

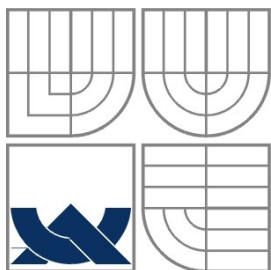
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

Fakulta informačních technologií  
Faculty of Information Technology

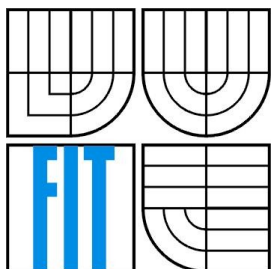
BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

Brno, 2016

Karel Krajíček



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# UNIVERZÁLNÝ SIMULÁTOR AUTOMATOV

A UNIVERSAL AUTOMATA SIMULATOR

BAKALÁRSKA PRÁCA  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

KAREL KRAJÍČEK

VEDÚCI PRÁCE  
SUPERVISOR

Prof. RNDr. ALEXANDER MEDUNA, CSc

BRNO 2016

## **Abstrakt**

Táto práca je o simulátore automatov, v ktorom si užívateľ môže vyskúšať aké to je pracovať s rôznymi typmi automatov a Turingovými strojmi. Bol vytvorený v C++ za pomoci grafickej knižnice SDL program, v ktorom užívateľ môže vytvoriť Turingov stroj a následne simulovať jeho funkciu.

## **Abstract**

This paper is about simulator of automata, where user can work with different types of automata and Turing machines. For that was created program in C++ with graphics library SDL, where user can create Turing machine and simulate it.

## **Kľúčové slová**

Turingov stroj, Konečné automaty, Zásobníkové automaty, Simulátor automatov, C++

## **Keywords**

Turing machine, Finite-state automaton, Pushdown automaton, Automata simulator, C++

## **Citace**

Karel Krajíček: Univerzálny simulátor automatov, bakalárska práca, Brno, FIT VUT v Brně, 2016

# Univerzálny simulátor automatov

## Prehlásenie

Prehlasujem že som túto bakalársku prácu vypracoval samostatne pod vedením pána prof. RNDr. Alexandra Meduny, CSc.

Uviedol som všetky literárne pramene z ktorých som čerpal.

.....  
Karel Krajíček  
16.5.2016

## Pod'akovanie

Rád by som poďakoval vedúcemu svojej práce, prof. RNDr. Alexandru Medunovi, CSc. za venovaný čas a pomoc pri riešení tejto práce.

© Karel Krajíček, 2016

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Obsah.....	1
1 <a href="#">Úvod</a> .....	2
2 <a href="#">Definície</a> .....	4
2.1 <a href="#">Konečný automat</a> .....	5
2.2 <a href="#">Obojsmerný konečný automat</a> .....	6
2.3 <a href="#">Deterministický zásobníkový automat</a> .....	7
2.4 <a href="#">Nedeterministický zásobníkový automat</a> .....	8
2.5 <a href="#">Turingov stroj</a> .....	9
2.6 <a href="#">Lineárne ohraničený Turingov stroj</a> .....	10
2.7 <a href="#">Vždy zastavujúci Turingov stroj</a> .....	10
2.8 <a href="#">Turingov stroj s viacerými páskami</a> .....	10
3 <a href="#">Popis programu</a> .....	11
3.1 <a href="#">Užívateľské rozhranie</a> .....	12
3.2 <a href="#">Príklady a testovanie</a> .....	17
3.2.1 <a href="#">Test 1</a> .....	17
3.2.2 <a href="#">Test 2</a> .....	20
3.2.3 <a href="#">Test 3</a> .....	23
3.2.4 <a href="#">Test 4</a> .....	25
3.2.5 <a href="#">Test 5</a> .....	27
4 <a href="#">Záver</a> .....	29
A <a href="#">Obsah CD</a> .....	32
B <a href="#">Manuál</a> .....	33

# 1 Úvod

Táto technická správa v prvej časti popisuje teóriu automatov a Turingových strojov a v druhej časti popisuje funkciu samotného programu.

Mojím cieľom je vytvoriť v C++ za pomoci grafickej knižnice SDL program, v ktorom užívateľ môže vytvoriť Turingov stroj a následne simulovať jeho funkciu. Vytvorením Turingovho stroja, myslím vytvorenie jeho prechodovej funkcie.

Formálne sa však nemôže jednáť úplný lineárne neohraničený Turingov stroj ktorý priama jazyky z množiny rekurzívne vyčísliteľných jazykov, pretože simulácia bude prebiehať na obyčajnom počítači s obmedzenou pamäťou. Takže najviac môžem vytvoriť lineárne ohraničený Turingov stroj ktorý priama jazyky z množiny kontextových jazykov.

Ja som sa ale rozhodol že svoj simulátor nebudem úmyselne obmedzovať a aj keď sa nebude jednáť o skutočný lineárne neohraničený Turingov stroj budem sa k nemu v texte ďalej odkazovať ako k Turingovu stroju.

Simulácia by mala fungovať tak že užívateľ vidí pásku, čítaciu/písaciu hlavu a stavy a vidí ako hlava pracuje nad páskou v každom kroku. Konečné automaty a zásobníkové automaty sú iba špeciálne prípady Turingových strojov, to znamená že každý konečný automat a zásobníkový automat sa dá implementovať aj ako Turingov stroj. Vytváranie stavov by malo fungovať ako vytváranie prechodových grafov .

Štruktúra programu ktorú som zvolil je objektovo orientovaná kde Turingov stroj je trieda ktorá obsahuje ako atribúty všetko čo formálne Turingov stroj obsahovať má a to množina stavov, vstupná abeceda ( abeceda symbolov ktoré sa môžu nachádzať na vstupe ), pásková abeceda ( abeceda všetkých symbolov ktoré sa môžu objaviť na páske ), symbol označujúci koniec pásky, prechodová funkcia, počiatočný stav, prijímací stav a zamietací stav.

Implementácia Turingovho stroja je objekt tejto triedy. Všetky vytvorené implementácie budú zapísané v textovom súbore. Rozhodol som sa že bude lepšie ak budem zapisovať zvlášť implementácie Turingových strojov a pásky, už len hlavne pre to že jeden Turingov stroj môže v inom čase pracovať nad viacerými páskami. Tým ale nehovorím o viac páskových Turingových strojoch ktoré pracujú nad viacerými páskami v tom istom čase. Pásky pre viac páskové Turingové stroje nad ktorými prebehne ten istý výpočet budú uložené v jednom súbore.

Implementácie Turingových strojov rozlišujú či sa jedná o nejaký špeciálny prípad Turingových strojov, ako sú konečné automaty, zásobníkové automaty, Turingove stroje s viacerými páskami. Rozlišuje sa to podľa prvého riadku v súbore.

Formálne je prechod v Turingovom stroji  $(Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{-,+,0\})$  čo bude v textovom súbore uložené ako päť symbolov  $q_1 t_1 q_2 t_2 s$  kde  $q_1, q_2 \in Q$ ,  $t_1, t_2 \in \Gamma$ ,  $s \in \{-,+,0\}$

kde  $Q$  je množina stavov,  $\Gamma$  je pásková abeceda,  $-$  je smer pohybu hlavy vľavo  $+$  je smer pohybu hlavy vpravo a  $0$  je stáť.

Štandardne je páska ukončená na jednom konci a nekonečná na druhom. Pre tento program ale budem uvažovať prípad že páska je nekonečná na oboch koncoch a symbol  $\$$  slúži na detekovanie prázdneho poľa.

Text ktorý bude na páske je kontrolovaný abecedou na páske ktorá je uvedená súbore s kódom implementácie Turingovho stroja.

## 2 Definície

Abeceda  $\Sigma$  je neprázdna konečná množina prvkov, ktoré sa nazývajú symboly.

Konečná sekvencia symbolov abecedy je reťazec nad abecedou  $\Sigma$ .

$\varepsilon$  je prázdny reťazec.

$\sim$  je označenie pre ľubovoľný symbol. Bude sa využívať v prechodových funkciách pri porovnávaní symbolov na páske, kde porovnanie ekvivalencie symbolu na páske so symbolom  $\sim$ , vždy bude true.

$|x|$  je dĺžka reťazca  $x$ , teda počet symbolov v reťazci  $x$

Konkatenácia reťazcov  $x$  a  $y$  je  $z$  kde  $z = xy$

$x^i$  je  $i$ -ta mocnina slova  $x$  kde:

$$x^0 = \varepsilon$$

$$x^i = xx^{i-1}$$

Prefixom reťazca  $x$  je  $y$  pokiaľ  $x = yz$

Suffixom reťazca  $x$  je  $z$  pokiaľ  $x = yz$

Podreťazec reťazca  $x$  je  $y$  pokiaľ  $x = yxz$

$\Sigma^*$  je množina všetkých slov nad abecedou  $\Sigma$ .

$\Sigma^+$  je  $\Sigma^* - \{\varepsilon\}$

Jazyk nad abecedou  $\Sigma$  je podmnožina  $\Sigma^*$

Doplnok jazyka:

$L'$  je doplnok jazyka  $L$  nad abecedou  $\Sigma$  sú všetky slová nad abecedou  $\Sigma$  ktoré nie sú v jazyku  $L$

$$L' = \{w \in \Sigma^* \mid w \notin L\}$$

Konkatenácia jazyka:

$$L^1L^2 = \{ab \mid a \in L^1, b \in L^2\}$$

Mocnina jazyka:

$L^i$  je  $i$ -ta mocnina jazyka  $L$  kde:

$$L^0 = \{\varepsilon\}$$

$$L^i = LL^{i-1}$$

Tieto definície boli prebraté z [1] a [2]



## 2.1 Konečný automat

Konečný automat načíta vstupné symboly zo vstupu, ktorý môže byť reprezentovaný páskou s čítacou hlavou ktorá sa môže pohybovať iba v jednom smere od začiatku pásky. Na základe vstupného symbolu a stavu v ktorom sa automat nachádza, automat zmení svoj stav. Tento krok je definovaný prechodovou funkciou. Konečný automat takto môže prijímať alebo zamietat' slová patriace do konkrétneho jazyka. Takýto jazyk patrí do množiny regulárnych jazykov. Automat môže byť deterministický alebo nedeterministický, pričom obe možnosti majú tú istú výpočetnú silu. Nedeterministický automat a v tomto prípade znamená, že prechodová funkcia má na ten istý vstupný symbol a stav viac možností ako zmeniť svoj stav. Deterministický automat má na ten istý vstupný symbol a stav najviac jednu možnosť ako zmeniť svoj stav.

