

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Editor Turingových strojů

Turing machine editor



2010

Petr Budílek

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně.

4. srpen 2010

Petr Budílek

Anotace

Turingův stroj slouží pro modelování algoritmů v teorii vyčíslitelnosti. Tato práce se zabývá implementací grafického editoru takového stroje. Po navržení stroje by v editoru mělo být možné stroji zadat vstupní řetězec a na tomto řetězci pak krokovat samotný výpočet. Dále by zde měla být možnost exportu přechodového diagramu či kroků výpočtu do MetaPost souboru.

Děkuji Mgr. Janu Konečnému, Ph.D. za vedení této bakalářské práce.

Obsah

1. Úvod	8
2. Turingův stroj	9
2.1. Neformální popis	9
2.2. Formální popis	10
2.2.1. Pomocné definice	10
2.2.2. Základní verze TS	10
2.2.3. Varianty Turingova stroje	11
2.3. Grafová reprezentace TS	12
3. Uživatelská příručka	13
3.1. Systémové požadavky	13
3.2. Hlavní části aplikace	13
3.3. Aplikací menu	13
3.3.1. File	14
3.3.2. Edit	15
3.3.3. Export	15
3.3.4. Help	16
3.4. Modální lišta	16
3.5. Kreslicí plátno	16
3.5.1. Mód Draw States	17
3.5.2. Mód Draw Edges	17
3.5.3. Mód Selection	18
3.5.4. Mód Computation	19
3.6. Panel vlastností	20
3.6.1. Mód Draw States	20
3.6.2. Mód Draw Edges	20
3.6.3. Mód Computation	21
3.7. Panel s páskami	22
3.8. Popis MetaPost souboru	22
3.8.1. Figure	22
3.8.2. Animation	24
4. Dokumentace zdrojového kódu	26
4.1. MainForm	27
4.2. Třída TM	28
4.2.1. Výpočet stroje	29
4.2.2. Uložení a načtení stroje	29
4.3. Třída State	29
4.4. Třída TTRow	29
4.5. Třída TTRowFrom	30

4.6. Třída TTRowTo	30
4.7. Třída TTRowTo	30
4.8. Třída Tape	30
4.9. Třída TapeItem	31
4.10. Třída Export	31
4.11. Třída Common.cs	31
Závěr	32
Reference	33
E. Obsah přiloženého CD	34

Seznam obrázků

1.	Základní schéma TS	9
2.	Příklad grafové reprezentace TS	13
3.	Hlavní pohled	14
4.	Menu – sekce File	15
5.	Formulář pro vytvoření nového TS.	15
6.	Menu – sekce Export	15
7.	Formulář pro export animace	16
8.	Modální lišta	16
9.	Stav	17
10.	Kreslení orientované hrany	17
11.	Řídící bod křivky	18
12.	Kreslení smyčky	18
13.	Hromadný výběr	19
14.	Vlastnosti stavu	20
15.	Vlastnosti hrany	21
16.	Ovládací prvky pro řízení výpočtu	21
17.	Panel s páskami	22
18.	UML diagram – hlavní třídy	26

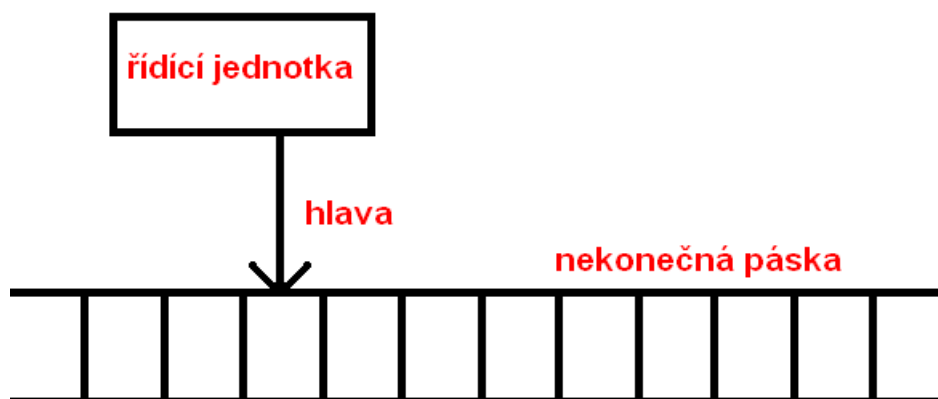
1. Úvod

Implementovaný editor Turingových strojů je určen pro potřeby předmětu Vyčíslitelnost a složitost. Měl by sloužit jak k demonstraci výpočtu Turingova stroje na přednáškách a cvičeních, tak jako pomoc studentům k vypracování zápočtových úloh.

2. Turingův stroj

2.1. Neformální popis

Turingův stroj (TS) je teoretické zařízení sloužící k modelování algoritmů v teorii vyčíslitelnosti. Byl navržen britským matematikem Alanem Turingem v roce 1936. Stroj se skládá z výpočetní jednotky, hlavy a pásky (obr. 1.).



Obrázek 1. Základní schéma TS

Páska symbolizuje paměť, která je v základní verzi zprava neomezená. Je to vlastně nekonečná posloupnost políček, kde každé políčko obsahuje nějaký symbol. Na začátku se na každém políčku nachází prázdný symbol, obvykle je to znak ' '.

Hlava přistupuje k políčkům pásky, ze kterých čte a na které zapisuje symboly. Po přečtení a eventuální změně symbolu se hlava posune buď doprava, nebo doleva.

Výpočetní jednotka obsahuje konečně mnoho stavů a přechodovou tabulku. Mezi stavy se nacházejí tři speciální. Jsou to počáteční, přijímající a zamítající. V přechodové tabulce jsou pak na řádcích uvedena samotná pravidla výpočtu.

Výpočet probíhá následovně. Na začátku se na pásku zapíše vstupní slovo. Jako aktuální stav je nastaven počáteční stav. Hlava se nachází nad prvním políčkem pásky. Nyní z tohoto políčka přečteme symbol. Projdeme přechodovou tabulku, najdeme vyhovující pravidlo pro tuto situaci a podle něj určíme, který symbol na pásku zapíšeme, kam posuneme hlavu a do jakého stavu přejdeme. Tento krok pak opakujeme do té doby, než přejdeme buď do přijímacího nebo zamítacího stavu. V tomto okamžiku výpočet končí a podle typu koncového stavu zjistíme zda vstupní slovo TS přijal, či nepřijal. Pokud se ani do jednoho z výše uvedeného stavu nepodaří dostat, znamená to, že daný TS cyklí.

Zdroje ke kapitole: [1], [4].

2.2. Formální popis

2.2.1. Pomocné definice

Nejprve si nadefinujeme pojmy, se kterými budeme později pracovat.

Definice 1. Abeceda je libovolná konečná neprázdná množina. Její prvky nazýváme symboly.

Definice 2. Řetězem (slovem) v nad abecedou X rozumíme libovolnou konečnou posloupnost symbolů z X , kde $w = a_1 \dots a_n \in X$. Číslo n nazýváme délkou řetězu a značíme $|w| = n$.

Definice 3. Množinu všech řetězů nad abecedou X nazýváme uzávěr množiny X a označujeme X^* .

