

System pre podporu práce s automatmi

Autor: Peter Bažík

Školiteľ: prof. RNDr. Branislav Rován, PhD.

Dôležitosť teórie formálnych jazykov

Zložitosť formálnych zápisov pre začiatočníkov

Skúmané existujúce systémy

Nevýhody skúmaných systémov, metódy vysvetľovania

Zhrnutie výsledkov práce

System – užívateľské rozhranie, moduly

Reprezentácia automatov, stavov, δ -funkcie, prázdneho slova

Knížnica *libfsm* ako integrovaný modul

Spôsob vysvetľovania algoritmov

Jazyk pre vstup a výstup

Jednoduchá extenzibilita

Triedy reprezentácie automatov: *FSM*, *DFA*, *NFA*

Konstrukčné algoritmy:

Zjednotenie, prienik, reverz, doplnok, zretazenie

Odepsilonovanie, determinizácia, „prasiatkový normálny tvar“

Odstránenie nedosiahnuteľných stavov

- *UnaryOperation*, *BinaryOperation*
- *Factory method* – typ operácie

Import a export vo formáte *Json* – pomocou *Gson*

Horizontálne:

- Výpočet
- Operácie
- Normálne tvary
- Ekvivalencia

Vertikálne:

- Špeciálne operácie – výber typu operácie
- Výber typu automatu, príklady, zo súboru
- Zadanie vstupného automatu, slova, typu konštrukcie
- Beh výpočtu alebo konštrukcie

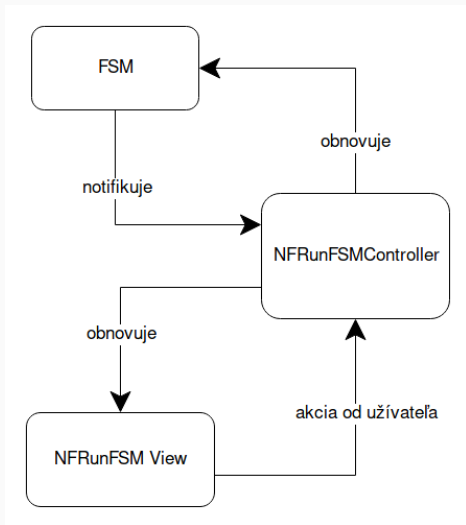
Model – View – Controller:

- *Model* – automat
- *View* – prechodový diagram, log
- *Controller* – manipuluje s automatom, obnovuje *View*

Mediator:

- Controllery v rámci sekcie – každý vo vertikálnom členení
- Predídenie úzkemu prepojeniu

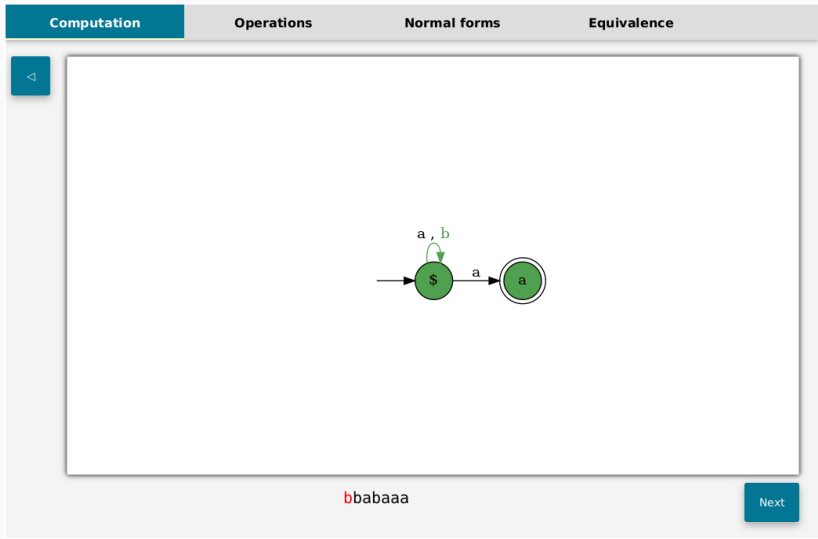
Kolobeh vykonávania algoritmu pomocou MVC



Obr. 1: Použitie MVC pri behu algoritmu

Farebné vyznačenie na diagrame

Prvý symbol neprečítanej časti slova

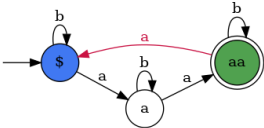
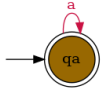



Obr. 2: Realizácia výpočtu na slove

Farebné vyznačenie na diagramoch

Detailný slovný popis konštrukcií

Metódy vysvetľovania – konštrukcie

Computation	Operations	Normal forms	Equivalence
	 <p>a,b</p>		 <p>a</p>
		 <p>a</p>	
			<pre>Added state [aa, qa] Added state [a, qa] 3. In this step we add transitions to result automaton in following way: delta([q_x, q_y], c) <- [delta_first(q_x, c), delta_second(q_y, c)] Added transition from: [\$, qa] on 'a' to: [a, qa] Added transition from: [aa, qa] on 'a' to: [\$, qa]</pre> <p>Next</p>

Obr. 3: Realizácia konštrukcie prieniku dvoch jazykov

Pomocou deterministických konečných automatov zostrojíme deterministický konečný automat pre prienik jazykov:

- $L_1 = \{w \in \{a, b\}^* \mid \#_a(w) \equiv 2 \pmod{3}\}$
- $L_2 = \{a\}^*$

- Implementovaný systém a knižnica *libfsm*
- Vizualizácia a vysvetlenie
- Možné pokračovanie práce:
 - Ďalšie algoritmy
 - Iné výpočtové modely regulárnych jazykov
 - Nadstavba s možnosťou voľby triedy jazykov
- <https://gitlab.com/pbazik/guimata>