its development. Additionally, it covers the actual implementation and user testing.

Keywords

Propositional logic, resolution method, web application, Flask, React, Bootstrap, MySQL

Abstrakt

Farlík, D. Výuková aplikace pro automatické dokazování ve výrokové logice.

Bakalářská práce. Brno: Mendelova univerzita, 2025.

Bakalářská práce je zaměřena na návrh a implementaci výukové aplikace zaměřené na používání rezoluční metody pro dokazování pravdivosti výrokových formulí. Aplikace bude sloužit pro lepší pochopení procesu rozkladu formulí díky vizualizaci jednotlivých kroků postupu.

V teoretické části jsou vysvětleny základní pojmy problematiky okolo výrokové logiky a použití rezoluční metody. Součástí je i analýza požadavků aplikace a aktuálně dostupné alternativy.

Praktická část se věnuje návrhu aplikace a výběru použitých technologií pro vytvoření. Dále se v ní také vyskytuje samotná implementace a uživatelské testování.

Klíčová slova

Výroková logika, rezoluční metoda, webová aplikace, Flask, React, Bootstrap, MySQL

Obsah

Obsah 8

Seznam obrázků 10

Seznam tabulek 11

1 Úvod a cíl práce 13

1.1 Úvod 13

1.2 Cíl práce 13

2 Analýza 14

2.1 Aktuální stav 14

2.2 Požadavky aplikace 14

2.2.1 Funkční požadavky 14

2.2.2 Nefunkční požadavky 14

2.3 Podobné aplikace 15

3 Automatické dokazování 16

3.1 Metody dokazování 16

3.1.1 Rezoluční metoda 16

3.1.2 Pravdivostní tabulka 17

3.2 Použití v informatice a umělé inteligenci 17

4 Návrh aplikace 18

4.1 Typ aplikace 18

4.2 Backend 18

4.2.1 Python 18

4.2.2 Flask 18

4.2.3 API 18

4.3 Frontend 19

4.3.1 Javascript 19

4.3.2 React 19

4.3.3 Bootstrap 19

4.3.4 Uživatelské rozhraní 19

4.4 Databáze 19

4.5 Bezpečnost 19

4.5.1 API Komunikace 19

4.5.2 Komunikace uživatele se serverem 19

5 Implementace 20

6 Diskuse a zhodnocení 21

6.1 Přínos aplikace 21

6.2 Omezení a možná vylepšení 21

6.3 Možnosti dalšího rozvoje 21

7 Závěr 22

7.1 Shrnutí výsledků 22

7.2 Možnost praktického využití 22

8 Literatura 23

8.1 Seznam bibliografických citací 23

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>

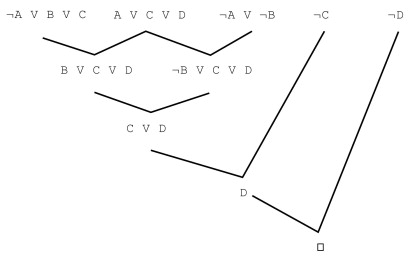
1. Automatické dokazování

Automatizované dokazování vět je zásadní pro různé aspekty (zejména praktické) informatiky. Obvykle se implementuje pomocí jednoduchých propozičních důkazových systémů. Objevování obtížných tautologií vrhá světlo na možnosti a užitečnost různých technik automatizovaného dokazování(Ben-Sasson and Wigderson, 2001).

Klíčovým prvkem většiny systémů pro dokazování vět je unifikační algoritmus, který pro danou dvojici vstupních výrazů vrací množinu unifikačních substitucí – přiřazení proměnným těchto výrazů, která je činí identickými nebo ekvivalentními. Typickým příkladem je unifikační algoritmus Robinsona [13] pro unifikaci atomických formulí v rámci dokazování vět metodou resoluce v predikátové logice prvního řádu [1, 6, 13](Stickel, 1981).