Definice 4. Jazykem L nad abecedou X nazýváme libovolnou podmnožinu množiny X^* .

2.2.2. Základní verze TS

Definice 5. *Turingův stroj* je uspořádaná sedmice $(Q, \Sigma, \Gamma, \delta, qS, qF, qR)$, kde Q, Σ, Γ a δ jsou konečné množiny a

1. Q je množina stavů
2. Σ je vstupní abeceda neobsahující symbol „-“
3. Γ je pásková abeceda, kde $\Sigma \subset \Gamma$
4. $\delta : Q \times \Sigma \longrightarrow Q \times \Sigma \times \{R, L\}$
5. $qS \in Q$ je počáteční stav,
6. $qF \in Q$ je přijímající stav,
7. $qR \in Q$ je zamítající stav, kde $qF \neq qR$.

Definice 6. Mějme libovolný TS $T = (Q, \Sigma, \Gamma, \delta, qS, qF, qR)$. Každou uspořádanou trojici $(q, \alpha, i) \in Q \times \Sigma^* \times \mathbb{N}$ nazveme konfigurací Turingova stroje T .

Definice 7. Krok výpočtu Turingova stroje T definujeme jako binární relaci \vdash (na množině všech konfigurací) takto: nechť $(q, a_1 \dots a_n, i)$ je taková konfigurace T , kde $q \neq q\pm$, $n \in \mathbb{N}$, $a_1, \dots, a_n \in \Gamma$ a $i \in 1, \dots, n$.

1. Je-li $2 \leq n$ a $\delta(q, a_i) = (q', b, L)$, pak $(q, a_1 \dots a_n, i) \vdash (q', a_1 \dots a_{i-1} b a_{i+1} \dots a_n, i - 1)$.
2. Je-li $\delta(q, a_1) = (q', b, L)$, pak $(q, a_1 \dots a_n, 1) \vdash (q', b a_2 \dots a_n, 1)$.
3. Je-li $\delta(q, a_i) = (q', b, P)$, pak $(q, a_1 \dots a_n, 1) \vdash (q', a_1 \dots a_{i-1} b a_{i+1} \dots a_n, i + 1)$.

Definice 8. *Výpočet Turingova stroje* definujeme prostřednictvím relace \vdash^* (tj. reflexivního a tranzitivního uzávěru relace \vdash). Každý výpočet pro vstupní slovo začíná konfigurací $(qS, w, 1)$ zvanou počáteční.

Definice 9. O každém TS $T = (Q, \Sigma, \Gamma, \delta, qS, qF, qR)$ řekneme, že

- *přijímá* vstupní slovo $w \in \Sigma^*$, právě když $(qS, w, 1) \vdash^* (qF, w', i)$, kde $w' \in \Sigma^*$ a $i \in \mathbb{N}$
- *zamítá* vstupní slovo $w \in \Sigma^*$, právě když $(qS, w, 1) \vdash^* (qR, w', i)$, kde $w' \in \Sigma^*$ a $i \in \mathbb{N}$
- pro vstupní slovo $w \in \Sigma^*$ *cyklí*, právě když w ani nepřijímá, ani nezamítá.

Definice 10. *Jazyk L přijímaný Turingovým strojem $T = (Q, \Sigma, \Gamma, \delta, qS, qF, qR)$* značíme $L(T)$ a definujeme následovně:

$$L(T) = \{w \in \Sigma^*; \text{TS } T \text{ přijímá } w\}.$$

V případě, že TS T pro žádné vstupní slovo $w \in \Sigma^*$ necyklí, říkáme, že T *rozhoduje* jazyk $L(T)$.

Definice 11. Dva stroje jsou *ekvivalentní*, právě když přijímají nebo rozhodují stejný jazyk.

Definice převzaty z [2] a [1].

2.2.3. Varianty Turingova stroje

Výše uvedená definice TS není jediná možná. Existuje mnoho různých typů Turingova stroje, které jsou ale ekvivalentní. Nyní si definujeme několik vylepšení oproti základní definici, která nám v praxi zjednoduší návrh samotného TS.

Ve výše uvedené definici TS se v každém kroku výpočtu musí hlava pohnout doleva, nebo doprava. Jako rozumná se jeví možnost nechat hlavu stát na místě. Změnu provedeme v přechodové funkci δ následovně:

$$\delta : Q \times \Sigma \longrightarrow Q \times \Sigma \times \{R, L, S\}$$

kde S znamená „Stát“.

Pokud se hlava nachází nad prvním políčkem pásky, může se v nynější definici pohnout pouze doprava. Toto omezení nyní odstraníme zavedením *oboustranně nekonečné pásky*. Bude si to vyžadovat změnu definice kroku výpočtu. Definice se nám odstraněním tohoto omezení zjednoduší.

Definice 12. *Krok výpočtu Turingova stroje T* definujeme jako binární relaci \vdash (na množině všech konfigurací) takto: nechť $(q, a_1 \dots a_n, i)$ je taková konfigurace T , kde $q \neq q_{\pm}$, $n \in \mathbb{N}$, $a_1, \dots, a_n \in \Gamma$ a $i \in 1, \dots, n$.

1. Je-li $\delta(q, a_1) = (q', b, L)$, pak $(q, a_1 \dots a_n, 1) \vdash (q', ba_2 \dots a_n, 1)$.
2. Je-li $\delta(q, a_i) = (q', b, R)$, pak $(q, a_1 \dots a_n, 1) \vdash (q', a_1 \dots a_{i-1}ba_{i+1} \dots a_n, i + 1)$.

Poslední rozšíření se bude týkat počtu pásek a jejich hlav. V původní verzi TS je přípustná jen jedna páska. Při zobecnění na k možných pásek dostáváme *vícepáskový TS*. U stroje pak musíme znovu upravit definici přechodové funkce a to následovně:

$$\delta : Q \times \Sigma^k \longrightarrow Q \times \Sigma^k \times \{R, L, S\}^k$$

S tím, že počáteční vstup se zapíše pouze na 1. pásku.

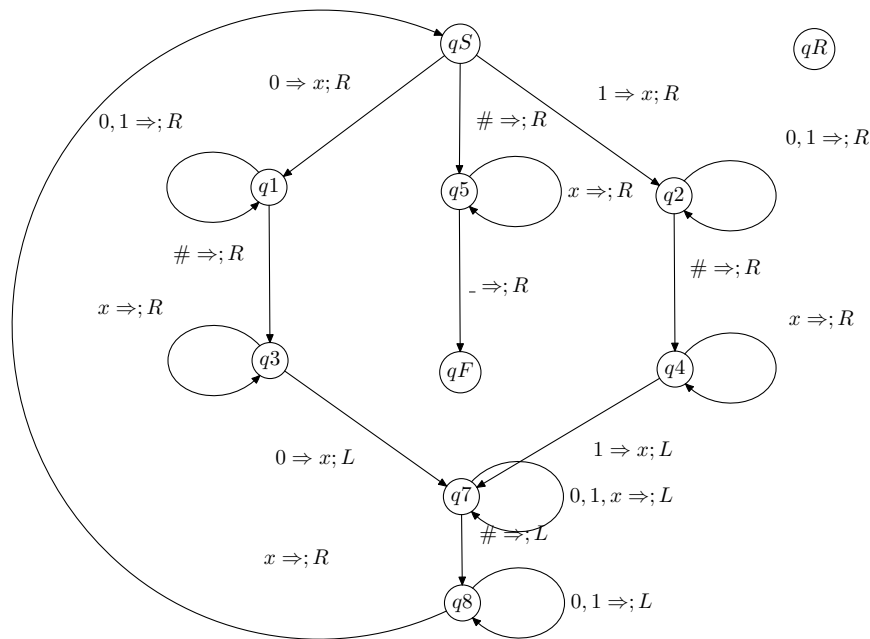
Důkazy, že tato rozšíření a odstranění omezení nemění třídu přijímaných jazyků můžeme nalézt např. v [1].