Formálna definícia:

Konečný automat  $M$  je  $M = (Q, \Sigma, \delta, s, F)$

kde:

- $Q$  množina stavov
- $\Sigma$  je vstupná abeceda
- $\delta$  je prechodová funkcia  $Q \times \Sigma \rightarrow Q$
- kde  $Q \times \Sigma$  je  $\{ (q, a) \mid q \in Q, a \in \Sigma \}$   
čo znamená, že prechodová funkcia rozhodne do ktorého stavu automat prejde na základe symbolu na vstupe a aktuálneho stavu
- $s \in Q$  je počiatkový stav
- $F$  je množina prijímacích stavov alebo konečných stavov,  $F$  je podmnožina  $Q$

$M$  akceptuje reťazec  $w \in \Delta^*$  ak  $sw \rightarrow^* f, f \in F$

formálne:  $L(M) = \{ w \mid w \in \Delta^*, sw \rightarrow^* f, f \in F \}$

Táto definícia bola prebratá z [2]

## 2.2 Obojsmerný konečný automat

Obojsmerný konečný automat načíta vstupné symboly z pásky, nad ktorou sa pohybuje čítacia hlava ktorá sa môže pohybovať v smere v ľavo alebo v pravo.

Napriek tomu že obojsmerný konečný automat vyzerá výpočtetne silnejší ako jednosmerný konečný automat, v skutočnosti má rovnakú výpočetnú silu.

Výpočet je ukončený keď automat skončí v prijímacom alebo zamietacom stave.

Formálna definícia:

Obojsmerný konečný automat je  $M = (Q, \Sigma, \&, \delta, s, t, r)$

kde

- $Q$  je množina stavov
- $\Sigma$  je vstupná abeceda
- $\&$  je symbol označujúci koniec pásky
- $\delta$  je prechodová funkcia  $\delta \subseteq (Q \times (\Sigma \cup \{\&\})) \rightarrow (Q \times \{L, R\})$   
kde  $L$  je pohyb v ľavo a  $R$  pohyb v pravo s čítacou hlavou
- $s \in Q$  je počiatočný stav
- $t$  je prijímací stav
- $r$  je zamietací stav kde  $r \neq t$

Táto definícia bola prebratá z [2]

## 2.3 Deterministický zásobníkový automat

*Deterministický zásobníkový automat načítá vstupné symboly rovnako ako jednosmerný konečný automat, teda ak by boli symboly na páske, tak sa čítacia hlava ktorá ich načítá môže pohybovať iba doprava od začiatku pásky o jedno pole. Čítacia hlava sa môže aj zastaviť.*

*Zásobníkový automat má navyše teoreticky nekonečný zásobník typu LIFO, last in first out, v ktorom sú uložené symboly zásobníkovej abecedy.*

*Automat načítá symbol na páske a symbol na zásobníku a na základe týchto symbolov a stavu v ktorom sa nachádza, zmení stav, zapíše symbol na zásobník, alebo odstráni symbol zo zásobníku a vykoná pohyb čítacou hlavou.*

*Jazyk ktorý je definovaný týmto automatom patrí do množiny deterministických bezkontextových jazykov, ktorý je menší ako množina bezkontextových jazykov.*

*Reťazec (slovo) je prijatý automatom pokiaľ po prečítaní pásky je automat v prijímacom stave. Môže ale aj fungovať tým spôsobom že nie sú definované žiadne prijímacie stavy a reťazec na vstupe je prijatý práve vtedy ak po prečítaní reťazca je zásobník prázdny.*

*Formálna definícia:*

*Deterministický zásobníkový automat je  $M = (Q, \Sigma, \Gamma, \delta, \perp, \$, s, F)$*

*kde*

- $Q$  je množina stavov
- $\Sigma$  je vstupná abeceda
- $\Gamma$  je zásobníková abeceda
- $\perp$  je počiatkový zásobníkový symbol
- $\$$  je symbol označujúci koniec pásky
- $\delta$  je prechodová funkcia  $\delta \subseteq (Q \times (\Sigma \cup \{\$\} \cup \{\varepsilon\} \times \Gamma)) \rightarrow (Q \times \Gamma^*)$   
*kde  $L$  je pohyb v ľavo a  $R$  pohyb v pravo s čítacou hlavou*
- $s \in Q$  je počiatkový stav
- $F$  je množina prijímacích stavov

*$\delta$  je deterministická, čo znamená práve jeden prechod nastat' v konkrétnej situácii. Formálne to znamená že pre každé  $p \in Q, a \in \Sigma \cup \{\$\} \text{ a } A \in \Gamma, \beta \in \Gamma^*$*

*$\delta$  obsahuje práve jednu prechodovú formu  $(p, a, A) \rightarrow (q, \beta)$  alebo  $(p, \varepsilon, A) \rightarrow (q, \beta)$*

*$\delta$  je obmedzené tak že  $\perp$  je vždy na spodku zásobníku. Automat môže vytiahnuť  $\perp$  zo zásobníku, ale musí ho okamžite uložiť späť.*

*To znamená že všetky prechody musia byť vo forme  $(p, a, \perp) \rightarrow (q, \beta\perp)$ .*

*Toto obmedzenie je potrebné aby sa automat nezasekol pri vyprázdňovaní zásobníku.*

Táto definícia bola prebratá z [2]

## 2.4 Nedeterministický zásobníkový automat

*Nedeterministický zásobníkový automat skrátene označovaný iba ako zásobníkový automat, je taký istý ako deterministický zásobníkový automat iba s malými rozdielmi, ale napriek tomu má vyššiu výpočetnú silu.*

*Jazyky Rozpoznané nedeterministickým zásobníkovým automatom patria do množiny nedeterministických bezkontextových jazykov. Príkladom rozdielu medzi nedeterministickými a deterministickými bezkontextovými jazykmi môže byť problém rozpoznania palindrómu.*

*Deterministický zásobníkový automat nie je schopný rozpoznať palindróm, pretože nie je pre neho možné určiť stred slova, zatiaľ čo nedeterministický zásobníkový automat s touto úlohou nemá problém.*

*Nedeterministický zásobníkový automat je definovaný formálne:*

$$M = ( Q, \Sigma, \Gamma, \delta, \perp, s, F )$$

*kde*

- $Q$  je množina stavov
- $\Sigma$  je vstupná abeceda
- $\Gamma$  je zásobníková abeceda
- $\perp \in \Gamma$  je počiatočný symbol na zásobníku
- $\delta$  je prechodová funkcia  $\delta \subseteq ( Q \times ( \Sigma \cup \{ \epsilon \} \times \Gamma ) ) \rightarrow ( Q \times \Gamma^* )$   
kde  $L$  je pohyb v ľavo a  $R$  pohyb v pravo s čítacou hlavou
- $s \in Q$  je počiatočný stav
- $F$  je množina prijímacích stavov

*Nedeterministický zásobníkový automat nepotrebuje mať špeciálny symbol pre koniec pásky, pretože vďaka jeho nedeterministickej vlastnosti, mať viac prechodov pre jednu situáciu, môžeme koniec pásky proste uhádnuť.*

Táto definícia bola prebratá z [2]

## 2.5 Turingov stroj

*Deterministický jedno páskový Turingov stroj je definovaný konečným počtom stavov, páskou ktorá je teoreticky nekonečná a hlavou ktorá sa môže pohybovať v ľavo, v pravo alebo stáť. Možnosť stáť nie je povinná. Hlava číta aj zapisuje symboly na pásku.*

*Vstupný reťazec je zapísaný na páske a je konečnej dĺžky.*

*Neoznačené pole na páske je označené symbolom \$. Stroj začína v stave  $s$ , s hlavou na prvom symbole zľava na páske.*

*Prechodová funkcia  $\delta$  v závislosti na prečítanom symbole  $a$  a na aktuálnom stave, zapíše nový symbol na pásku, zmení svoj stav a posunie hlavu vľavo, vpravo, alebo stojí. Reťazec na páske je akceptovaný pokiaľ Turingov stroj vstúpi do prijímacieho stavu  $t$  a je zamietnutý pokiaľ vstúpi do zamietacieho stavu  $r$ .*

*Môže nastať aj prípad keď neskončí v žiadnom z týchto stavov a bude cykliť do nekonečna.*

Štandardne je páska ukončená na jednom konci a nekonečná na druhom. Pre tento program ale budem uvažovať prípad že páska je nekonečná na oboch koncoch a symbol \$ slúži na detekovanie prázdneho poľa.

*Jazyk definovaný Turingovým strojom patrí do množiny rekurzívne vyčísliteľných jazykov.*

*Formálna definícia:*

*Turingov stroj  $M$  je  $M = (Q, \Sigma, \Gamma, \$, \delta, s, t, r)$*

*kde*

- $Q$  je množina stavov
- $\Sigma$  je vstupná abeceda ( abeceda symbolov ktoré sa môžu nachádzať na vstupe )
- $\Gamma$  je pásková abeceda ( abeceda všetkých symbolov ktoré sa môžu objaviť na páske )
- $\$$  je symbol označujúci koniec pásky
- $\delta$  je prechodová funkcia  $\delta \subseteq (Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{L, R, S\})$   
kde  $L$  je pohyb v ľavo a  $R$  pohyb v pravo a  $S$  je stáť s čítacou hlavou.
- $s \in Q$  je počiatkový stav
- $t \in Q$  je prijímací stav
- $r \in Q$  je zamietací stav kde  $r \neq t$

Táto definícia bola prebratá z [2]

## 2.6 Lineárne ohraničený Turingov stroj

*Lineárne ohraničený Turingov stroj je špeciálny prípad Turingovho stroja ktorý ma omedzenie pri zápise na pásku. Na rozdiel od bežného Turingovho stroja, totiž nesmie zapisovať na neobmedzene mnoho buniek pásky, ale iba na prvých  $n$  buniek kde  $n$  je omedzené lineárnou funkciou vzhľadom k dĺžke vstupného slova.*

*V určitom zmysle je tak tento stroj bližší bežným počítačom.*

*Lineárne ohraničené Turingove stroje vedia akceptovať jazyky z triedy kontextových jazykov. Dá sa dokázať že pre každý lineárne ohraničený Turingov stroj je možné zostrojiť Turingov stroj ktorý nepotrebuje pásku dlhšiu ako pásku vstupného slova. Táto definícia bola prebratá z [3]*

## 2.7 Vždy zastavujúci Turingov stroj

*Vždy zastavujúci Turingov stroj alebo tiež nazývaný totálny Turingov stroj je Turingov stroj ktorý zastaví na každom vstupe. Kvôli tomu že vždy zastaví, rozhoduje o tom či reťazec patrí do jazyka z množiny rekurzívnych jazykov. Problém je ale, že samotné rozhodnutie či Turingov stroj zastaví na každom vstupe je algoritmicke neriešiteľné. Táto definícia bola prebratá z [4]*

## 2.8 Turingov stroj s viacerými páskami

*Viac páskový Turingov stroj je ako obyčajný Turingov stroj s viacerými páskami. Každá páska ma svoju vlastnú hlavu na čítanie a písanie.*

*Tento model vyzerá výpočetne silnejší ako jedno páskový Turingov stroj, ale každý viac páskový Turingov stroj môže byť simulovaný na jedno páskovom Turingovom stroji, s využitím kvadraticky viac výpočetného času., to znamená že viac páskový Turingov stroj nemôže vyriešiť viac problémov ako jedno páskový Turingov stroj, a žiaden z robustných problémov ktoré sú vypočítateľné v polynomiálnom čase, sú ovplyvnené prechodom medzi jedno páskovým Turingovým strojom a viac páskovým Turingovým strojom.*

Turingov stroj  $M$  je  $M = (Q, \Sigma, \Gamma, \$, \delta, s, t, r)$

kde

- $\delta$  je prechodová funkcia  $\delta \subseteq (Q \times \Gamma^k) \rightarrow (Q \times (\Gamma \times \{L, R, S\})^k)$

Táto definícia bola prebratá z [5]

## 3 Popis programu

Program univerzálny simulátor automatov je implementovaný hlavne ako simulátor Turingovho stroja a ostatné automaty sú implementované ako Turingov stroj s obmedzeniami.

