

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

DIPLOMOVÁ PRÁCE

Simulátor konečných automatů



2026

Vedoucí práce:
doc. RNDr. Miroslav Kolařík,
Ph.D.

Bc. Jiří Mlčoušek

Studijní program: Aplikovaná informatika,
Specializace: Vývoj software

Bibliografické údaje

Autor: Bc. Jiří Mlčoušek
Název práce: Simulátor konečných automatů
Typ práce: diplomová práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2026
Studijní program: Aplikovaná informatika, Specializace: Vývoj software
Vedoucí práce: doc. RNDr. Miroslav Kolařík, Ph.D.
Počet stran: 13
Přílohy: elektronická data v úložišti katedry informatiky
Jazyk práce: český

Bibliographic info

Author: Bc. Jiří Mlčoušek
Title: Finite automaton simulator
Thesis type: master thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2026
Study program: Applied Computer Science, Specialization: Software Development
Supervisor: doc. RNDr. Miroslav Kolařík, Ph.D.
Page count: 13
Supplements: electronic data in the storage of department of computer science
Thesis language: Czech

Anotace

Tato diplomová práce se zabývá návrhem a implementací interaktivního simulátoru a vizualizátoru modelů z teorie formálních jazyků. Teoretická část práce podrobně mapuje Chomského hierarchii s důrazem na konečné automaty, zásobníkové automaty a Turingovy stroje. Práce dále popisuje klíčové algoritmy pro transformaci a minimalizaci těchto modelů, které tvoří jádro realizované aplikace určené pro podporu výuky teoretické informatiky.

Synopsis

This master thesis focuses on the design and implementation of an interactive simulator and visualizer for models in formal language theory. The theoretical part provides a detailed map of the Chomsky hierarchy, emphasizing finite automata, pushdown automata, and Turing machines. Furthermore, the thesis describes key algorithms for transformation and minimization of these models, which form the core of the implemented application designed to support the education of theoretical computer science.

Klíčová slova: formální jazyky; konečné automaty; Turingovy stroje; vizualizace; simulace; algoritmy; Chomského hierarchie

Keywords: formal languages; finite automata; Turing machines; visualization; simulation; algorithms; Chomsky hierarchy

Rád bych poděkoval vedoucímu práce doc. RNDr. Miroslavu Kolaříkovi, Ph.D. za cenné rady, trpělivost a odborné vedení během zpracování této diplomové práce. Dále děkuji své rodině a blízkým za podporu při studiu.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1	Úvod	7
2	Teoretický základ	7
2.1	Formální jazyky a Chomského hierarchie	7
2.2	Konečné automaty (DFA a NFA)	7
2.3	Determinizace a minimalizace	7
2.4	Regulární výrazy a jejich převod	8
2.5	Bezkontextové jazyky a zásobník	8
2.6	Turingovy stroje	8
3	Návrh systému	8
4	Implementace	8
5	Testování a experimenty	8
	Závěr	9
	Conclusions	10
A	První příloha	11
B	Druhá příloha	11
C	Obsah elektronických dat	11
	Seznam zkratk	13

Seznam tabulek

1 Úvod

Teorie formálních jazyků a automatů tvoří teoretický fundament moderní informatiky. Porozumění těmto konceptům je nezbytné pro návrh překladačů, analýzu složitosti algoritmů i pro pochopení limitů samotné vyčíslitelnosti. Přestože je tato oblast primárně matematická, její praktická aplikace v softwarovém inženýrství je nepopíratelná.

Motivací pro tuto diplomovou práci je vytvořit moderní, interaktivní nástroj, který by studentům a zájemcům o teoretickou informatiku umožnil vizualizovat abstraktní modely, jako jsou konečné automaty či Turingovy stroje. Tato práce si klade za cíl nejen popsat teoretické základy, ale také implementovat algoritmy, které tyto modely transformují, minimalizují a simulují v reálném čase.

2 Teoretický základ

Tato kapitola shrnuje matematický aparát nezbytný pro pochopení problematiky formálních jazyků. Text čerpá primárně zavedenou terminologii z publikací [cerna2002] a [jancar2007].

2.1 Formální jazyky a Chomského hierarchie

Základem teorie je definice abecedy Σ jako konečné množiny symbolů. Slovo je konečná posloupnost těchto symbolů a jazyk je libovolná množina slov.

Klíčovým konceptem je Chomského hierarchie, která dělí jazyky do čtyř tříd podle jejich generativní síly:

1. **Typ 0 (Rekurzivně spočetné):** Jazyky přijímané Turingovými stroji.
2. **Typ 1 (Kontextové):** Jazyky přijímané lineárně omezenými automaty.
3. **Typ 2 (Bezkontextové):** Jazyky popsané bezkontextovými gramatikami a přijímané zásobníkovými automaty.
4. **Typ 3 (Regulární):** Nejslabší třída, přijímaná konečnými automaty.