Model TS se všemi těmito rozšířeními je základním modelem používaným v implementovaném *Editoru Turingova stroje*. Vstupní a pásková abeceda obsahuje všechny znaky anglické abecedy a vesměs všechny standardní symboly s výjimkou '*' a ' ' - mezery.

Příklad takového TS můžeme (již v grafové reprezentaci) vidět na obrázku 2. Daný stroj rozhoduje jazyk $L = \{w\#w \mid w \in \{0, 1\}^*\}$.

2.3. Grafová reprezentace TS

Grafová reprezentace TS je taková reprezentace, kde jsou stavy reprezentovány vrcholy grafu a přechodová funkce orientovanými hranami mezi nimi (obr. 2.).



Obrázek 2. Příklad grafové reprezentace TS

3. Uživatelská příručka

3.1. Systémové požadavky

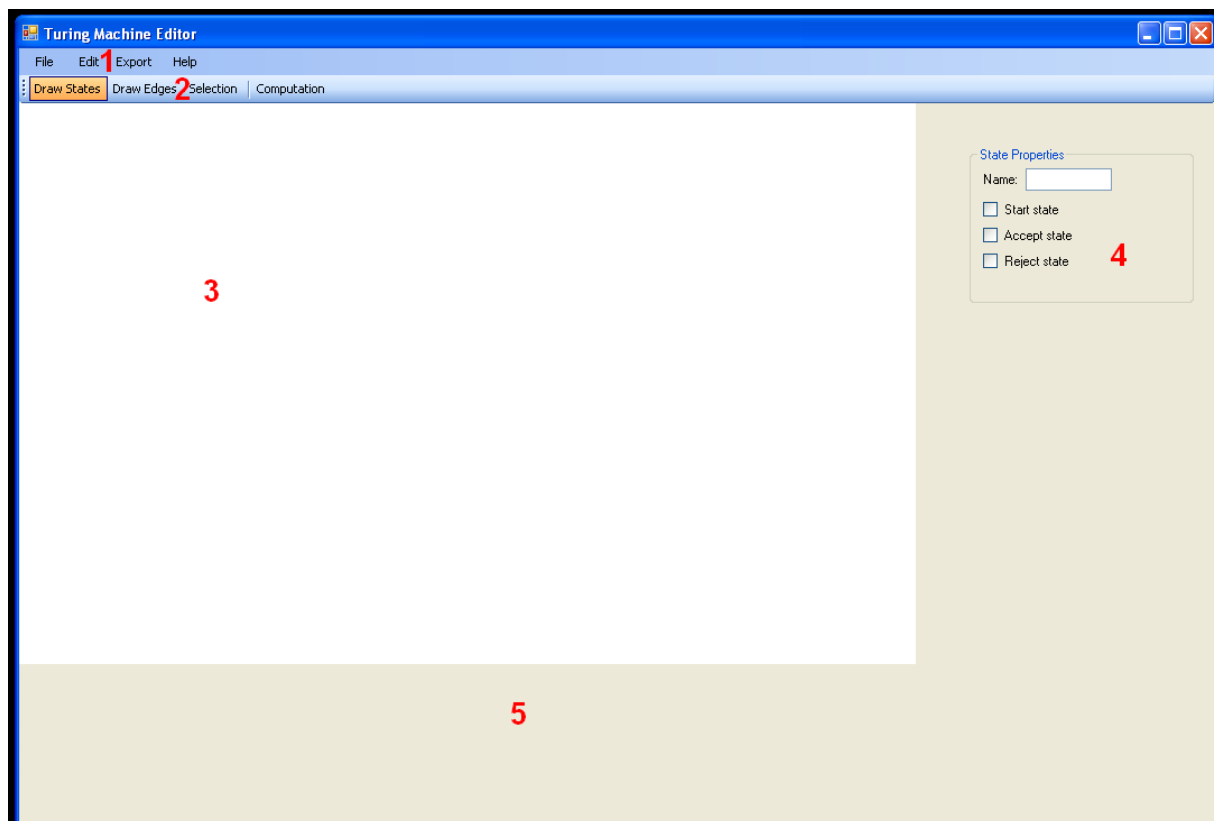
Aplikace vyžaduje operační systém Windows XP (a vyšší verze) s nainstalovaným frameworkem .NET 3.5.

3.2. Hlavní části aplikace

Popis obrázku 3.:

1. Aplikační menu.
2. Modální lišta. Uživatel si zde vybírá, jaké typy úprav si přeje provádět.
3. Kreslicí plátno. Podle zvoleného módu se určí, co se dá na plátno nakreslit.
4. Panel vlastností. Zde se podle zvoleného módu zadávají vlastnosti stavů, přechodové funkce nebo samotného výpočtu.
5. Panel obsahující pásy Turingova stroje. Jsou vidět pouze když se aplikace nachází v módu Computation. O módech viz níže.

3.3. Aplikační menu



Obrázek 3. Hlavní pohled

3.3.1. File

Obsahuje položky umožňující práci se samotným souborem (obr. 4.).

New – Zobrazí formulář umožňující nastavení vlastností nového souboru (obr. 5.).

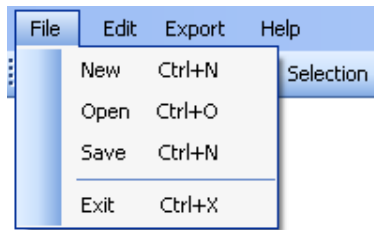
Open – Slouží k otevření již existujícího souboru. Otevírat lze pouze soubory ve formátu TM. Jedná se o binární datový soubor popisující TS.

Save – Slouží k uložení právě upravovaného souboru. Ukládá se opět ve formátu TM.

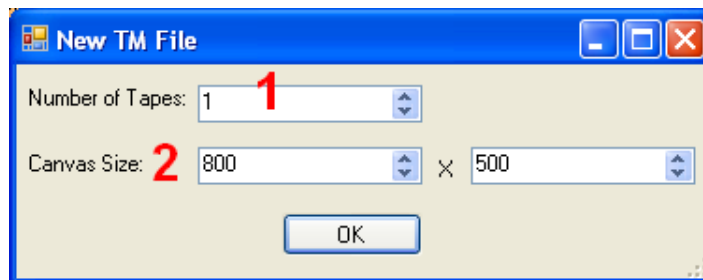
Exit – Ukončí program.

Popis obrázku 5.:

1. Udává počet pásek nově vytvářeného TS.
2. Udává rozměry krelícího plátna



Obrázek 4. Menu – sekce File



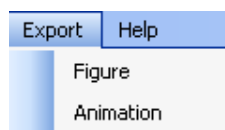
Obrázek 5. Formulář pro vytvoření nového TS.