Hlavnou časťou programu je trieda Turingov\_stroj ktorá obsahuje atribúty :

```
std::vector<int> m_mnozina_stavov;
std::vector<char> m_vstupna_abeceda;
std::vector<char> m_paskova_abeceda;
//názov je zavádzajúci, celý vektor je prechodová funkcia a jednotlivé členy vektoru sú prechody
std::vector<prechodova_funkcia> m_mnozina_prechodovych_funkcii;
int m_pociatocny_stav;
int m_primaci_stav;
int m_zamietaci_stav;
std::vector<char> m_paska;
std::vector<char>::size_type pozicia;
```

kde prechodova\_funkcia je trieda ktorá obsahuje atribúty:

```
int m_stav_vstupu;
char m_symbol_vstupu;
int m_stav_vystupu;
char m_symbol_vystupu;
char m_smer; //0 stoj, 1 do lava, 2 do prava
```

Konkrétne Turingove stroje sú uložené v textovom súbore, odkiaľ sa načítajú do programu.

Štruktúra Turingovho stroja uloženého v súbore:

Prvý riadok označuje typ Turingovho stroja. “a“ znamená deterministický konečný automat, “z“ znamená deterministický zásobníkový automat a číslo znamená Turingov stroj, kde číslo označuje počet pásov.

Druhý riadok je zoznam stavov. Tretí riadok je vstupná abeceda. Štvrtý riadok je pásková abeceda. Na piatom riadku je počiatočný stav.

Pokiaľ sa jedná o konečný automat alebo o zásobníkový automat, tak na šiestom riadku je zoznam koncových stavov, pokiaľ je to ale Turingov stroj, tak prvý symbol šiesteho riadku je prijímací stav a druhý symbol je zamietací stav. Nasledujúce riadky sú samotné prechody v prechodovej funkcií.

Formát prechodov v konečnom automate je:

Prvý symbol je aktuálny stav, druhý symbol je prečítaný symbol na páske a tretí symbol je stav na ktorý sa má zmeniť.

Formát prechodových stavov v jedno páskovom Turingovom stroji je:

Prvý symbol je aktuálny stav, druhý symbol je prečítaný symbol na páske, tretí symbol je stav na ktorý sa má zmeniť, štvrtý symbol je symbol ktorý sa ma zapísať na pásku a piaty symbol označuje smer pohybu hlavy, kde 0 znamená stáť, 1 alebo – znamená vľavo a 2 alebo + znamená vpravo.

Implementácia konečného automatu:

Konečný automat je implementovaný nasledujúcim spôsobom:

K zadaným stavom budú pridané dva nové stavy, prijímací a zamietací stav. Od každého stavu ktorý je označený ako koncový bude vytvorený nový prechod v prechodovej funkcii, do prijímacieho stavu kde prečítaný a zároveň zapísaný symbol na páske je symbol označujúci koniec pásky \$.

K zoznamu prechodov kde je zadaný iba aktuálny stav, prečítaný symbol na páske a stav na ktorý sa má zmeniť, je pridaný symbol ktorý sa ma zapísať na pásku ktorý je rovnaký ako prečítaný symbol a smer pohybu hlavy ktorý je v pravo.

Via páskový Turingov stroj som pre zjednodušenie implementoval zvlášť od jedno páskového Turingovho stroja. Do triedy Turingov\_stroj som pridal atribúty:

```
int m_pocet_pasok;  
std::vector<std::vector<char>> > m_pasky;  
std::vector<std::vector<char>::size_type> pozicie;  
std::vector<prechodova_funkcia_mttm> m_mnozina_prechodovych_funkcii_mttm;  
std::vector<prechodova_funkcia_mttm> m_mnozina_prechodovych_funkcii_mttm;
```

kde prechodova\_funkcia\_mttm je trieda ktorá obsahuje atribúty:

```
int m_stav_vstupu;  
string m_symbol_vstupu;  
int m_stav_vystupu;  
string m_symbol_vystupu;  
string m_smer; //0 stoj, 1 do lava, 2 do prava
```

Formát prechodových stavov vo viac páskovom Turingovom stroji je:

Prvý symbol je aktuálny stav, potom nasledujúcich n symbolov sú prečítané symboly na páske kde n je počet pásek, nasledujúci symbol je stav na ktorý sa má zmeniť, nasledujúcich n symbolov sú symboly ktoré sa majú zapísať na pásku a posledný symbol označuje smer pohybu hlavy, kde 0 znamená stáť, 1 alebo – znamená vľavo a 2 alebo + znamená vpravo.



Formát prechodov v zásobníkovom automate je:

Prvý symbol je aktuálny stav, druhý symbol je prečítaný symbol na páske, tretí symbol je prečítaný symbol na zásobníku a štvrtý symbol je stav na ktorý sa má zmeniť. zvyšné symboly sú symboly ktoré sa majú zapísať do zásobníku. Pokiaľ tam nie je žiaden tak je to vybranie symbolu.

Implementácia deterministického zásobníkového automatu:

Deterministický zásobníkový automat je implementovaný ako špeciálny prípad dvoj páskového Turingovho stroja.

K zadaným stavom budú pridané dva nové stavy, prijímací a zamietací stav úplne rovnakým spôsobom ako v konečnom automate.

Z každého prechodu  $(q_1, a, b) \rightarrow (q_2, b^n)$

kde  $q_1, q_2 \in Q, a \in \Sigma, b \in \Gamma$

v prípade že  $n \geq 1$

sa vytvorí  $n-2$  prechodov  $(w_x, a, b_x) \rightarrow (w_{x+1}, a, b_{x+1}, \{S\}, \{R\})$

, jeden prechod  $(q_1, a, b) \rightarrow (w_1, a, b_1, \{S\}, \{R\})$

a jeden prechod  $(w_{n-2}, a, b_{n-1}) \rightarrow (q_2, a, b_n, \{R\}, \{R\})$

kde  $q_1, q_2, w_x \in Q, a, b \in \Sigma, b_x \in \Gamma^n$

A v prípade že  $n = 0$

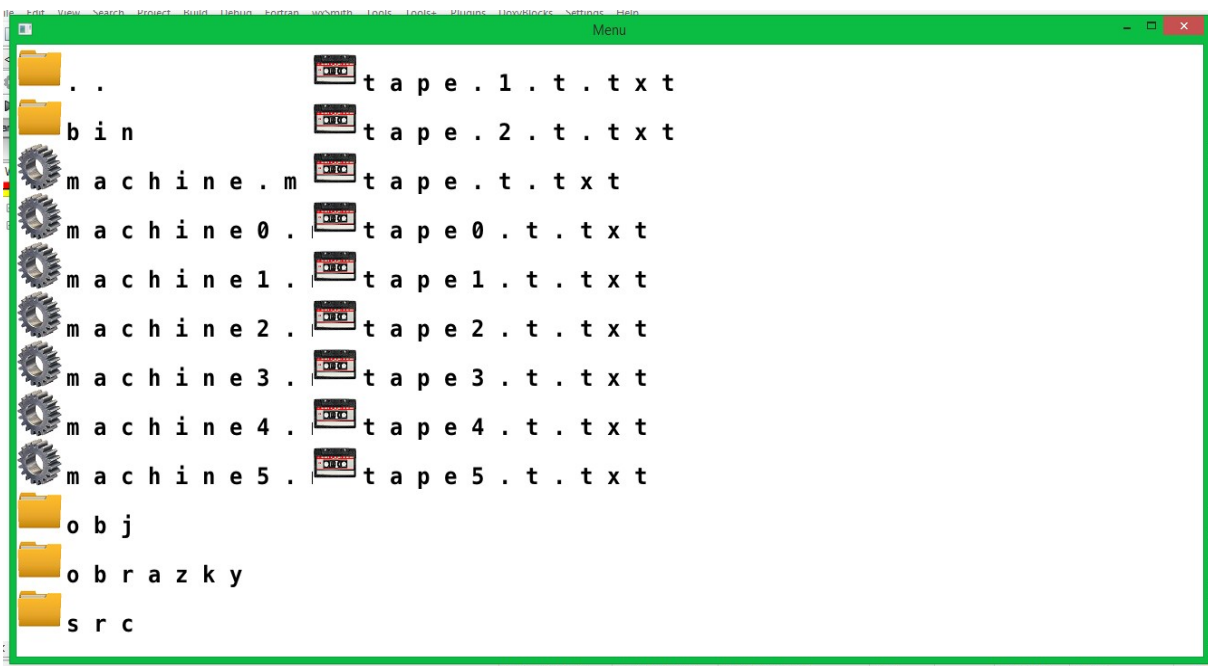
sa vytvorí prechod  $(q_1, a, b) \rightarrow (q_2, a, \$, \{R\}, \{L\})$

## 3.1 Užívateľské rozhranie

Samotný program pracuje v troch módoch: Menu, Turing machine a Machine creator a medzi ktorými sa dá ľubovoľne prepínať. V Menu si užívateľ vyberie s ktorými súbormi má pracovať. Súbor typu m.txt sú súbory v ktorých je uložený kód Turingovho stroja alebo automatu a v súbore typu t.txt sú uložené súbory obsahujúce pásku.

Užívateľ si vyberie jeden súbor typu m.txt a jeden súbor typu t.txt a program bude automaticky prepnutý do módu Turing machine.

