

# Dokumentace

## Determinizace konečného automatu

### 1 Úvod

Simulace konečných automatů je program, který by měl být schopen zpracovat deterministický i nedeterministický konečný automat. Toho se dosáhne tím, že se automat bude moci nacházet ve více stavech najednou – tzv. *aktivní stavy*. Pokud po zpracování/přečtení celého vstupního řetězce je aktivní alespoň jeden koncový stav, je řetězec přijat.

Program dále umožňuje tvorbu a správnou interpretaci epsilon-pravidel (vymazávací pravidla). Množina pravidel se stejným výchozím i cílovým stavem lze graficky reprezentovat jedinou hranou s rozšířeným ohodnocením na více znaků najednou tzv. *násobné pravidlo*. Prakticky je pak možné, aby se v automatu zapsalo pravidlo nejen pro jeden znak, ale pro celou skupinu znaků. Například všechny číslice (0-9), všechna malá písmena (a-z), množina alfanumerických znaků (0-9,a-z,A-Z).

Mezi transformačními algoritmy nad konečnými automaty má tento nástroj odstranění epsilon-pravidel a determinizaci (vytvoření deterministického konečného automatu z ekvivalentního nedeterministického). Determinizace lze navíc provádět s podrobným didaktickým krokováním.

### 2 Návrh

Aplikace je vyvíjena v Javě. Nevýhodou tohoto řešení je, že Java je interpretovaný jazyk a dále, že musí být instalováno speciální prostředí pro spuštění dané aplikace. Naopak výhodou je, že aplikace se snadněji může v budoucnu převést na aplikaci, která bude v reálném čase fungovat na www stránkách (formou Java apletů).

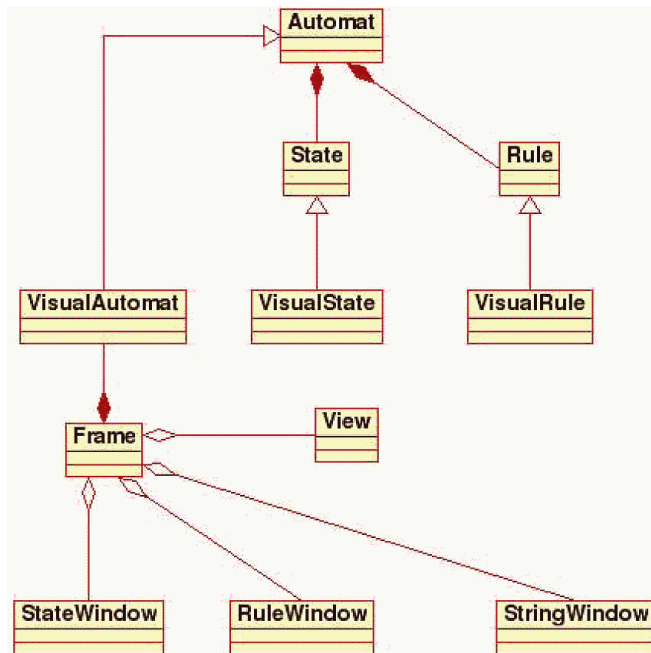
Projekt byl vyvíjen v NetBeans IDE v5.0 a vyšším. Aktuální verze požadovaného virtuálního stroje Java Runtime Environment 6 i použité vývojářské nástroje jsou zdarma k dispozici na <http://java.sun.com> a <http://www.netbeans.org>.

Implementace byla dekomponována na:

1. Vlastní logika automatu.
2. Ovládání a vykreslování automatu.
3. Uživatelské rozhraní, okna.
4. Formát XML, ukládání a načítání automatu do souboru.

## Diagram tříd

Neúplný diagram tříd s nejdůležitějšími třídami je zachycen v následujícím UML diagramu tříd. Popisuje vztahy hlavních tříd v programu.



### Automat, State a Rule

Třída *Automat* reprezentuje samotný konečný automat. Skládá se ze stavů (*State*) a pravidel (*Rule*). V těchto třídách by měla být implementována vlastní logika konečných automatů – především simulace a metoda pro převod na deterministický konečný automat.

### VisualAutomat

Třídy *VisualAutomat*, *VisualState* a *VisualRule* rozšiřují třídy *Automat*, *State*, resp. *Rule* o grafickou reprezentaci. Třída *VisualAutomat* je odpovědná za vykreslení automatu a za ukládání automatu do souboru. *VisualAutomat* sám stavy a pravidla nevykresluje. Místo toho volá patřičné metody objektů tříd *VisualState* a *VisualRule*.