3.3.2. Edit

Obsahuje položku „Canvas Size“, po jejímž stisknutí se zobrazí formulář, pomocí kterého je možné měnit velikost kreslicího plátna.

3.3.3. Export

Slouží k exportu do MetaPostu. Na výběr jsou dvě položky (obr. 6.).

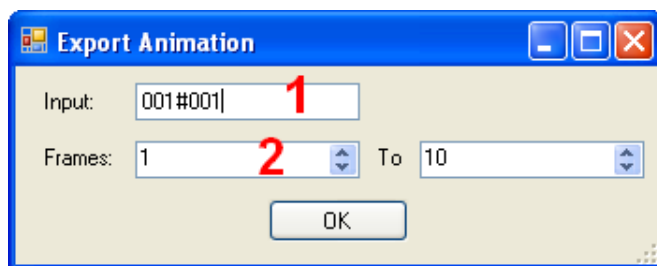


Obrázek 6. Menu – sekce Export

Popis obrázku 6.:

Figure – Exportuje jen grafový diagram TS, bez pásky a bez zvýraznění aktuálního stavu.

Animation – Zobrazí formulář umožňující nastavení vlastností exportované animace (obr. 7.).



Obrázek 7. Formulář pro export animace

Popis obrázku 7.:

1. Textbox obsahuje vstupní řetězec. Tedy řetězec, který se před začátkem výpočtu zapíše na první pásku.
2. Udává rozmezí snímků „od-do“ exportované animace.

3.3.4. Help

Spustí nápovědu. Ke spuštění je potřeba mít nainstalován prohlížeč PDF souborů.

3.4. Modální lišta

Stiskem určitého tlačítka uživatel vybere mód, podle nějž se bude řídit další chování programu.



Obrázek 8. Modální lišta

Draw states – zadávání stavů a jejich editace.

Draw edges – zadávání a editace řádků přechodové funkce.

Selection – hromadné výběry a pokročilejší editace.

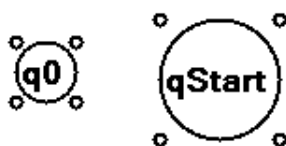
Computation – zadávání údajů potřebných k výpočtu TS a řízení samotného výpočtu.

3.5. Kreslicí plátno

Akce proveditelné na kreslicím plátně souvisí s tím, jaký mód je zrovna aktivní.

3.5.1. M3d Draw States

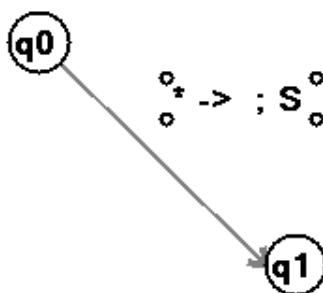
Po kliknutí do prázdného prostoru se vykreslí kruh symbolizující stav výpočtu TS. Uvnitř kruhu se nachází jméno stavu. Čím je jméno stavu delší, tím je kruh větší (obr. 9.). Pokud uživatel klikne do prostoru, kde se již nějaký stav nachází, pak se tento označí jako *vybraný*. Jako vybraný se též nastaví nově vytvořený stav. Pokud stav obsahuje smyčku (viz níže), je tato vybrána s ním. S takto vybraným stavem pak může uživatel v panelu vlastností provádět další úpravy. Dále je možné, při držení levého tlačítka myši, měnit pozici stavu.



Obrázek 9. Stav

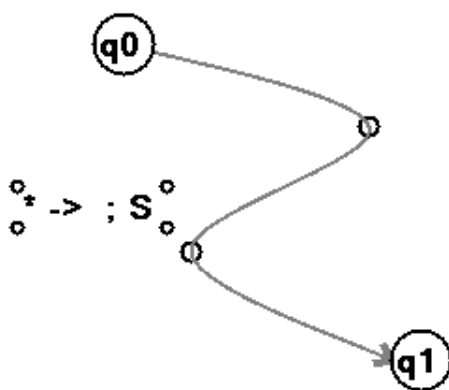
3.5.2. M3d Draw Edges

Vybráním nějakého stavu se začne vykreslovat orientovaná hrana reprezentující řádek přechodové funkce. Tento stav je výchozím stavem dané hrany. Další vybraný stav se pak nastaví jako koncový. Takto vznikne orientovaná hrana mezi těmito dvěma stavy. Tato hrana se pak, stejně jako u stavu, považuje za vybranou. Po vzniku orientované hrany se na její geometrický střed doplní popisec obsahující zbytek dat o aktuálně vybraném řádku přechodové funkce (obr. 10.).



Obrázek 10. Kreslení orientované hrany

Pokud před zadáním koncového stavu klikneme někam do prázdného prostoru, nastavíme tím řídicí bod křivky. Takovýchto bodů může být mezi počátečním a koncovým stavem libovolně mnoho. Křivka je pak postupně těmito body prokládána (obr. 11.). Pro zrušení posledního řídicího bodu je třeba stisknout pravé tlačítko myši, pro zrušení celé hrany pak klávesu ESC.



Obrázek 11. Řídicí bod křivky

Při kreslení hrany, která má stejný počáteční i koncový stav, v terminologii teorie grafů se jedná o smyčku, pak stačí jako řídicí bod křivky vybrat směr, jakým se má smyčka vykreslit a následně opět původní stav (obr. 12.).