2.2 Konečné automaty (DFA a NFA)

Konečný automat je nejjednodušším modelem s konečnou pamětí. Formálně definujeme deterministický konečný automat (DFA) jako pětici $M = (Q, \Sigma, \delta, q_0, F)$.

2.3 Determinizace a minimalizace

Nedeterministické automaty (NFA) umožňují více přechodů pro jeden symbol. Pro účely simulace je však často nutná jejich determinizace pomocí tzv. *podmnožinové konstrukce*. Po determinizaci je dalším krokem minimalizace automatu,

která vede na kanonický tvar s nejmenším možným počtem stavů při zachování přijímaného jazyka [balun2025].

2.4 Regulární výrazy a jejich převod

Regulární výrazy poskytují syntakticky úsporný způsob popisu regulárních jazyků. V implementované aplikaci je využíván **Thompsonův algoritmus** pro převod regulárního výrazu na NFA s ε -kroky, což umožňuje vizualizaci postupu od textového zápisu ke grafové reprezentaci.

2.5 Bezkontextové jazyky a zásobník

Pro jazyky typu 2 (např. vyvážené závorky) již konečný automat nestačí. Je nutné zavést zásobníkový automat (PDA), který rozšiřuje konečný automat o paměť typu LIFO. V práci se zaměříme na simulaci práce se zásobníkovou abecedou Γ .

2.6 Turingovy stroje

Turingův stroj (TM) představuje univerzální model výpočtu. Skládá se z řídicí jednotky, nekonečné pásky a čtecí hlavy. V simulátoru je kladen důraz na vizualizaci konfigurace stroje, která je definována jako trojice (stav, obsah pásky, pozice hlavy) [jancar2007].

3 Návrh systému

- Požadavky na systém. - Uživatelské rozhraní a jeho funkce. - Grafická reprezentace automatů. - Struktura dat pro uchování automatů.

4 Implementace

- Použité technologie. - Realizace interaktivního editoru automatů. - Implementace krokového režimu. - Ukládání a načítání automatů. - Minimalizace automatů. - Generování automatů z regulárních výrazů.

5 Testování a experimenty

- Testovací scénáře. - Výsledky testování. - Srovnání s jinými existujícími nástroji.

Závěr

Závěr práce v „českém“ jazyce.

Conclusions

Thesis conclusions in “English”.

A První příloha

Text první přílohy

B Druhá příloha

Text druhé přílohy

C Obsah elektronických dat

Na samotném konci textu práce je uveden stručný popis obsahu elektronických dat odevzdaných v systému katedry informatiky spolu s textem. Tato data jsou nedílnou součástí práce a tvoří (datovou) přílohu textu práce. Povinné položky struktury dat jsou:

text/

Adresář s textem práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech (textových) příloh, a všechny soubory potřebné pro bezproblémové vytvoření PDF dokumentu textu (případně v ZIP archivu), tj. zdrojový text textu a příloh, vložené obrázky, apod.

README.*

Textový soubor (s příponou např. .txt) s informacemi o opakovatelném způsobu použití ostatních dat práce – typicky plně reprodukovatelný co nejúplnější funkční postup zprovoznění software vytvořeného v rámci práce, tzn. jeho případné instalace/nasazení a spuštění, včetně uvedení všech požadavků pro bezproblémový provoz; za zprovoznění software se nepovažuje zpřístupnění (např. po Internetu) již někde zprovozněného software.

Adresáře a soubory s veškerými ostatními autorskými daty práce (případně v ZIP archivu) – typicky spustitelné a další soubory software vytvořeného v rámci práce potřebné pro bezproblémový provoz software, případně jeho instalační program, a kompletní zdrojové texty software a další data nutná pro plně reprodukovatelné korektní vytvoření spustitelných souborů.

Dále mohou data obsahovat například:

- ukázková a testovací data použitá v práci nebo pro potřeby posouzení práce v rámci její obhajoby,
- položky bibliografie v elektronické podobě, příp. jiná relevantní literatura a dokumentace vztahující se k práci,

- cizí data (software) potřebná pro bezproblémové použití autorských dat práce (software), která nejsou standardní součástí předpokládaného (softwareového) vybavení uživatele.

U veškerých cizích obsažených materiálů jejich zahrnutí dovolují podmínky pro jejich veřejné šíření nebo přiložený souhlas držitele práv k užití. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy, je uveden jejich zdroj, např. webová adresa, v bibliografii nebo textu práce nebo souboru README.*.