Samotná třída *Automat* je abstraktní. V programu se vyskytují jen „Visual“ objekty. Jistě by bylo možné sloučit tyto třídy a zjednodušit tím návrh. Toto rozdělení však umožňuje oddělit logiku automatu od logiky zobrazování, což přináší mnoho potenciálních výhod.

1. Udržovatelnost – V případě přechodu na jinou platformu/knihovnu bude stačit nahradit jen tu část, která je závislá na dané platformě či grafické knihovně.
2. Oddělení implementace – zmenšení komplexnosti problému vede k jeho lepšímu řešení

### View

*View* reprezentuje okno, které vykresluje automat – tedy pohled na automat. Přijímá zprávy o událostech myši a klávesnice a na základě nich upravuje automat. Spolupracuje především se třídou *VisualAutomat*.

## Frame

*Frame* je hlavní okno aplikace. Obsahuje nástrojovou lištu, menu a všechna další okna. Má na starosti především zpracovávání zpráv z menu a nástrojové lišty a správu ostatních oken.

## StateWindow a RuleWindow

Dialogová okna pro úpravu vlastností stavů a pravidel. Díky nim může uživatel nastavovat název stavu, nastavovat počáteční a koncové stavy či epsilon-pravidla.

## StringWindow

Dialogové okno pro zápis vstupního řetězce. Umožňuje zadávat řetězec přímo a nebo jej načíst z textového souboru.

# 3 Návrh uživatelského rozhraní

Využívá se pouze jediného tlačítka myši – to umožňuje použití programu v typickém prostředí Mac OS X. Podle stisku kláves Shift a Control se mění význam jednotlivých kliknutí.

První myšlenka byla, že na nástrojové liště budou tlačítka pro vytváření stavů a tlačítka pro vytváření pravidel. Takové klikání z plochy na lištu a zpět se však ukázalo jako neefektivní. Proto pro rychlé vytváření stavů a přechodů slouží přímo samotná plocha.

## Událost Význam

Dvojklik na prázdný prostor = Vytvoření stavu

Dvojklik na stav = Nastavení vlastností stavu

Držení klávesy Control = Vytváření pravidel. Stačí klikat postupně na počáteční stav, řídicí body a nakonec na druhý stav.

Dvojklik na šipku pravidla = Vytvoří nový řídicí bod

Dvojklik na řídicí bod = Zrušení řídicího bodu

Dvojklik na popis pravidla = Nastavení vlastností pravidla

Kliknout a táhnout = Přesun stavů, řídicích bodů a popisků pravidel.

## Návrh formátu souborů

Formát souborů je založen na XML. Byl vytvořen co nejjednodušeji tak, aby pokrýval všechny potřebné informace k znovusestavení automatu. Příklad:

```
<?xml version="1.0" encoding="UTF-8" ?>
<automat>
  <state cislo="s1" x="1516" y="1264" first="0" final="1"
    active="0">f</state>
  <state cislo="s2" x="1319" y="1234" first="1" final="0"
    active="0">s</state>
  <rule begin="s2" end="s1" active="0" pos_x="16" pos_y="-35"
    char="A-Z">
    <point x="1416" y="1196"/>
  </rule>
  <rule begin="s2" end="s2" active="0" pos_x="-8" pos_y="56"
    char="0-9">
    <point x="1288" y="1280"/>
    <point x="1352" y="1303"/>
  </rule>
</automat>
```

### ***Element state***

Popisuje stav. Obsah elementu je textový zápis názvu stavu. Název může být libovolný řetězec.

Atributy:

cislo – Unikátní identifikátor. Může být libovolný řetězec

x, y – Souřadnice středu grafické reprezentace tohoto stavu

first – Je tento stav počátečním stavem? (1 = ano, 0 = ne)

final – Je tento stav koncovým stavem? (1 = ano, 0 = ne)

active – Je tento stav aktivní? (1 = ano, 0 = ne)

### ***Element rule***

Popisuje pravidlo. Obsahuje objekty třídy Point, což jsou řídicí body tohoto pravidla.

Atributy:

begin – Stav, ze kterého toto pravidlo vychází.

end – Stav, do kterého toto pravidlo směřuje.

active – Je toto pravidlo právě aktivní? (1 = ano, 0 = ne)

pos\_x, pos\_y – Posun textu s popisem toho pravidla relativně k prvnímu bodu.

char – Množina znaků na které se toto pravidlo aplikuje. Jednotlivé znaky se oddělují čárkou.