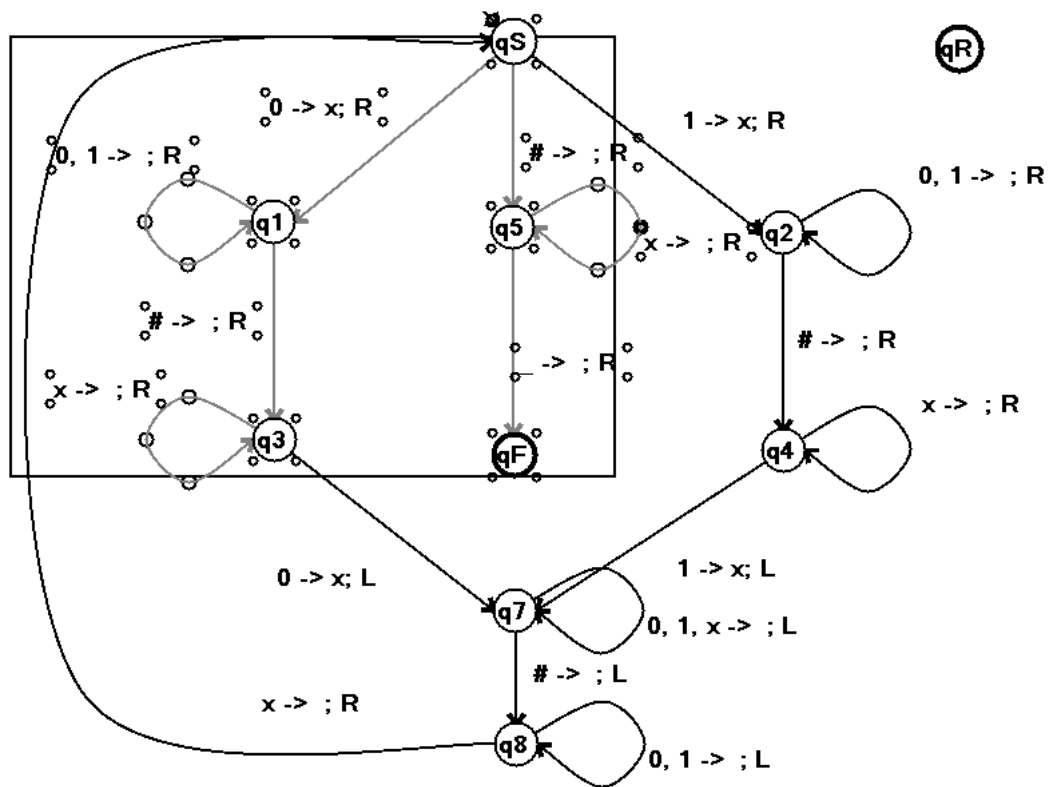


Obrázek 12. Kreslení smyčky

Pokud v tomto módu klikneme na popisek již vytvořené hrany, považuje se, hrana za vybranou. Po stisku levého tlačítka a následném pohybu kurzoru myši je možné měnit pozici popisku hrany. Stejně tak je možné měnit pozici řídicích bodů.

3.5.3. Mód Selection

V tomto módu je možné provádět tzv. hromadný výběr. Hromadný výběr provedeme prostým klikáním na jednotlivé grafické objekty nebo pomocí selekčního obdelníku (obr. 13.). Tyto dva způsoby lze kombinovat.



Obrázek 13. Hromadný výběr

Po provedení výběru lze všem označeným objektům najednou měnit pozici nebo je mazat klávesou Delete. Výběr lze zrušit kliknutím do prázdného prostoru.

3.5.4. Mód Computation

V tomto módu se formou žlutého obarvení určitého stavu uživatel dozví, v jakém stavu výpočtu se TS nachází. Přijímající stav je obarven zeleně, zamítající pak červeně.

3.6. Panel vlastností

3.6.1. Mód Draw States

V tomto módu se na panelu zobrazí vlastnosti právě vybraného stavu (obr. 14.).



Obrázek 14. Vlastnosti stavu

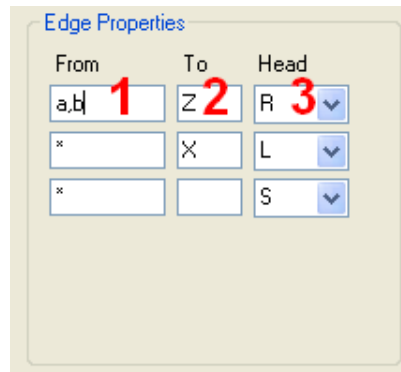
1. Textové pole obsahující jméno stavu. Změna se ihned projeví i na plátně.
2. Checkboxy značící, zda je daný stav počáteční, přijímající, nebo zamítající (v tomto pořadí). Pokud už nějakému jinému stavu byla v historii některá vlastnost nastavena, pak je u něj zrušena. Po zaškrtnutí některého z checkboxů se též změní jméno stavu na qS, qF nebo qR podle toho, který checkbox uživatel zaškrtnl.

3.6.2. Mód Draw Edges

V tomto módu se na panelu zobrazí vlastnosti právě vybrané hrany (obr. 15.). Každý řádek zde popisuje chování hlavy na té které pásce. Počet řádků se tedy rovná počtu pásek.

Popis obrázku 15.:

1. Textové pole obsahuje symboly, které, pokud se nějaký z nich vyskytuje na pásce, jsou nahrazeny symbolem ze 2. Jako speciální symbol je zde vyhrazen '*' - hvězdička. Symbol značí „Jakýkoli symbol.“
2. Textové pole obsahuje symbol, který bude eventuálně zapsán na pásku. Jako speciální symbol je zde vyhrazen ' ' - mezera. Symbol značí „Nezapiš nic.“
3. Combobox obsahuje tři možnosti pohybu hlavy po tom, co zapsala (či nezapsala, viz speciální symbol ' ') nějaký symbol na pásku. Možnosti jsou doprava (*R*), doleva (*L*) a stůj (*S*).



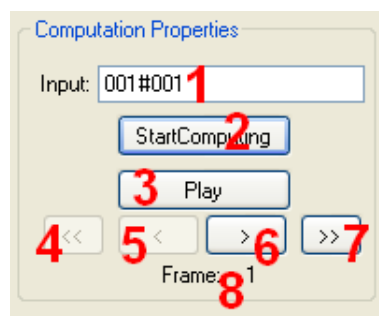
Obrázek 15. Vlastnosti hrany

V příkladu na obrázku 15. se tedy vlastně říká (postupně podle řádků):

1. Přepiš symbol a nebo b symbol z první pásky na symbol Z a hlavu posuň doprava.
2. Přepiš jakýkoli symbol na X a hlavu posuň doleva.
3. Jakýkoli symbol ponechej a hlavu nech stát.

3.6.3. Mód Computation

V tomto módu se na panelu zobrazí ovládací prvky sloužící pro samotné řízení výpočtu (obr. 16.).



Obrázek 16. Ovládací prvky pro řízení výpočtu

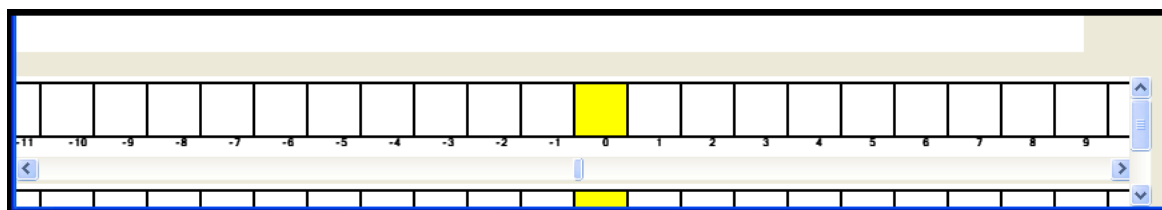
Popis k obrázku 16.:

1. Textbox obsahuje vstupní řetězec.
2. Odstartuje samotný výpočet. Nastaví počáteční stav jako aktuální a na první pásku na políčko s indexem 0 vloží vstupní řetězec.

3. Spustí animaci. To znamená, že každých půl vteřiny se provede jeden krok výpočtu TS. Pokud je animace spuštěna, změní toto tlačítko svůj popis i svou funkci na „Pause“. Pokud se výpočet dostane do jednoho z koncových stavů, animace končí.
4. Provede skok na začátek výpočtu.
5. Vrábí výpočet o jeden krok zpět.
6. Provede jeden krok výpočtu TS. Pokud se výpočet dostane do jednoho z koncových stavů, něděje se nic.
7. Provede skok na úplný konec výpočtu. To jest, když se výpočet nachází v jednom z koncových stavů.
8. Značí, kolik již proběhlo kroků od začátku výpočtu TS.

3.7. Panel s páskami

V případě, že se aplikace nachází v módu Computation, zobrazí se pásky TS, viz obr. 17.