Ukážka módu Menu:



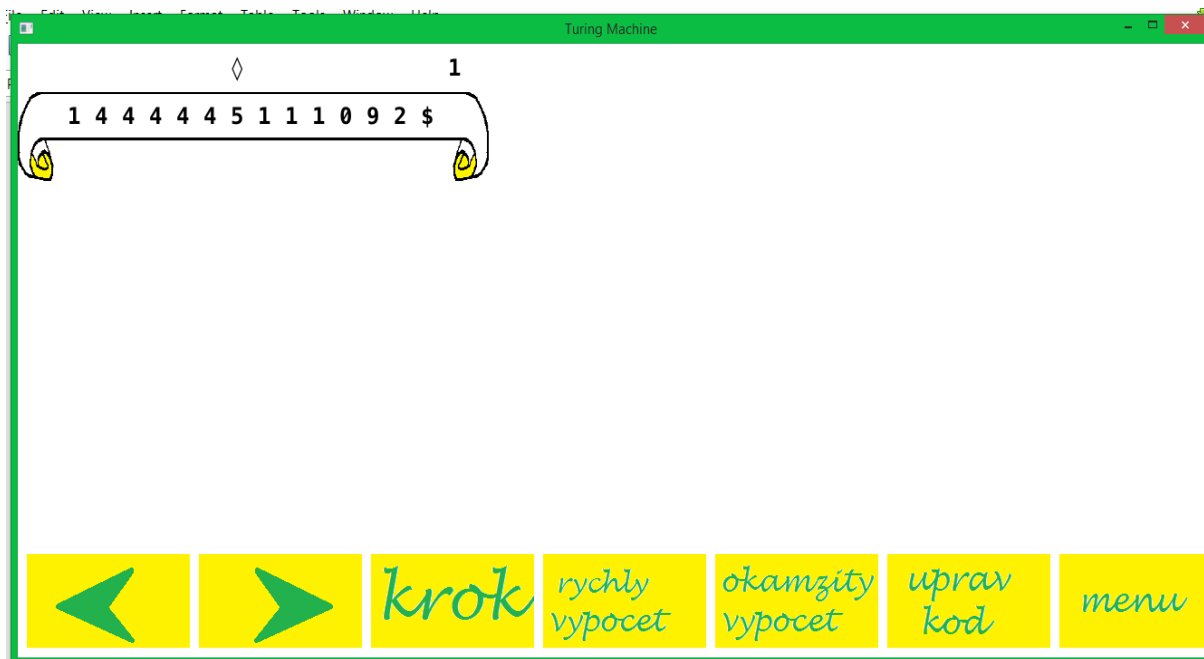
V tomto móde Turing machine už prebieha simulácia samotná, kde užívateľ môže postupne krokovať svoj program a prezerat' jednotlivé kroky výpočtu.

Tlačítkom **krok** v okne, simulátor vykoná jeden krok výpočtu. Tlačítkom **rychly vypocet**, simulátor vykoná všetky kroky výpočtu až do konca, pokiaľ sa stroj nezacyklí a užívateľ uvidí celý výpočet v zrýchlenej forme na obrazovke. Tlačítkom **okanzity vypocet**, simulátor vykoná všetky kroky výpočtu až do konca okamžite a na obrazovke zobrazí až výsledok výpočtu.

Šípkami na ľavej strane okna sa môže užívateľ vracat' späť vo výpočte a ísť zase dopredu, ale najviac iba do bodu kde skončil výpočet.

Tlačítkom **uprav kod** prejdeme do módu Machine creator.

Ukážka módu Turing machine:



V móde Machine creator si užívateľ môže prezrieť a upraviť kód svojho programu vo forme prechodového grafu.

Prechody sú zobrazené ako úsečky zo symbolmi medzi kruhmi označujúcimi stavy.

V prípade že sa jedná o Turingov stroj, tak jeden prechod sú tri symboly kde prvý je očakávaný symbol na páske, druhý je smer pohybu kde + je doprava – doľava a 0 je stáť a tretí symbol ktorý sa zapíše na pásku.

V prípade že sa jedná o konečný automat tak je tam iba jeden symbol, a to očakávaný symbol na vstupe / na páske.

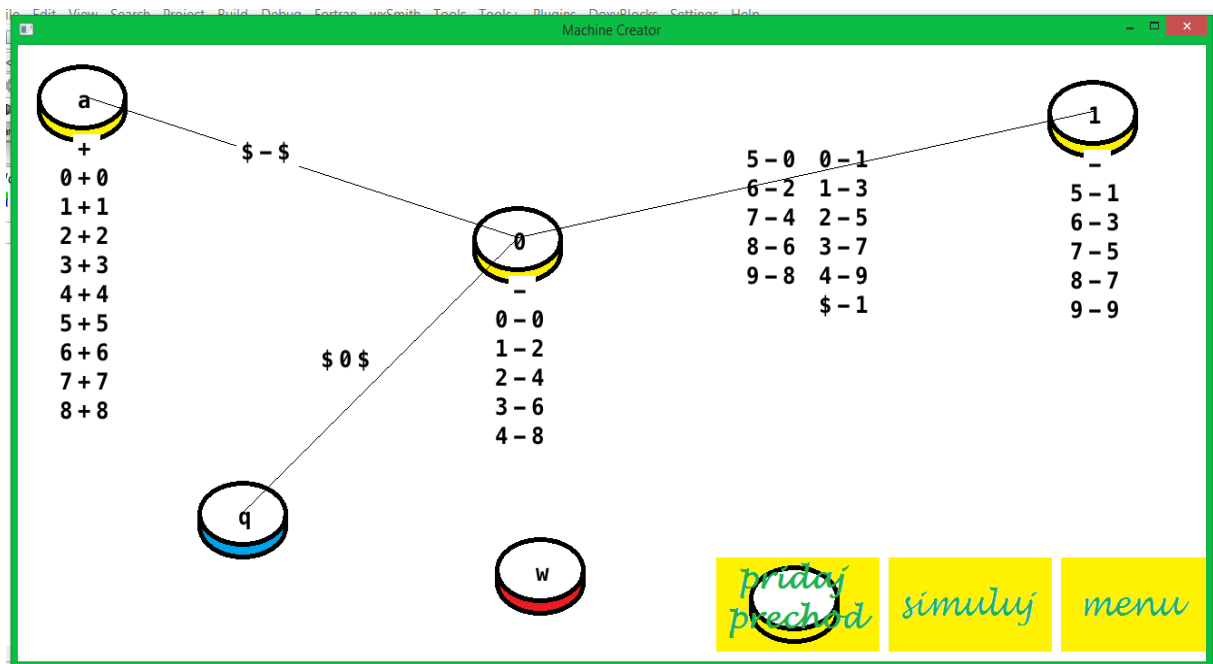
V prípade že sa jedná o viac páskový Turingov stroj tak sú tam jednotlivé symboly umiestnené po blokoch symbolov, kde veľkosť jedného bloku je rovný počtu páso. U zásobníkového automatu machine creator nefunguje.

Ovládanie je myšou a klávesami. Myšou môžeme posúvať malé kruhy reprezentujúce stavy kliknutím na ne, držaním klávesy myši posunutím na požadovanú pozíciu a uvoľnením klávesy myši. Stlačením kláves šípka dole a šípka hore, prechádzam postupne všetky prechody, podľa poradia ako sú zapísaná v súbore. Vybraný prechod je zobrazený žltou farbou.

Medzi samotnými symbolmi v prechode prechádzam pomocou kláves šípka doprava a šípka doľava.

Daný symbol potom zmením stlačením ľubovolnej inej klávesy na klávesnici. Zmena sa zapíše do súboru.

Ukážka módu Machine creator:



## 3.2 Príklady a testovanie

### 3.2.1 Test 1

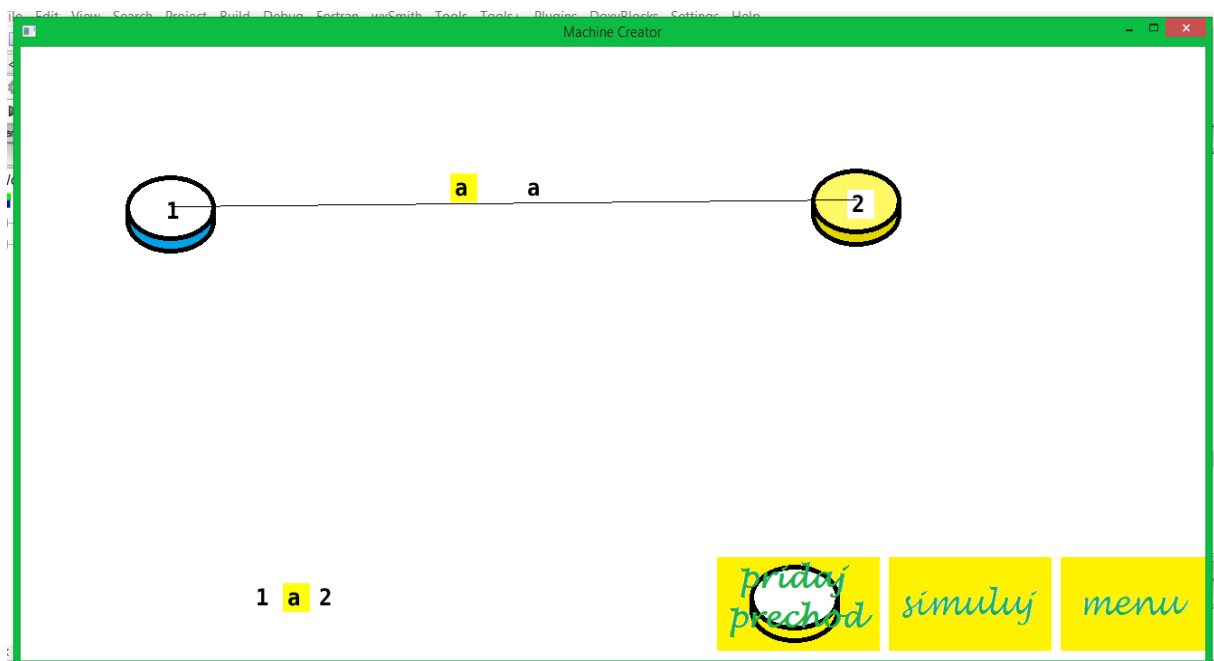
Automat ktorý vezme reťazec zložený iba zo znakov "a" a zistí či je zložený z párneho počtu znakov:

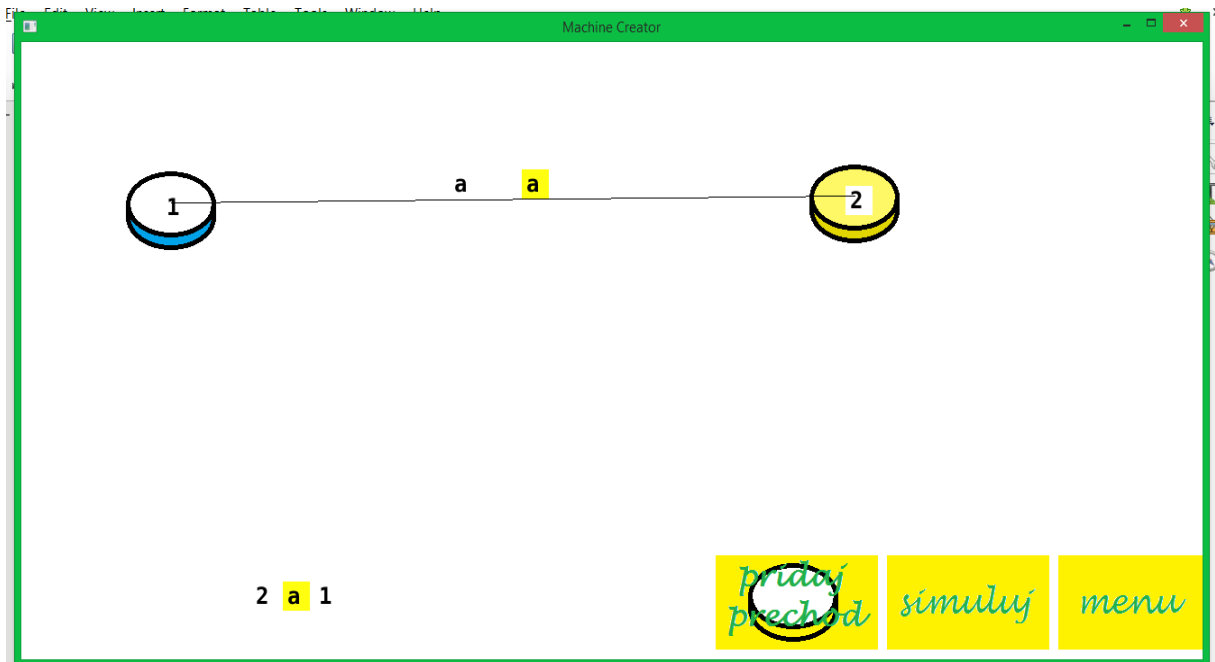
prvý riadok: a -automat  
druhý riadok: 12 -stavy  
tretý riadok: a -vstupná abeceda  
štvrtý riadok: a\$ -pásková abeceda  
piaty riadok: 1 -počiatočný stav  
šiesty riadok: 1 -zoznam koncových stavov

prechody:

1a2

2a1





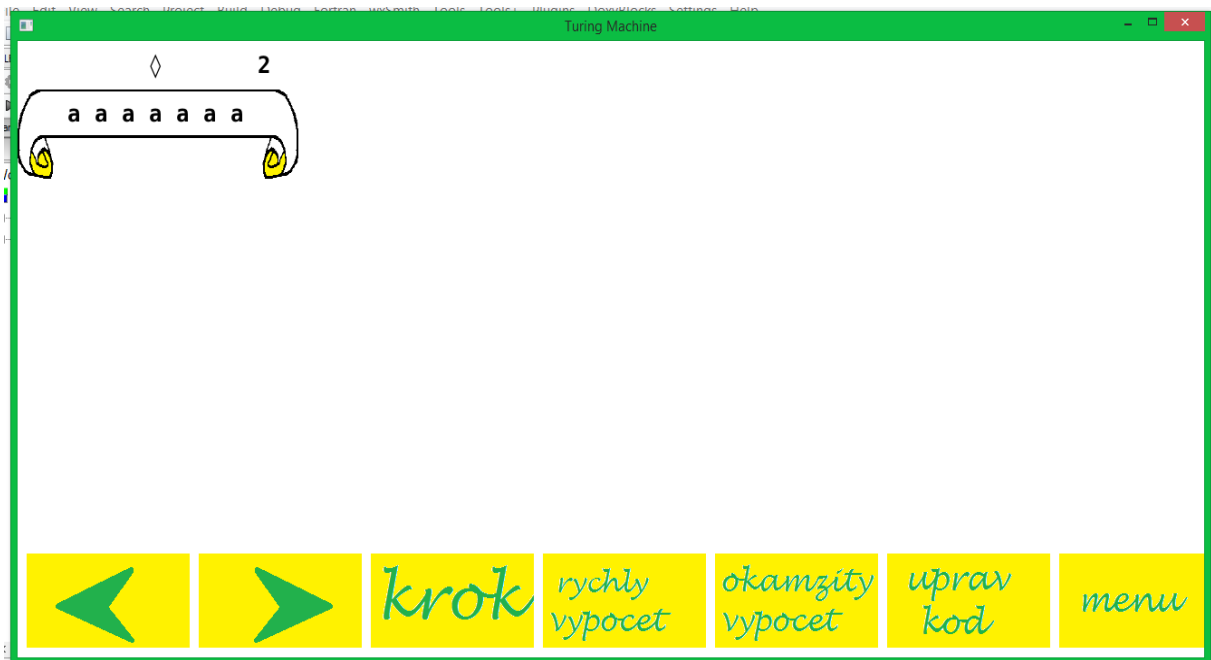
Teraz daný automat otestujeme nad páskou: aaaaaaa

Počet písmen a na páske je 7, takže reťazec na páske nepatrí do jazyka. Otestujeme ako na to zareaguje program.

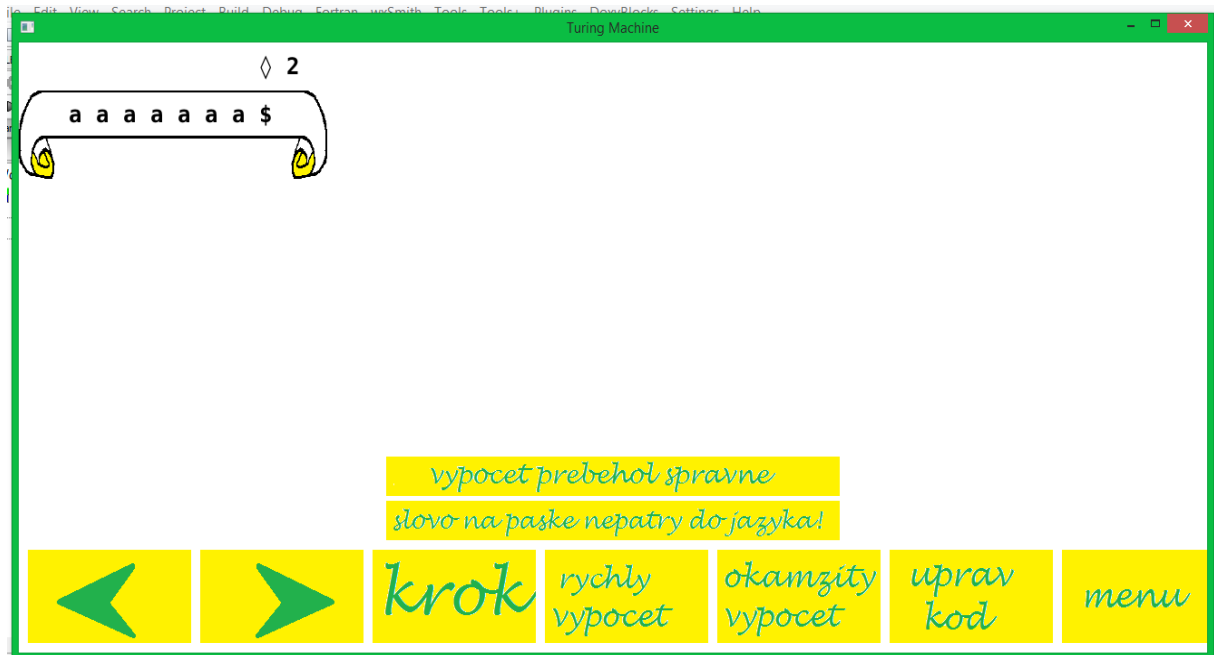
Páska na začiatku výpočtu:



Páska uprostred výpočtu:



Páska na konci výpočtu:



Výsledok testu bol správny, výpočet skončil v stave 2 ktorý nie je prijímací.

Teraz otestujeme ten istý automat nad páskou: aaaaaa

Počet písmen a na páske je 6, takže reťazec na páske patrí do jazyka. Otestujeme ako na to zareaguje program.

Páska na konci výpočtu:



Výsledok testu bol správny, výpočet skončil v stave 1 ktorý je prijímací.

### 3.2.2 Test 2

Turingov stroj ktorý vezme reťazec zložený iba zo znakov "b" a zistí či je zložený zo párneho počtu znakov a zmení ich na znak "a":

prvý riadok: 1 -jedno páskový Turingov stroj

druhý riadok: 1234 -stavy

tretý riadok: ab -vstupná abeceda

štvrtý riadok: ab\$ -pásková abeceda

piaty riadok: 1 -počiatočný stav

šiesty riadok: 34 -prijímací stav a zamietací stav

prechody:

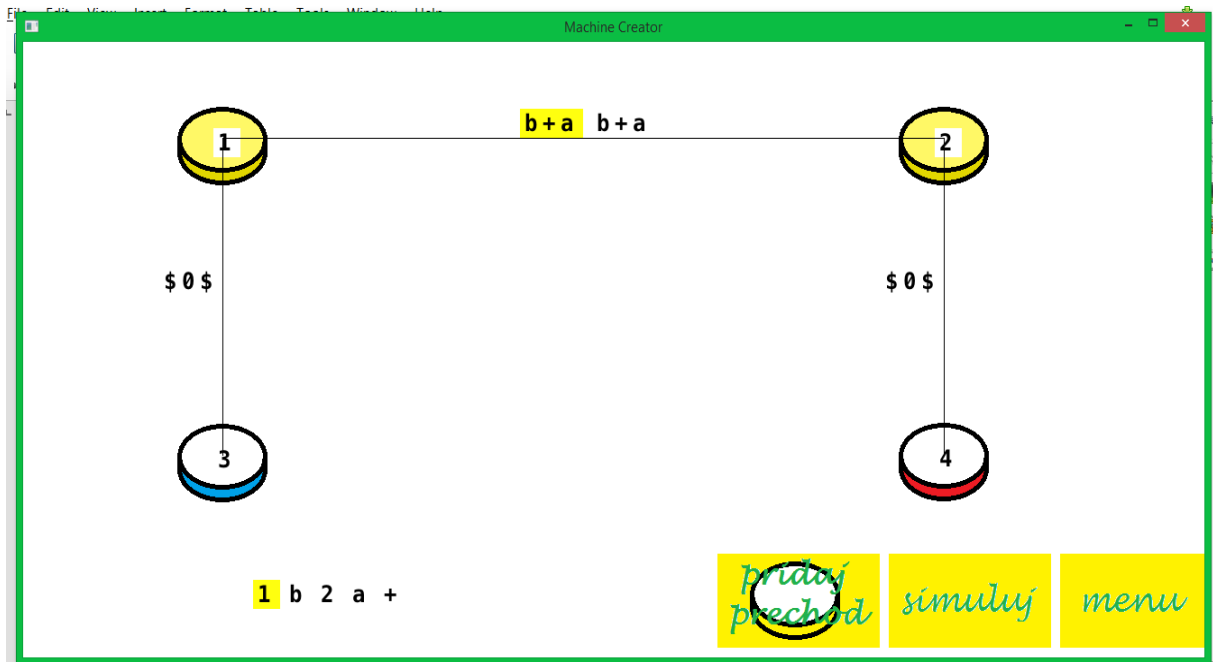
1b2a+

2b1a+

1\$3\$0

2\$4\$0





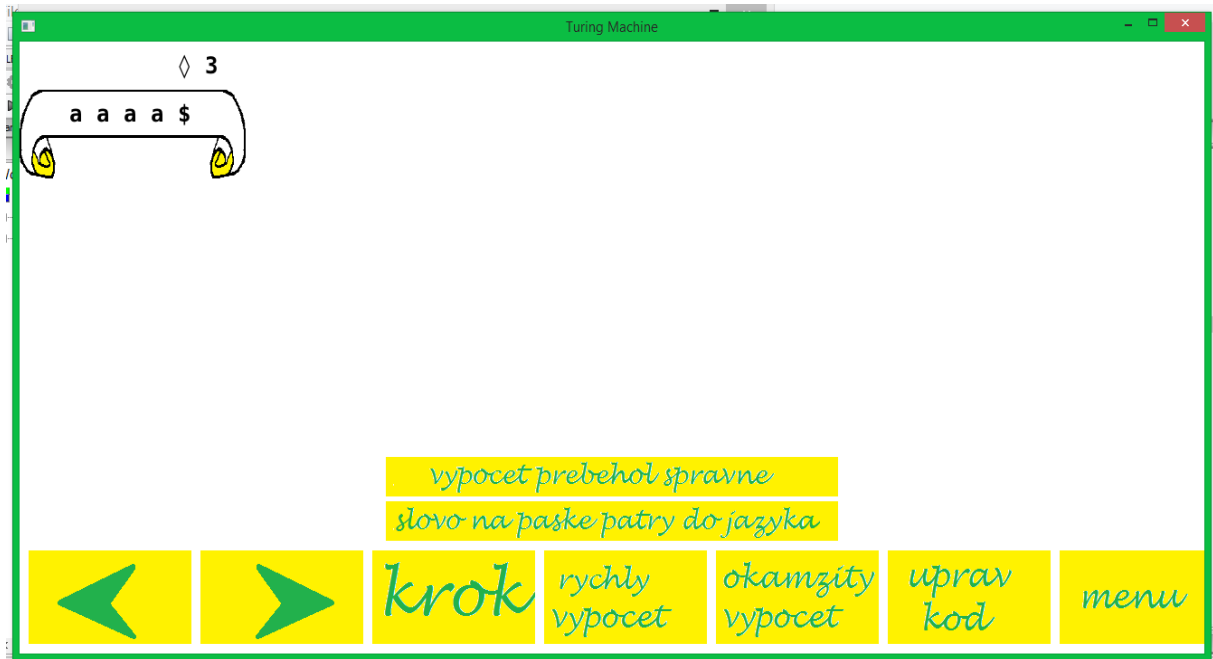
Teraz daný Turingov stroj otestujeme nad páskou: bbbb

Počet písmen b na pásce je 4, takže reťazec na pásce patrí do jazyka.

Páska na začiatku výpočtu:



Páska na konci výpočtu:



Výsledok testu bol správny, výpočet skončil v stave 3 ktorý je prijímací.

Teraz otestujeme ten istý Turingov stroj nad páskou: bbb

Počet písmen b na páske je 3, takže reťazec na páske nepatrí do jazyka.

Páska na konci výpočtu:



Výsledok testu bol správny, výpočet skončil v stave 4 ktorý je zamietací.

### 3.2.3 Test 3

Turingov stroj ktorý vezme reťazec zložený iba zo znakov "a" a zdvojí reťazec.

- prvý riadok: 1 -jedno páskový Turingov stroj
- druhý riadok: 10rlc23 -stavy
- tretý riadok: ab -vstupná abeceda
- štvrtý riadok: ab\$ -pásková abeceda
- piaty riadok: 1 -počiatočný stav
- šiesty riadok: 23 -prijímací stav a zamietací stav

prechody:

1\$0\$+

1a0a0

0arb+

rara+

rbrb+

r\$lb-

lblb-

la0a0

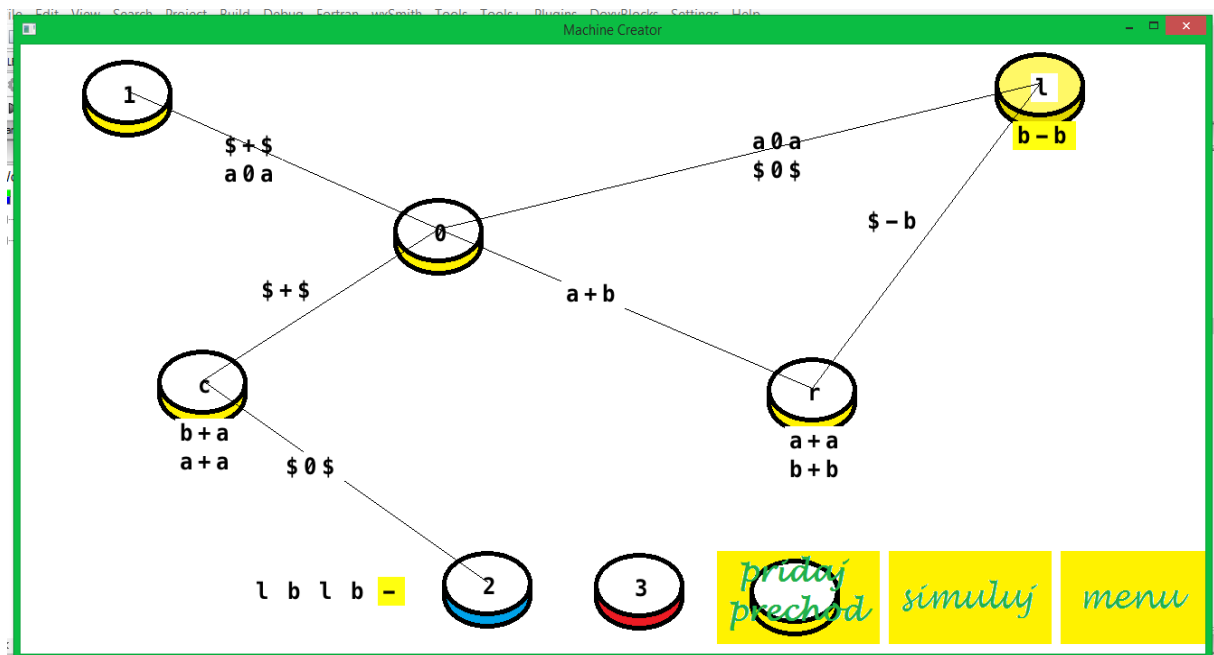
l\$0\$0

0\$c\$+

cbca+

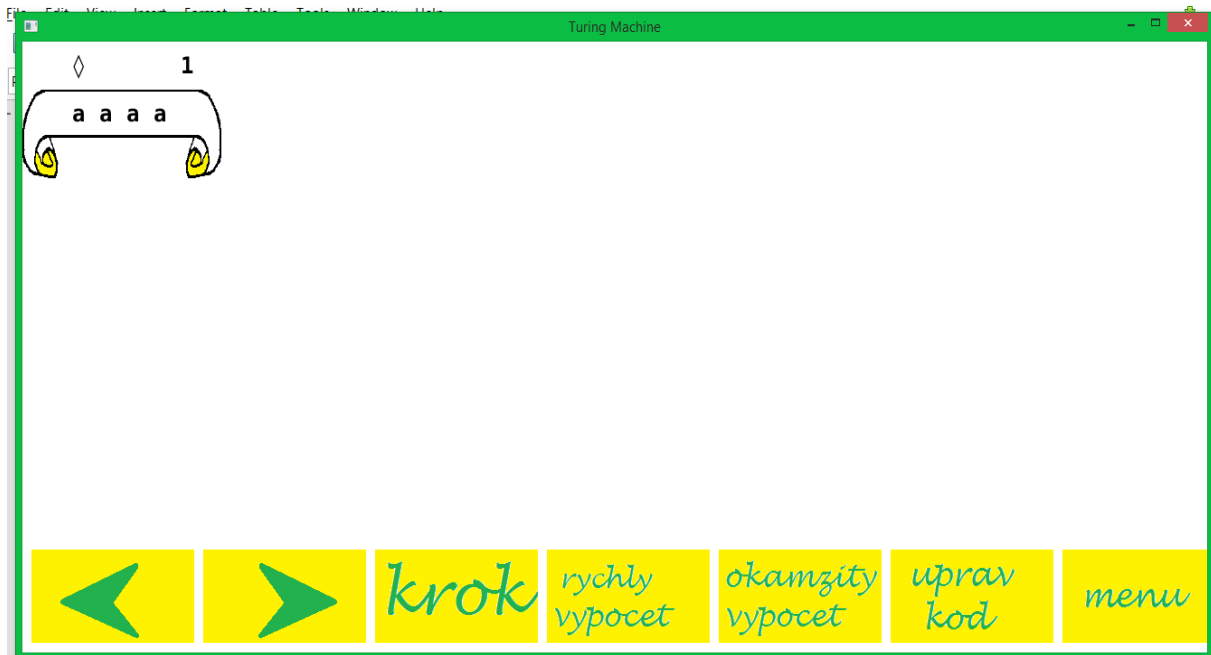
caca+

c\$2\$0

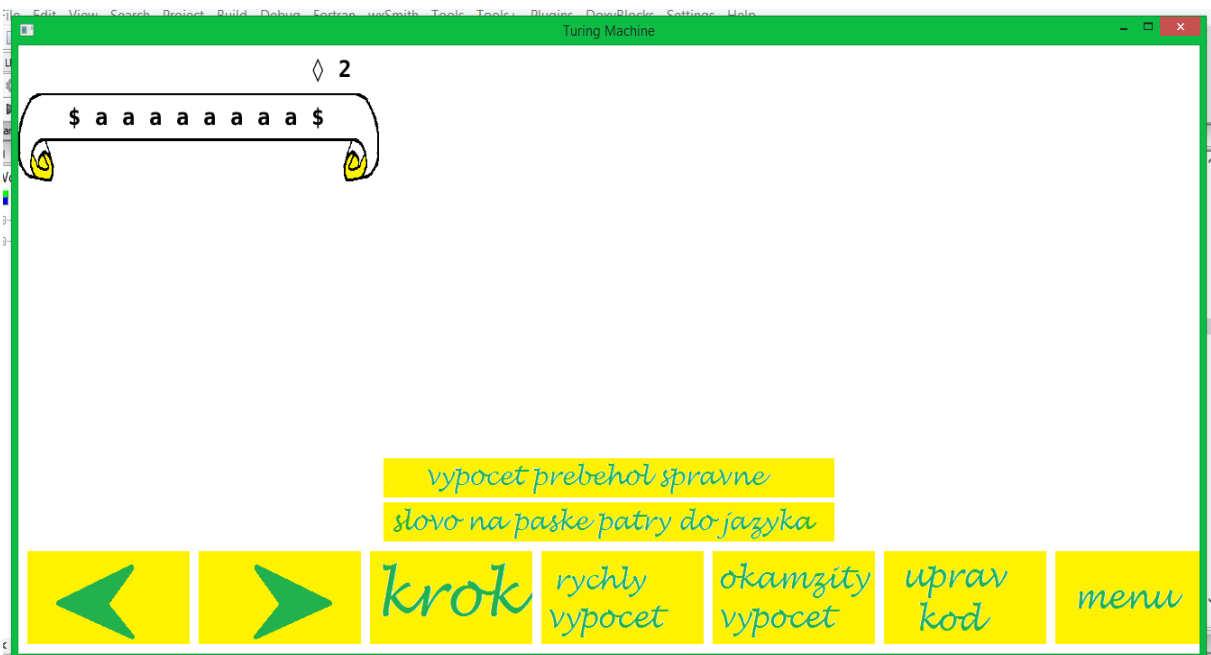


Turingov stroj otestujeme nad páskou: aaaa

Páska na začiatku výpočtu:



Páska na konci výpočtu:



Páska na konci výpočtu je: \$aaaaaaa\$ kde \$ označuje koniec pásky. Počet písmen a je 8 čo je dvojnásobok oproti pôvodnej páske: aaaa . Turingov stroj skončil v stave 2 ktorý je prijímací. Turingov stroj splnil svoj účel.