Znak mínus označuje skupinu znaků. Například:

Všechny alfanumerické znaky: “a-z,A-Z,0-9”

### ***Element point***

Označuje řídicí bod pravidla.

## **4 Implementace**

### ***Výběr knihoven***

Standardní knihovny jazyka Java s Java Development Kitem verze 6 obsahuje širokou škálu tříd a proto byla pro implementaci využita pouze standardní knihovna prvků grafického uživatelského rozhraní Swing (též standardní součást tříd JDK 6).

### ***Implementace automatu***

Široký výběr tříd pro reprezentaci datových struktur pro automat umožnila pohodlnou implementaci (bez kladení důrazu na efektivitu algoritmů).

### ***Grafická reprezentace***

Pro zobrazení pravidel se využívá přímek a lomených křivek. Pravidla jsou zakončena šipkou do cílového stavu.

Stav je zobrazen jako elipsa opsaná obdélníku s popisem stavu. Výpočet správné velikosti elipsy a nalezení vhodných bodu na okraji elipsy se nakonec ukázalo jako triviální problém analytické geometrie.

### ***Zpracování XML***

Načítání XML je uskutečněno za pomoci Document Object Model (DOM).

# 5 Uživatelský manuál

## 5.1 Ovládání programu

### 5.1.1 Práce se stavy

- *Vytvoření* Stav lze vytvořit dvojklikem levým tlačítkem myši na pracovní plochu.
- *Označení* Označený stav je zeleně vybarven, označení stavu se provádí jedním kliknutím levým tlačítkem myši. Lze označit i více stavů a to držením klávesy "Shift" a klikáním levým tlačítkem myši na stavy, které chcete označit. Chcete-li stav odznačit, klikněte mimo něj.
- *Posunutí* Lze provést jak s označeným tak s neoznačeným stavem, pokud se provádí s neoznačeným stavem, tento stav se označí. Stav se přesune držením levého tlačítka myši a táhnutím na místo, kam chcete stav umístit.
- *Smazání* Nejprve si označte stav/stavy, který/které chcete smazat. Potom stiskněte klávesu "Delete".
- *Editace* Je nutno otevřít editační okno. Toto lze provést třemi způsoby:
  - 1) dvojklikem na příslušný stav
  - 2) označením stavu a stisknutím tlačítka "Stav" v panelu nástrojů
  - 3) označením stavu a následně přes menu: automat,stav

### 5.1.2 Práce s přechody

- *Vytvoření*
  - 1) Bez pomocných bodů: Ctrl + levé tlačítko myši na výchozí start, Ctrl+levé tlačítko myši na cílový stav.
  - 2) S pomocnými body: Ctrl + levé tlačítko myši na výchozí start, Ctrl + levé tlačítko myši na místo pomocného bodu (toto lze opakovat libovolně), Ctrl+levé tlačítko myši na cílový stav. 3) Dovytyvoření pomocného bodu: dvojklikem levým tlačítkem myši na čáru přechodu.
- *Označení*
  - 1) Celého přechodu: klikněte na znaky přechodu levým tlačítkem myši
  - 2) Pomocného bodu: klikněte levým tlačítkem myši na pomocný bod  
Označené objekty se zbarví nazeleno. Chcete-li označit více objektů, držte klávesu "Shift" a levým tlačítkem označujte vybrané objekty.
- *Posunutí*
  - 1) Hrany: Posunout hranu lze tím, že se posouvají pomocné body. Držte levé tlačítko myši a táhněte myš na místo, kam chcete bod umístit.
  - 2) Znaky přechodu: Lze přesunout držením levého tlačítka myši a posunutím na novou pozici.
- *Smazání*
  - 1) Pomocného bodu: označte bod a stiskněte klávesu "Delete".
  - 2) Celého přechodu: označte přechod a stiskněte klávesu "Delete".
- *Editace* – Při editaci je nutno otevřít okno pro editaci přechodu. Toto lze několika způsoby:
  - 1) dvojklikem na znak/znaky příslušící danému přechodu
  - 2) označením přechodu a stisknutím tlačítka "Přechod" v panelu nástrojů
  - 3) označením přechodu a následně přes menu: automat->přechod

## 5.1.3 Nastavení a ovládání simulace

Simulace se ovládá dvojím způsobem:

- 1) Z panelu nástrojů (tlačítka Start, Stop, Reset, Krok, panel pro zadání rychlosti, tlačítko řetězec)
- 2) Přes menu: v položce automat jsou možnosti resetovat automat, Start, Stop, Krok, zadání řetězce

Více o ovládání se dovíte v tématu nápovědy simulace.