Obrázek 17. Panel s páskami

3.8. Popis MetaPost souboru

Obsah vyexportovaného MP souboru závisí na tom, zda exportujeme grafový diagram – Figure, nebo animaci – Animation.

3.8.1. Figure

Jak již bylo řečeno, při exportu Figure se exportuje pouze grafový diagram. Nyní si na příkladu popíšeme jednotlivé části vyexportovaného souboru.

Hlavička:

```
input boxes;
verbatimtex
etex
h=0.5pt; w=0.5pt;
```

Nejdůležitější částí hlavičky je pravděpodobně import balíčku `boxes`, který nám umožní používat makra pro kreslení tzv. boxů. Boxem je zde myšlen geometrický útvar uvnitř obsahující nějaký text. Informace o tom, jak s boxy pracovat, jsem bral z [3]. Dále si nadefinujeme proměnné `h` a `w`, které později použijeme vždy, když nějakému grafickému objektu budeme udávat souřadnice. Proměnné tedy vlastně slouží k udání měřítka celého výsledného obrázku.

Dále si nadefinujeme jednotlivé kruhové boxy a určíme jejich pozice. Tyto boxy reprezentují stavy TS. Boxy následně pomocí příkazu `drawboxed` vykreslíme.

```
beginfig (1);
circleit.qS(btex $qS$ etex);
qS.c = (382w,565h);

circleit.q1(btex $q1$ etex);
q1.c = (202w,430h);

circleit.q2(btex $q2$ etex);
q2.c = (583w,422h);

circleit.qF(btex $qF$ etex);
qF.c = (382w,256h);

circleit.qR(btex $qR$ etex);
qR.c = (715w,560h);

pickup pencircle scaled 0.5pt;
drawboxed(qS, q1, q2, qF, qR);
```

Nakonec příkazem `drawarrow` vykreslíme hrany mezi stavy, příkazem `label` vykreslíme jejich popisky. Pomocí `bpath` pak přistupujeme k samotnému rámci boxu.

```
drawarrow qS.c..q1.c cutbefore bpath qS cutafter bpath q1;
label.rt(btex \vbox{\hbox{$0 \rightarrow x; R$} } etex,(196w,527h));

drawarrow qS.c..q2.c cutbefore bpath qS cutafter bpath q2;
label.rt(btex \vbox{\hbox{$1 \rightarrow x; R$} } etex,(505w,517h));
```

```

drawarrow q2.c..q4.c cutbefore bpath q2 cutafter bpath q4;
label.rt(btex \vbox{\hbox{\$ \# \Rightarrow ; R\$} } etex,(592w,353h));
endfig;
end

```

3.8.2. Animation

Při exportu animace je každý její snímek tvořen grafovým diagramem (jako u figure), obarveným stavem (barví se stejnými barvami jako při animaci v programu, viz výše) ve kterém se TS nachází a jednou, či více páskami. Vykreslení snímku je vždy ohraničeno dvojicí příkazů:

```

beginfig (1);

endfig;

```

Kde parametr u `beginfig` určuje index snímku. Eventuální obarvení stavu se provádí příkazem `fill` a nachází se těsně před vykreslením boxů.

```

beginfig (1);
circleit.qR(btex $qR$ etex);
qR.c = (715w,560h);

fill bpath qS withcolor (1,1,0);

pickup pencircle scaled 0.5pt;
drawboxed(qS, q1, q2, qF, qR);

```

Jednotlivá políčka pásky definujeme až po kompletním vykreslení grafového diagramu. Začínáme příkazem `boxjoin`, který pomocí zadané rovnice provede to, že boxy budou horizontálně naskládány jeden vedle druhého. Následně boxy vykreslíme. Vykreslovaných políček je vždy 40. Pro ukázkou je však v příkladu použito pouze prvních 5. Následně pomocí příkazu `label.top(btex ∇ etex, cell20.n)}` vykreslíme hlavu TS. Poslední parameter `cell20.n` nám říká, kde se hlava nachází. Úplně nakonec pak vykreslíme indexy jednotlivých políček, opět příkazem `label`.

```

drawarrow q2.c..q4.c cutbefore bpath q2 cutafter bpath q4;
label.rt(btex \vbox{\hbox{\$ \# \Rightarrow ; R\$} } etex,(592w,353h));

boxjoin(a.se=b.sw; a.ne=b.nw);
boxit.cell0(btex $\_$ etex);
boxit.cell1(btex $\_$ etex);
boxit.cell2(btex $\_$ etex);
boxit.cell3(btex $\_$ etex);

```



```
boxit.cell4(btex $\_ $ etex);

drawboxed(cell0, cell1, cell2, cell3, cell4);
label.top(btex $\nabla$ etex, cell20.n);

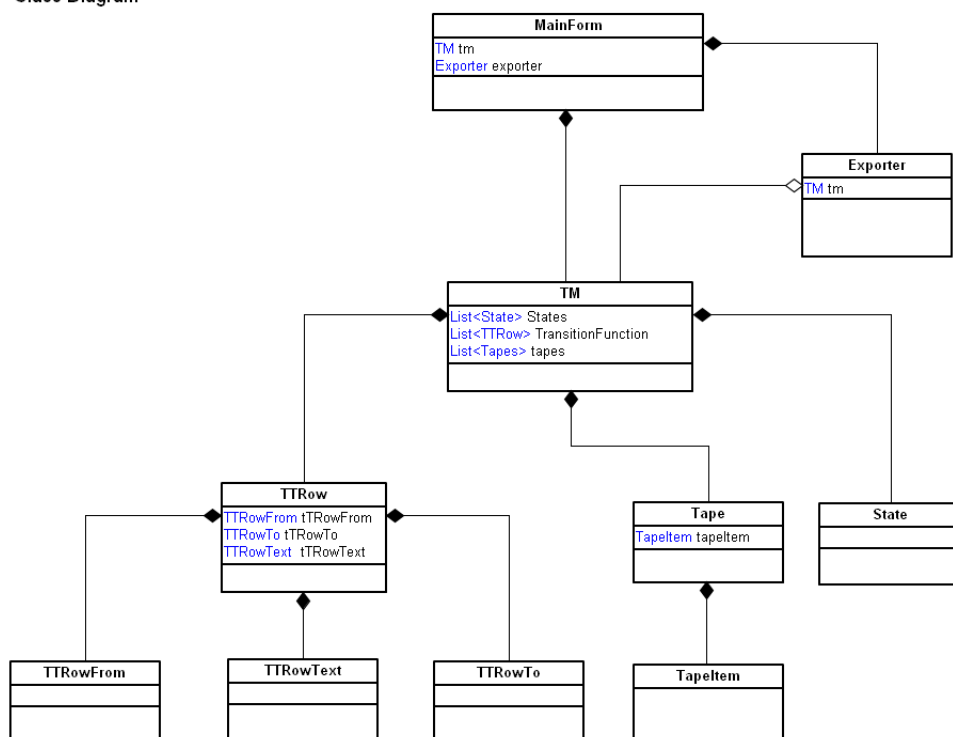
label.bot(btex $_{-20}$ etex scaled .6, cell0.s);
label.bot(btex $_{-19}$ etex scaled .6, cell1.s);
label.bot(btex $_{-18}$ etex scaled .6, cell2.s);
label.bot(btex $_{-17}$ etex scaled .6, cell3.s);
endfig;
```

4. Dokumentace zdrojového kódu

Aplikace je napsána v jazyku C# za použití frameworku .NET verze 3.5. Vývoj probíhal ve studiu Microsoft Visual C# Express Edition. Jako referenční příručku jsem použil [5].