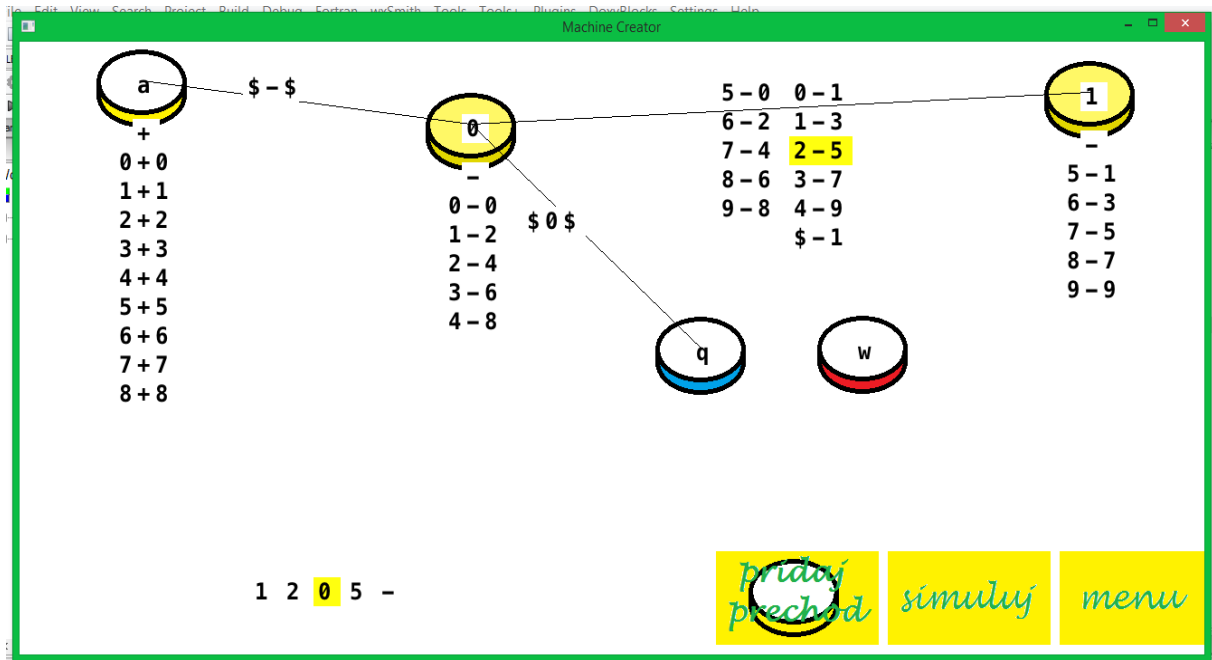
### 3.2.4 Test 4

Turingov stroj ktorý vezme reťazec zložený iba z čísel a vynásobí ho číslom 2.

prvý riadok: 1 -jedno páskový Turingov stroj  
druhý riadok: a01qw -stavy  
tretý riadok: 0123456789 -vstupná abeceda  
štvrtý riadok: 0123456789 \$ -pásková abeceda  
piaty riadok: a -počiatočný stav  
šiesty riadok: qw -prijímací stav a zamietací stav

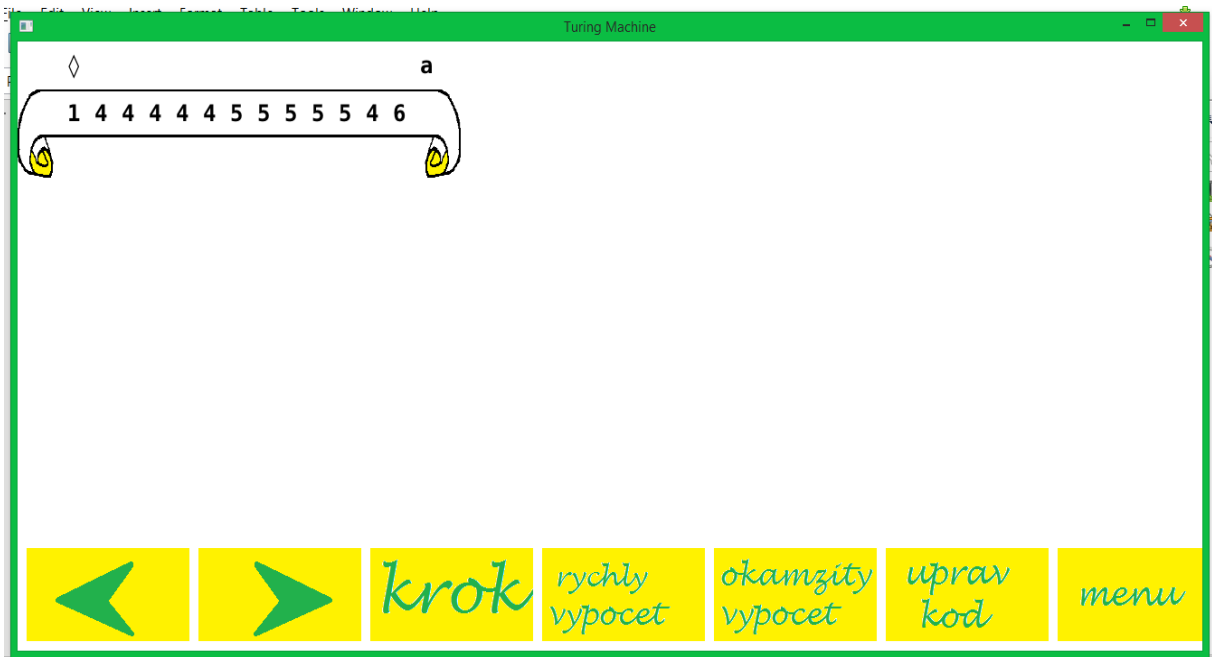
prechody:

a a +	a\$0\$-
1 1 -	0000-
0 0 -	0102-
a0a0+	0204-
a1a1+	0306-
a2a2+	0408-
a3a3+	0510-
a4a4+	0612-
a5a5+	0714-
a6a6+	0816-
a7a7+	0918-
a8a8+	1001-
a9f9+	1103-
	1205-
	1307-
	1409-
	1511-
	1613-
	1715-
	1817-
	1919-
	0\$q\$0
	1\$01-

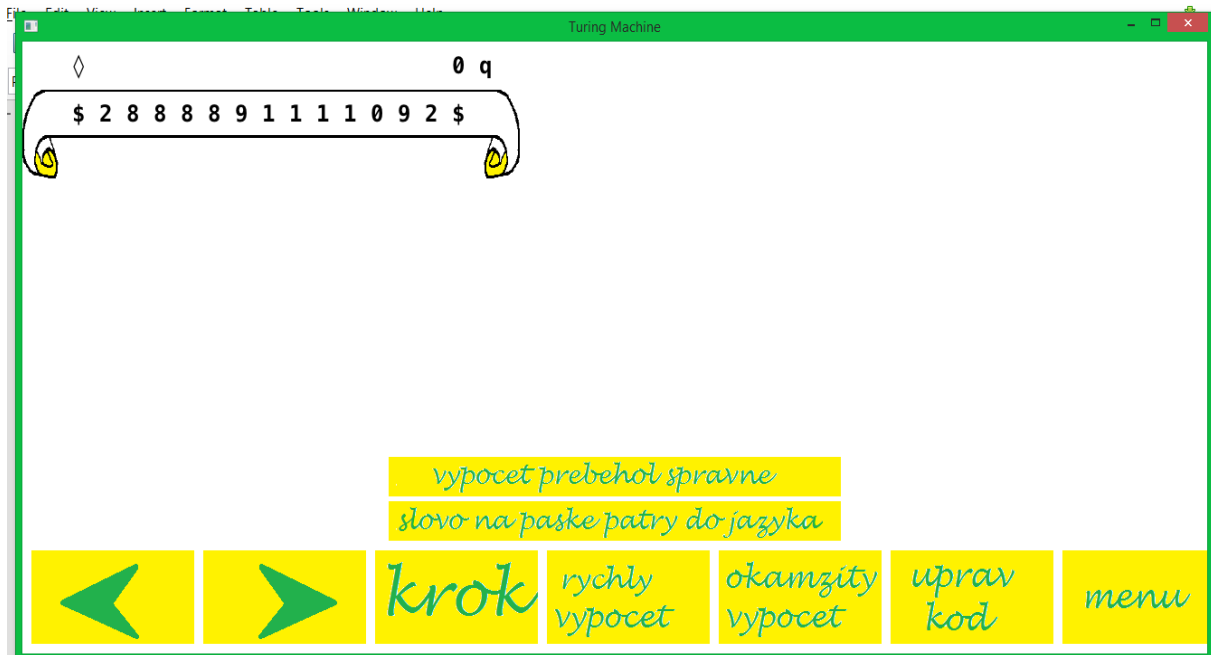


Turingov stroj otestujeme nad páskou: 144444555546

Páska na začiatku výpočtu:



Páska na konci výpočtu:



Páska po výpočte je: \$2888891111092\$ kde číslo 2888891111092 je skutočne dvojnásobok čísla 1444445555546. Turingov stroj pracuje podľa očakávania.