## 5.1.4 Možnosti automatu

- *Uložení* Uložení se provádí přes menu: soubor, uložit, tímto soubor uložíte do předem připraveného formátu XML.
- *Nahrání ze souboru* Otevřít soubor, který jste uložili do připraveného formátu XML, lze přes menu: soubor, otevřít.
- *Kontrola determinističnosti* Zkontrolovat, zda je automat deterministický, lze přes menu: automat, zkontrolovat, jestli je DFA.
- *Převedení na deterministický automat* Chcete-li převést automat na deterministický, můžete stisknout tlačítko "DFA" v panelu nástrojů nebo v menu: automat, konvertovat na DFA. Tato konverze je nevratitelná.
- *Odstranění epsilon pravidel* Z automatu lze odstranit epsilon pravidla a to buď tlačítkem "epsilon pravidla" nebo přes menu: automat, odstranit epsilon pravidla.

## 5.1.5 Vyvolání nápovědy

- V menu vyberte položku nápověda, nápověda.
- Označte stav nebo přechod, stiskněte klávesu "F1", otevře se příslušná nápověda buď pro hranu nebo pro přechod.

## 5.2 Stav

### 5.2.1 Co je to stav

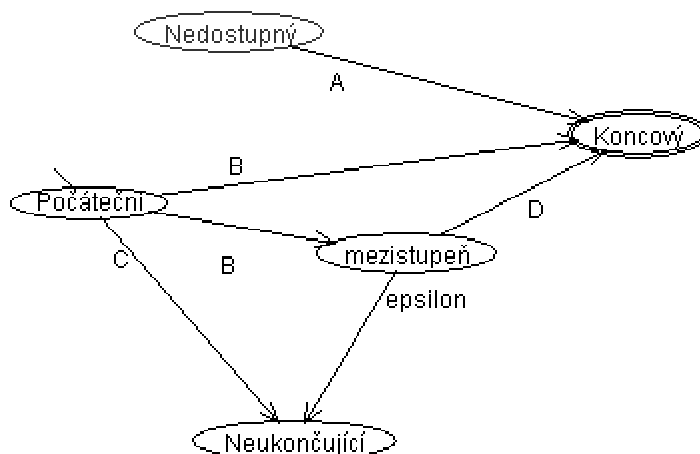
Stav je součástí každého automatu. Každý automat je definován množinou stavů, počátečním stavem, který je jedním ze stavů automatu, a množinou stavů koncových, které jsou podmnožinou množiny všech stavů automatu. Více o automatu a jeho definici.

V některých typech konečných automatů se mohou vyskytovat také nedostupné stavy a stavy neukončující.

- Nedostupné stavy jsou ty, do kterých se nelze žádnou kombinací pravidel dostat z počátečního stavu.
- Neukončující jsou ty stavy, ze kterých se nelze kombinací pravidel dostat do žádného koncového stavu.

Při návrhu automatu se doporučuje stavy, které jsou nedostupné a neukončující, odstranit, neboť v automatu neplní žádnou funkci.

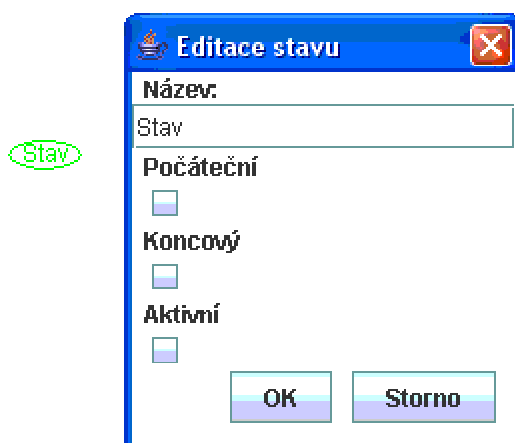
### Příklad nedostupného a neukončujícího stavu



## 5.2.2 Jak vytvořit a ovládat stav v tomto programu

Stav lze vytvořit dvojklikem levého tlačítka myši tam, kde se ještě žádný nenachází. Pokud chcete vytvořený stav přesunout, učiníte tak držením levého tlačítka myši a táhnutím na místo, kam chcete stav přemístit. Stav lze editovat buď dvojklikem levým tlačítkem myši, nebo jeho označením a pak stisknutím tlačítka "Stav" v panelu nástrojů. Pokud potřebujete daný stav smazat, stačí jej označit levým tlačítkem myši, stav by měl být obarven na zeleno, nyní stiskněte klávesu "Delete" na klávesnici a stav bude vymazán. **POZOR!!!** Mažete-li stav, budou vymazány i všechny přechody, které vedly z/do tohoto stavu!!!