Aplikace je rozdělena do několika tříd. Z nichž každá je pak obvykle uložena v souboru pojmenovaném „NázevTřidy.cs“. UML diagram(obr. 18.) ukazuje rozvržení hlavních tříd.

Class Diagram



create and share your own diagrams at gliffy.com



Obrázek 18. UML diagram – hlavní třídy

Následuje krátký popis každé třídy s výčtem jejich nejdůležitějších slotů a metod. Třídy State, TTRow, TTRowText, TapeItem a Tape reprezentují grafické objekty. Všechny obsahují slot ZeroPoint, což je instance třídy Point a označuje levý horní roh grafického objektu, a nějakým způsobem implementují metodu void Draw(Graphics dc). Metoda slouží k vykreslení objektu na zařízení dc, obvykle se jedná o nějaký PictureBox.

První tři jmenované obsahují navíc metodu `bool IsVisible(Point mousePointer)` a slot `bool Selected`. Metoda `bool IsVisible(Point mousePointer)` vrací `true` v případě, že parametr `mousePointer` geometricky leží v prostoru ohraničeném rámcem grafického objektu (například u `State` je to kruh). Využívá se například při testování, zda bylo na objekt kliknuto. Slot `bool Selected` označuje, zda byl stav vybrán.

4.1. MainForm

Třída obstarává veškerou komunikaci s uživatelem. Zároveň sdružuje všechny ostatní objekty.

Sloty třídy:

`TM tm` – reference na objekt typu `TM`, tj. aktuální upravovaný `TS`.

`Exporter exporter` – reference na objekt typu `Exporter`, který zajišťuje export do `MetaPostu`.

`Mode Mode` – slot výčtového typu `Mode`. Může nabývat hodnot `State`, `TTRow`, `Selection` nebo `Computation`. Při změně módu se pomocí setteru nastaví např. panel vlastností, zobrazí/skryje páska atd.

`State stateFrom`, `stateTo`, `selectedState` – sloty využívané při editaci stavů a kreslení hran.

`bool mouseDown` – značí, zda je stisknuté tlačítko myši, této informace se pak využívá při kreslení grafických objektů.

`Point mouseDownPoint` – souřadnice, kde na kreslícím plátně bylo tlačítko myši stisknuto, stejně jako v předchozím případě se dále používá pro kreslení graf. objektů.

Metody a události třídy:

`void InitializeTapesControls(int numberOfTapes)` – inicializuje ovladací prvky pro jednotlivé pásy, jejich počet závisí na vstupním parametru `numberOfTapes`.

`InitializeTTRowPropertiesComponents()` – inicializuje příslušný počet ovladacích prvků pro zadávání řádků přechodové funkce.

`void UpdateStatePropertiesPanel()` – aktualizuje stavový panel vlastností. Využívá se například při vybrání nějakého stavu.

`void UpdateTTRowPropertiesPanel()` – stejně jako v předchozím případě, akorát pro panel vlastností hran.

`void RedrawTapes()` – aktualizuje pictureboxy všech pásek.

`string StateNameGenerator()` – metoda pro generování jmen nově vytvářených stavů.

`void canvas_MouseDown(...)` - událost formulářového prvku `Canvas` (typu `PictureBox`). Po vyvolání této události se nejprve nastaví `mouseDown`. Dále se podle příslušného módu zavolá jedna z následujících metod: `void ModeState(Point mousePointer)`, `void ModeTTRow(MouseEventArgs e)`, `void ModeSelection(Point mousePointer)`. Které pak dále řídí proces kreslení graf. objektů.

`void canvas_MouseMove(...)` – událost formulářového prvku `Canvas`. Podle příslušného módu a dalších stavových slotů řídí proces kreslení graf. objektů při pohybu myši.

`void canvas_MouseUp(...)` – událost formulářového prvku `Canvas`. Nastavuje slot `mouseDown` a několik dalších stavových slotů.

V dalších metodách a událostech třídy se pak provádí různá nastavení ovládacích prvků, reakce na vstupy uživatele atd.

4.2. Třída TM

Třída `TM` reprezentuje `TS`. Třída de fakto odráží definici `TS` a implementuje metody pro jeho výpočet. Dále jsou zde metody k uložení a načtení `TS`.

Sloty třídy:

`List<State> States` – seznam referencí na objekty typu `State`. Slouží k uložení stavů `TS`.

`List<TTRow> TransitionFunction` – seznam referencí na objekty typu `TTRow`. Slouží k uložení hran `TS`.

`State StartState, AcceptState, RejectState` – reference na `State`, značí startovní, přijímající a zamítající stav.

`State CurrentState` – reference na `State`, značí aktuální stav.

`List<Tape> Tapes` – seznam referencí na objekty typu `Tape`.

`bool ExportingAnimation` – značí, zda se při výpočtu exportuje animace. Pokud je `true` pak se při výpočtu neaktualizují hlavní formulář a celý výpočet je proveden tzv. v tichosti.

`Stack<StepBack> History` – zásobník, sloužící k vracení se zpět ve výpočtu, obsahuje reference na objekty typu `StepBack`. Každý `StepBack` obsahuje předchozí stav, symboly na všech páskách a poslední pohyb hlavy.

4.2.1. Výpočet stroje

Na začátku se pomocí metody `StartComputation()` nastaví `CurrentState` na referenci ze `StartState`. Zapiše vstupní řetězec na první pásku a vynuluje `History`. Metoda `void DoStep()` provede jeden krok výpočtu TS. Nejprve si zjistí všechny symboly aktuálních políček (těch nad nimiž je hlava). Projde celou `TransitionFunction` a zjistí, jestli k aktuální konfiguraci existuje vyhovující pravidlo. Pokud existuje, stroj podle něj provede změnu stavu, zápis symbolů na pásku a konečně posun hlavy. V opačném případě jako `CurrentState` nastaví referenci na stav z `RejectState`.

Metoda `void DoStepBack()` z vrcholu zásobníku sebere poslední `StepBack` a podle jeho obsahu provede potřebné změny v konfiguraci.

4.2.2. Uložení a načtení stroje

Uložení TS se provádí metodou `void Save(string path)`. Metoda nejprve vytvoří serializovatelnou verzi `TM`, to jest verzi (třída `SavedTM`) obsahující pouze data potřebná pro rekonstrukci původního `TM`. Tuto následně pomocí objektu typu `BinaryFormatter` serializuje a uloží ve formě souboru na místo určené proměnnou `path`.

Načtení TS se pak provádí metodou `void Open(string path)`. Soubor je opět pomocí objektu typu `BinaryFormatter` načten. Serializovaný objekt typu `SavedTM` je deserializován a pomocí informací v něm obsažených jsou nastaveny sloty `TM`.

4.3. Třída State

Třída `State` reprezentuje stav TS.