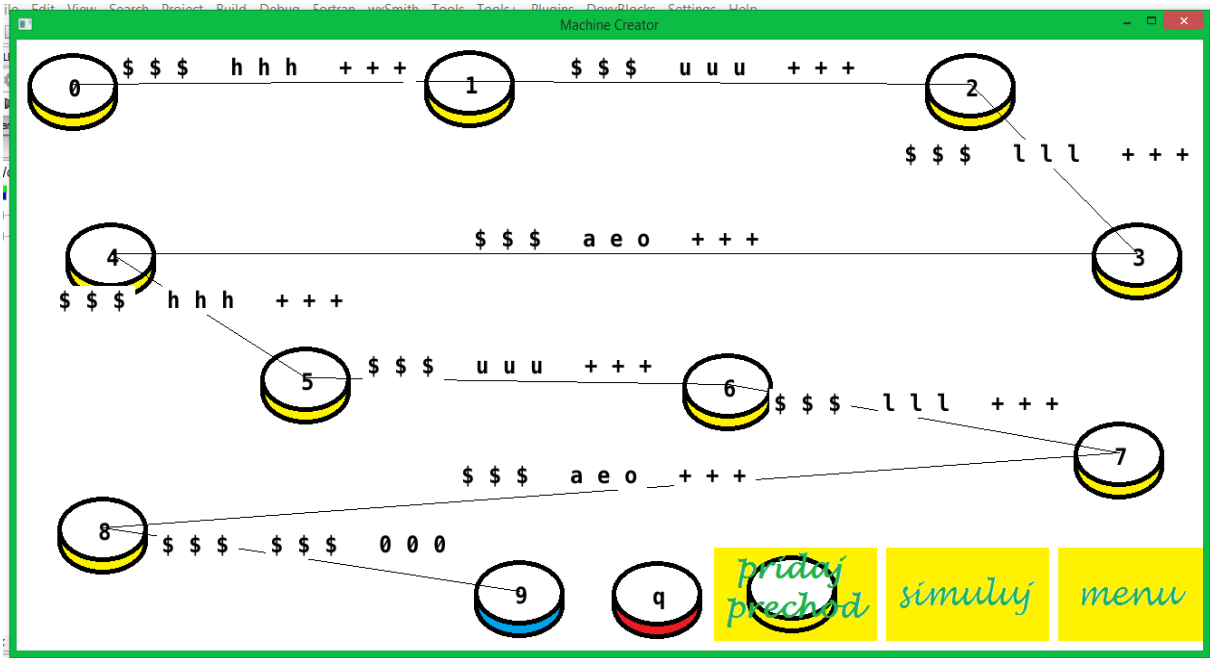
### 3.2.5 Test 5

Troj páskový Turingov stroj ktorý zapíše na prázdne pásky slová hulahula, hulehule a hulohulo.

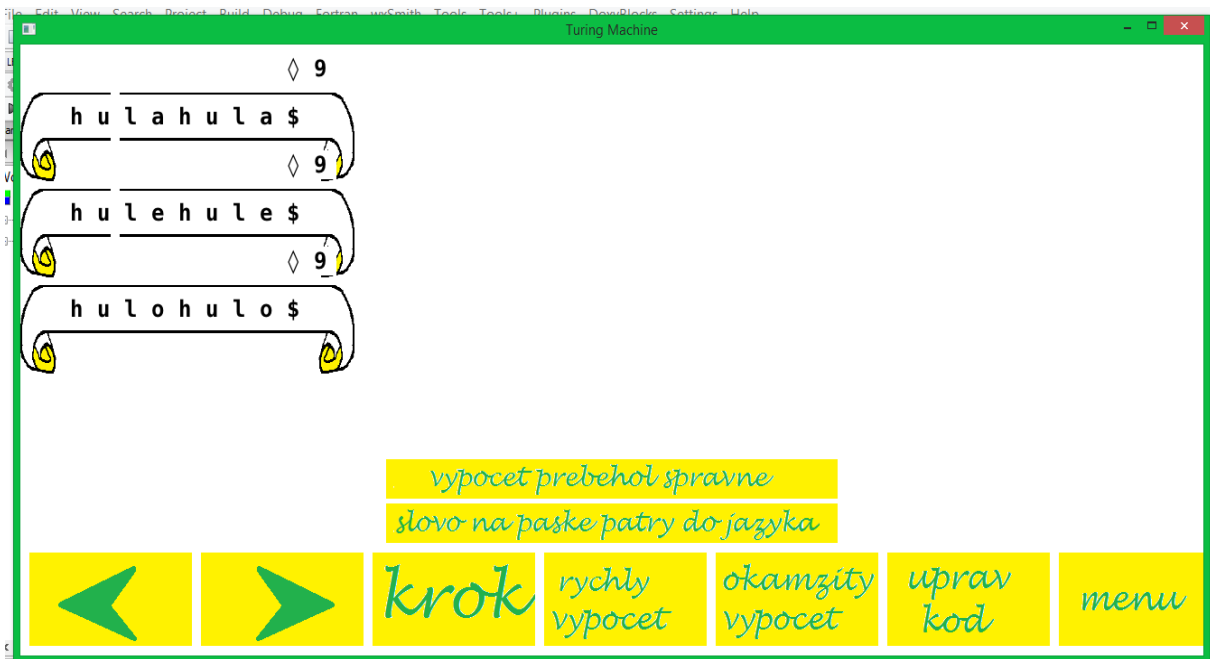
- prvý riadok: 3 -troj páskový Turingov stroj
- druhý riadok: 0123456789q -stavy
- tretý riadok: \$ -vstupná abeceda
- štvrtý riadok: hulao\$ -pásková abeceda
- piaty riadok: 0 -počiatočný stav
- šiesty riadok: 9q -prijímací stav a zamietací stav

prechody:

- 0\$\$\$1hhh+++
- 1\$\$\$2uuu+++
- 2\$\$\$3lll+++
- 3\$\$\$4aeo+++
- 4\$\$\$5hhh+++
- 5\$\$\$6uuu+++
- 6\$\$\$7lll+++
- 7\$\$\$8aeo+++
- 8\$\$\$9\$\$\$000



Páska na konci výpočtu:





## 4 Záver

Simulátor automatov má veľa chýb a nedostatkov. Simulátor nie je ošetrený proti zacykleniu, ani proti prípadu keď užívateľ zapíše chybný údaj do textového súboru s kódom Turingovho stroja. V takom prípade sa môže simulátor správať nevyspytateľne. Problémom je tiež úprava kódu deterministického zásobníkového automatu, kde sa program nespráva podľa očakávania. Najväčšou nevýhodou je, že pri spustení programu nie je vidno všetky časti okna pokiaľ program beží na obrazovke s nižším rozlíšením ako 1310x635 pixelov.

Vo výsledku by som povedal že simulátor je úspešný a plní svoju funkciu, výpočet na všetkých testovaných programoch prebehol podľa očakávania.

V testoch nie je uvedený žiaden deterministický zásobníkový automat, ale aj na ten som vytvoril kód a testoval som ho, kde som zistil že samotná simulácia je v poriadku, až na to že na jeden krok v deterministickom zásobníkovom automate, spotrebujem viac krokov a Turingovom stroji, a to kvôli prevodu z deterministického zásobníkového automatu na dvoj páskový Turingov stroj. Ako som uviedol vyššie, tak zásobníkový automat v móde Machine creator nefunguje správne.

Testy na simuláciu konečných automatov, jedno páskových Turingových strojov a viac páskových Turingových strojov fungujú správne a podľa očakávania. Aj ich úprava v móde Machine creator funguje správne podľa toho čo som otestoval.

Samotný kód je nepríjemne robustný a pokiaľ by som chcel robiť nejaké grafické úpravy v budúcnosti, ťažko by sa mi v kóde orientovalo. Ale pokiaľ by som chcel robiť zmeny vo funkcionalite samotného simulátoru, tak by to nemal byť problém, pretože tá časť kódu je dostatočne priehľadná. Pôvodne som chcel do programu zahrnúť možnosť kompletného vytvorenia implementácie Turingovho stroja v móde Machine creator, ale táto časť sa ukázala zbytočne náročná. Takže Machine creator nevytvára implementáciu, iba ju umožňuje upravovať a pridávať nové prechody. Užívateľ ale už musí mať vytvorený súbor s hlavičkou obsahujúcu potrebné informácie. Túto možnosť si nechávam na prípadné rozšírenie do inžinierskeho štúdia. Ako ďalšie rozšírenie v inžinierskom štúdiu by som chcel do programu zahrnúť aj prácu s gramatikami a prevody medzi gramatikami a automatami alebo Turingovými strojmi.

Program je rozdelený do siedmich súborov `main.cpp` , `program.cpp` , `program.h` `turingov_stroj.cpp` , `turingov_stroj.h` , `prechodova_funkcia.cpp` , `prechodova_funkcia.h`

V `main.cpp` je grafická časť programu kde pracujeme hlavne s knižnicou SDL, kde vykresľujem všetko do okna a prímam vstupy z klávesnice a myši.

V `program.cpp` pracujeme zo súborom v ktorom máme uložený kód Turingovho stroja a s páskou a pracujeme zo samotnou simuláciou. Program `turingov_stroj.cpp` obsahuje metódy na prácu nad samotným Turingovým strojom a `prechodova_funkcia.cpp` obsahuje metódy na prácu nad prechodmi v prechodovej funkcii v Turingovom stroji.

Program pracuje spolu ako celok ktorý je hlavne založený na volaní funkcií. Hlavná funkcia `main` je rozdelená na 2 časti, v prvej pracuje mód `Menu` a v druhej sú volané funkcie pre módy `Turing machine` a `Machine creator`. Takým spôsobom volaním rôznych funkcií sú súbory `main.cpp` a `program.cpp` navzájom po-prepájané pretože funkcie z jedného súboru volajú funkcie z toho druhého.

Pri bežaní celej aplikácie je v pozadí otvorené príkazové okno, ktoré môže byť vhodné pri nazretí na to čo sa nachádza v programe počas jeho behu, ale nie je zmysľané na to aby s ním užívateľ pracoval pri normálnom behu programu.

# Literatúra

- [1] Meduna, A.: Formal Languages and Computation: Models and Their Applications. Taylor & Francis, 2014, ISBN 978-1-4665-1345-7.
- [2] KOZEN, Dexter. Automata and computability. New York: Springer, c1997. ISBN 0387949070.
- [3] Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-[cit. 2016-05-16].  
URL [https://cs.wikipedia.org/wiki/Line%C3%A1rn%C4%9B\\_ohrani%C4%8Den%C3%BD\\_Turing%C5%AFv\\_stroj](https://cs.wikipedia.org/wiki/Line%C3%A1rn%C4%9B_ohrani%C4%8Den%C3%BD_Turing%C5%AFv_stroj)
- [4] Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-[cit. 2016-05-16].  
URL [https://en.wikipedia.org/wiki/Machine\\_that\\_always\\_halts](https://en.wikipedia.org/wiki/Machine_that_always_halts)
- [5] Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-[cit. 2016-05-16].  
URL [https://en.wikipedia.org/wiki/Multitape\\_Turing\\_machine](https://en.wikipedia.org/wiki/Multitape_Turing_machine)

# Príloha A

## Obsah CD

- túto prácu vo formáte PDF
- túto prácu vo formáte ODT
- priečink **projekt** v ktorom sú umiestnené všetky súbory potrebné pre spustenie aplikácie a preklad.

# Príloha B

## Manuál

CD obsahuje priečinok projekt a kde je spustiteľná aplikácia projekt.exe . Žiadne inštalácie nie sú potrebné pretože knižnice SDL sú v tom súbore. Knižnicu SDL som prevzal z <https://www.libsdl.org/> . Zdrojové kódy sú umiestnené v priečinku src okrem main.cpp. Pri spustení pokiaľ si užívateľ nevyberie žiaden program ani pásku tak program automaticky vyberie machine5.m a tape5.t