### Ukázka okna pro nastavení stavu



## 5.2.3 Nastavitelné vlastnosti stavu

- název - lze vyplnit název stavu, na přijetí řetězce automatem toto nemá vliv, je to pouze orientační označení pro přehlednost automatu
- koncový - pokud je stav koncový, zaškrtněte políčko u "koncový", že je stav nastavený na koncový, poznáte podle toho, že stav je olemován dvojitou čarou
- počáteční - pokud je stav počátečním, zaškrtněte políčko u "počáteční", nastavení stavu na počáteční poznáte podle krátké čáry, která vede šimko doleva nahoru od okraje stavu
- aktivní - využívá se při simulaci, nastavení stavu na aktivní poznáte podle toho, že je zbarven načerveno, pozor ovšem pokud máte nějaký stav označen zelenou barvou, jeho aktivitu zjistíte buď otevřením "State window" z panelu nástrojů, bude-li v otevřeném okně zaškrtnuto políčko "aktivní", stav je aktivní (toto okno lze otevřít dvojitým kliknutím na daný stav), nebo kliknete levým tlačítkem na jiný stav

Stav může být zároveň počáteční i koncový, počáteční stav je v každém automatu pouze jeden, pokud nastavíte počátečních stavů více, budete na toto upozorněni při spuštění simulace.

### Ukázka počátečního, koncového, aktivního a označeného stavu



## 5.3 Přejchod

### 5.3.1 Co je to přechod

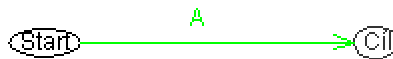
Přejchody v automatu reprezentují pravidla, která jsou určena znaky. Pokud automat přijímá řetězec, posouvá se ze stavu do stavu po přechodech a to po takových, kterým je přiřazen následující znak řetězce. U některých automatů existuje taková hrana jen jedna, někde jich může být více, někde neexistuje žádná. Více o vlastnostech automatů.

### 5.3.2 Jak vytvořit a ovládat přechod v tomto programu

Přejchod se vytvoří stisknutím klávesy "Ctrl", následuje (za stálého držení klávesy "Ctrl") kliknutí levým tlačítkem na stav, ze kterého má přechod vést, a kliknutí levým tlačítkem na stav, do kterého má přechod vést. U vytvořeného přechodu je automaticky vygenerována šipka, která určuje cílový stav. Dále je přechodu automaticky vybráno jedno písmeno abecedy jako přiřazený znak pro přechod z počátečního do cílového stavu. Smazání hrany se provádí označením hrany (kliknutím levého tlačítka myši na znak přechodu) a následným stiskem klávesy "Delete" na klávesnici.

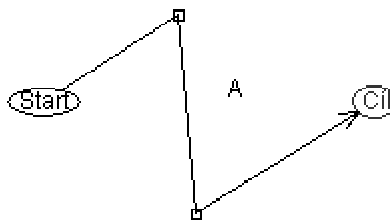


### Příklad hrany s přechodovým znakem "A"



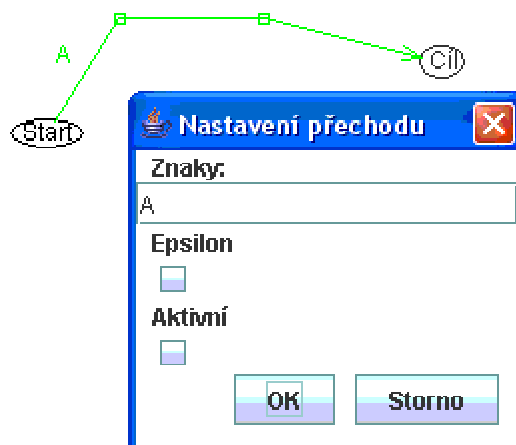
Pokud je to nutné, je možné vytvořit pomocné body, aby se hrany nemusely příliš křížit. V tomto případě po kliknutí na počáteční stav následuje kliknutí do místa, kam chcete pomocný bod umístit a potom na stav, do kterého má přechod vést. Pomocných bodů lze vytvořit i více než jeden. Tyto pomocné body lze dvojklikem levým tlačítkem na daný bod zrušit, držením levého tlačítka a táhnutím lze bod přemístit. Pomocné body lze k přechodu dodat, a to dvojklikem na čáru přechodu.