Sloty třídy:

- `string Name` – jméno stavu
- `int Radius` – poloměr kruhového rámce stavu
- `bool StartState, AcceptState, Reject State` – sloty udávající, zda je stav startovním, přijímajícím nebo zamítajícím

Z metod stojí za zmínku `void SwitchOn()` která při zavolání vyplní vnitřek stavu podle jeho typu (`Start`, `Accept`, `Reject`) buď žlutě, zeleně, nebo červeně.

4.4. Třída TTRow

Třída reprezentuje řádek přechodové funkce (hranu). Obsahuje instance tříd `TTRowFrom`, `TTRowTo` a `TTRowText`. Třída zabezpečuje vykreslení křivky mezi stavy, k tomu si ve slotu `List<CurvePoint>` uchovává seznam jejich řídicích bodů.

V metodě `void Draw(Graphics dc)` se pak mimo kreslení křivky samotné volá `Draw` instance třídy `TTRowText` k vykreslení textu.

4.5. Třída `TTRowFrom`

Třída obsahuje sloty `State State` a `List<List<char>> TapesSymbols`. Seznam seznamů charů v druhém jmenovaném je použit z toho důvodu, že hlava stroje může při čtení reagovat na více symbolů na více páskách. Tyto sloty představují levou část přechodové funkce.

4.6. Třída `TTRowTo`

Třída obsahuje sloty `State State`, `List<char> TapesSymbol` a `List<HeadShift> HeadsShift`. U posledních dvou jmenovaných se jedná o seznamy z důvodu zobecnění na více pásek. Tyto sloty představují pravou část přechodové funkce.

4.7. Třída `TTRowTo`

Třída zodpovědná za vykreslení textu hrany. V konstruktoru dostane reference na objekty tříd `TTRowFrom` a `TTRowTo` a podle těch pak sestavuje výsledný text.

4.8. Třída `Tape`

Třída reprezentuje pásku Turingova stroje.

Sloty třídy:

- `int HeadPosition` – pozice hlavy.
- `List<TapeItem> negativeList`, `positiveList` – seznamy obsahující reference na políčka pásky, `negativeList` slouží pro uložení záporné části, `positiveList` pro uložení kladné.

Metody třídy:

- `List<TapeItem> CurrentList()` – vrací referenci na aktuální seznam, podle toho, zda se hlava nachází v kladné části, nebo záporné (určuje se podle `HeadPosition`).
- `int Index()` – metoda vrací index políčka v aktuálním seznamu.
- `void HeadToRight()` – zvýší `HeadPosition` o 1, pokud jsme na konci některého `positiveListu`, přidá další políčko.
- `void HeadToLeft()` – obdobně jako `void HeadToRight()`, akorát inverzně.

- `void SetTapeContent(string tapeContent)` – nastaví postupně prvních n políček `positiveListu` podle předaného parametru.
- `char GetSymbol()` – vrací symbol z políčka, nad nímž se nachází hlava.
- `void SetSymbol(char symbol)` – nastavuje symbol políčku, nad nímž se nachází hlava.
- `void Draw(Graphics dc)` – vykreslí všechna políčka z obou výše zmíněných seznamů, čímž vykreslí celou pásku.

4.9. Třída `TapeItem`

Třída reprezentuje políčko pásky. Z dosud nezmíněných slotů a metod obsahuje `char Symbol`, říkající, co za symbol se na políčku vyskytuje. Defaultně je to `'.'`.

4.10. Třída `Export`

Třída zajišťuje export do MetaPost souboru (MP souboru).
Sloty třídy:

- `TM tm` – reference na stroj, ze kterého se budou brát data pro sestavení MP souboru.
- `StringBuilder mpFile` – v tomto slotu se postupně vytváří řetězec, který vlastně představuje MP soubor.

Metody třídy:

- `void ExportFigure(...)` – slouží k exportu grafového diagramu. Postupně volá metody `MakeHead()`, `DrawStates()`, `void DrawTTRows()` a `void MakeTail()` a sestavuje tak výsledný MP soubor. Po vytvoření je soubor zapsán na místo dané parametrem `path`.
- `void ExportAnimationFigure(int i)` – pracuje podobně jako `void ExportFigure(...)`, ale na rozdíl od ní volá i `void DrawTapes()`.
- `void ExportAnimation(...)` – slouží k exportu animace. Postupně volá metodu `DoStep()` stroje `tm`, po každém takovém zavolání vytvoří pomocí `void ExportAnimationFigure(int i)` snímek animace. Po vytvoření animace je soubor zapsán na místo dané parametrem `path`.

4.11. Třída `Common.cs`

Třída obsahuje pomocné metody využívané všemi ostatními třídami. Dále jsou zde uvedeny definice výčtových typů `HeadShift` a `Mode`.

Závěr

Většina implementací editorů TS, vyskytujících se na internetu, používá pro zadávání stavů a přechodové funkce čistě textové prostředí. Výstup výpočtu TS je pak též realizován pouze textem. Pokud chce pak člověk stroj prezentovat v podobě grafového diagramu, musí ho celý překreslit v nějakém grafickém editoru. Mojí snahou bylo vytvořit editor, který by umožňoval grafický návrh TS a následnou simulaci jeho výpočtu.

Jako možnosti dalšího vylepšení, na které již však v závěru práce bohužel nezbyl čas, bych viděl přidání funkcí „Undo/Redo“ pro samotné kreslení stroje a vyznačení startovního a koncových stavů v exportovaném MetaPost souboru.

Reference

- [1] Sipser, Michael. *Introduction to the Theory of Computation, Second Edition* MIT, 2006.
- [2] Martinek, Pavel. *Základy teoretické informatiky* Univerzita Palackého, Olomouc, 2006.
- [3] Hobby, John D. *Drawing Boxes with MetaPost*.
- [4] *Turing machine - Wikipedia, the free encyclopedia. 2010, [Online].*
http://en.wikipedia.org/wiki/Turing_machine
- [5] *MSDN. 2010, [Online].* <http://msdn.microsoft.com>

E. Obsah přiloženého CD

`bin/`

Obsahuje program `TURINGMACHINEEDITOR.EXE` a soubor s nápovědou `USERDOC.PDF`.

`doc/`

Obsahuje soubor `BAKALARKA.PDF` a dále zdrojové soubory v archívu `BAKALARKA.ZIP` umožňující sestavení `BAKALARKA.PDF`.

`src/`

Archív `TURINGMACHINEEDITOR.ZIP` obsahuje kompletní zdrojové texty programu.

`readme.txt`

Navíc CD/DVD obsahuje:

`data/`

Obsahuje dva testovací stroje: `infinityRightLeft.tm` a `sipser145.tm`. Každý ve svém vlastním adresáři.

`infinityRightLeft.tm` obsahuje dvě pásky, na první zapisuje symbol a do plus nekonečna, na druhé pak do mínus nekonečna. V jeho adresáři je dále vyexportovaná animace 20-30 snímků, kde je vidět plynulý přechod na políčka „mimo záběr“.

`sipser145.tm` rozhoduje jazyk $L = \{w#w \mid w \in \{0,1\}^*\}$. V jeho adresáři je vyexportován jeho grafový diagram, animace končící přijímáním vstupního řetězce a animace končící zamítnutím vstupního řetězce.

`install/`

Obsahuje instalátor frameworku .NET 3.5.