* 1. Metody dokazování
     1. Rezoluční metoda

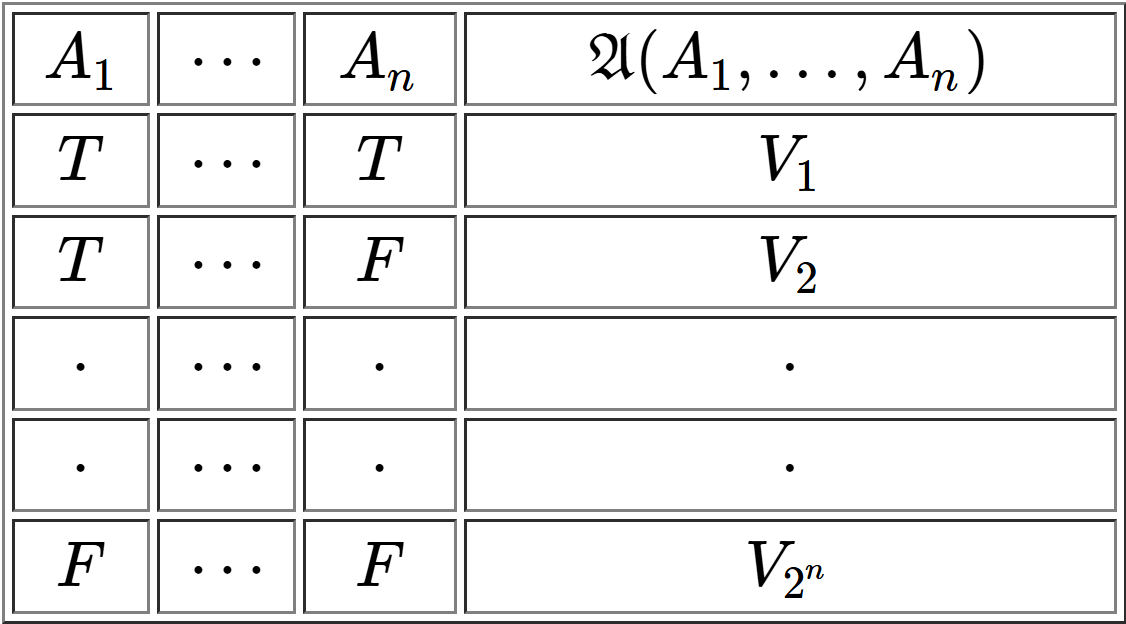
Důkazový systém Resolution, na který se tento článek zaměřuje, je možná tím nejjednodušším netriviálním systémem. Všechny tvrzení v tomto důkazovém systému jsou klauzule (tedy disjunkce literálů). Tautologie je reprezentována svou negací – jako množina protikladných klauzulí. (To je vždy možné díky NP-úplnosti problému SAT.) Důkaz (nebo vyvrácení) využívá jednoduché dedukční pravidlo k odvozování dalších klauzulí z těchto „axiomových klauzulí“, dokud není dosaženo kontradikce ve formě prázdné (nepravdivé) klauzule. Resolution tvoří základ mnoha automatizovaných procedur dokazování vět, které se v praxi používají(Ben-Sasson and Wigderson, 2001).



Obr. Diagram postupu rezoluční metody

* + 1. Pravdivostní tabulka

Pravdivostní tabulka je tabulka vyjadřující pravdivostní hodnoty složené výrokové formule na základě pravdivostních hodnot jednoduchých výroků, které ji tvoří (viz Pravdivostní hodnota). Pravdivostní tabulka má podobu tabulky níže, kde T označuje „pravda“ a F označuje „nepravda“.  
V tabulce Obr. 2 jsou A1, …, An​ výrokové proměnné, A(A1, …, An) je výroková formule a její pravdivostní hodnota je určena pravdivostními hodnotami A1, …, An. Každý řádek tabulky odpovídá jedné z 2n možných kombinací pravdivostních hodnot výrokových proměnných. Dále, *vi* je pravdivostní hodnota A(A1, …, An), pokud mají   
A1, …, An ​ pravdivostní hodnoty uvedené v *i*-tém řádku(*Truth table - Encyclopedia of Mathematics*, no date).



Obr. 2 Pravdivostní tabulka (Truth table - Encyclopedia of Mathematics, no date)

* 1. Použití v informatice a umělé inteligenci

1. 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

V odvětví vývoje software termín „backend“ označuje část aplikace, kterou běžný uživatel nevidí. Slouží pro práci s daty a řeší logiku dané aplikace, jako například autentizaci uživatele nebo práci s databází. Tato kapitola se věnuje výběru konkrétních technologií použitých pro tvorbu aplikace.

* + 1. Python

Pro backend aplikace je použit programovací jazyk python. Je to interpretovaný, objektově orientovaný jazyk. Zahrnuje moduly, výjimky, dynamické datové typy a třídy (*Python 3.13.2 documentation*, no date). I přes to, že python není nejrychlejší programovací jazyk, tak pro potřeby této aplikace je plně dostačující. Zde zvolen, jelikož nabízí různé frameworky, které pokrývají všechny veškeré požadavky aplikace. Výhodou je také, jednoduchá práce databázemi, API, ale i samotným vývojem algoritmu pro řešení logických formulí.

* + 1. Flask

Flask je lehký WSGI framework pro webové aplikace. Je navržen tak, aby umožňoval rychlý a snadný začátek, s možností rozšíření na složité aplikace (*Flask Documentation (3.1.x)*, no date). Flask je použit z důvodu jeho jednoduchosti a flexibilitě, také nabízí různé knihovny pro práci s databázemi nebo řešením autentizace, což mnohdy usnadní vývoj.

* + 1. API
  1. Frontend
     1. Javascript
     2. React
     3. Bootstrap
     4. Uživatelské rozhraní
  2. Databáze
  3. Bezpečnost
     1. API Komunikace
     2. Komunikace uživatele se serverem

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

Ben-Sasson, E. and Wigderson, A. (2001) ‘Short proofs are narrow—resolution made simple’, *Journal of the ACM*, 48(2), pp. 149–169. Available at: https://doi.org/10.1145/375827.375835.

*Flask Documentation (3.1.x)* (no date). Available at: https://flask.palletsprojects.com/en/stable/ (Accessed: 31 March 2025).

*Python 3.13.2 documentation* (no date). Available at: https://docs.python.org/3/tutorial/index.html (Accessed: 31 March 2025).

Stickel, M.E. (1981) ‘A Unification Algorithm for Associative-Commutative Functions’, *Journal of the ACM*, 28(3), pp. 423–434. Available at: https://doi.org/10.1145/322261.322262.

*Truth table - Encyclopedia of Mathematics* (no date). Available at: https://encyclopediaofmath.org/index.php?title=Truth\_table (Accessed: 30 March 2025).

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