### Příklad hrany s pomocnými body



Znak přiřazený hraně lze libovolně posunout a to táhnutím myši za držení levého tlačítka. Tento znak lze editovat dvojklikem, nebo kliknutím levým tlačítkem myši a následným stisknutím tlačítka "Přechod" na panelu nástrojů. Tímto se otevře okno, ve kterém lze buď ručně zadat znak/množinu znaků, nebo lze odškrtnout epsilon, čímž se vytvoří tzv. epsilon pravidlo, také lze nastavit aktivitu hrany. Malé okno lze zavřít buď tlačítkem "OK", při tomto zavření se uloží provedené změny, nebo stisknutím tlačítka "Storno", tímto se změny vrátí zpět.

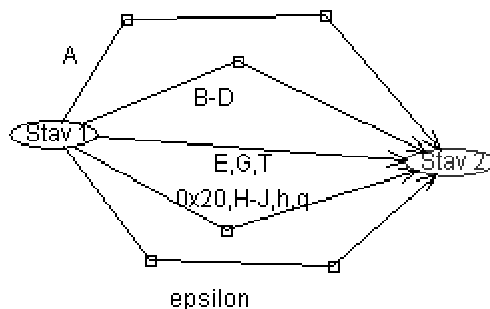
### Ukázka okna pro nastavení přechodu



### 5.3.3 Nastavitelné vlastnosti přechodu

- **znaky** – do tohoto pole se nastavuje množina symbolů, při kterých přejde automat po daném přechodu. Při zaškrtnutí epsilon pravidla je toto ignorováno. Lze zadat typicky jeden znak, ale také množinu znaků a to buď výčtem (př.: A,B,Z) nebo intervalem (př.: A-Z). Tyto dvě možnosti lze také kombinovat (př.: A-D,Q) Lze zadávat i jiné znaky, třeba mezera, v grafu se objeví hexadecimálně podle ASCII tabulky

#### Možné znaky u přechodů



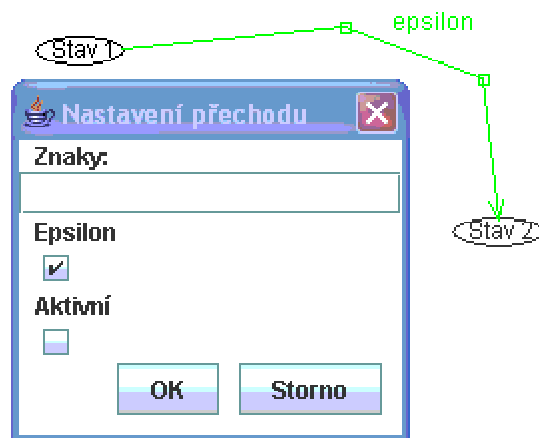
- **epsilon** - při zaškrtnutí je vytvořeno tzv. epsilonové pravidlo, v grafickém vyjádření se projeví slovem "epsilon" v místě znaku pro přechod
- **aktivní** - využívá se při simulaci, v obrázku se objeví jako červená hrana, pokud ovšem není zeleně označena. Pokud je zeleně označena, aktivitu zjistíte kliknutím levým tlačítkem myši jinam, popřípadě stisknutím "Přechod" v panelu nástrojů, kde v případě aktivity bude zaškrtnuto aktivní.

### 5.3.4 Epsilonové pravidlo

Epsilonové pravidlo je takové pravidlo, kde znakem pro přechod z jednoho stavu do druhého je znak epsilon (symbolizující prázdný řetězec). Toto pravidlo funguje tak, že může a nemusí proběhnout. Pokud jsme se použitím nějakého pravidla dostali do stavu, ze kterého vede epsilon pravidlo, dostali jsme se vlastně také do stavu, do kterého toto epsilon pravidlo vede. Je-li toto pravidlo použito v automatu, pak je tento automat nedeterministický (nedeterminističnost spočívá v tom, že nelze rozhodnout, zda se epsilon pravidlo provede nebo ne). Epsilon pravidlo poznáte podle řetězce "epsilon" u přechodu. Nastavit jej lze v okně pro nastavení přechodu zaškrtnutím políčka "epsilon".

Pokud chcete vytvořit ekvivalentní automat bez epsilon pravidel, stiskněte tlačítko "e-pravidla" v panelu nástrojů nebo v menu: automat, odstranit epsilonová pravidla.

## Ukázka epsilonového pravidla

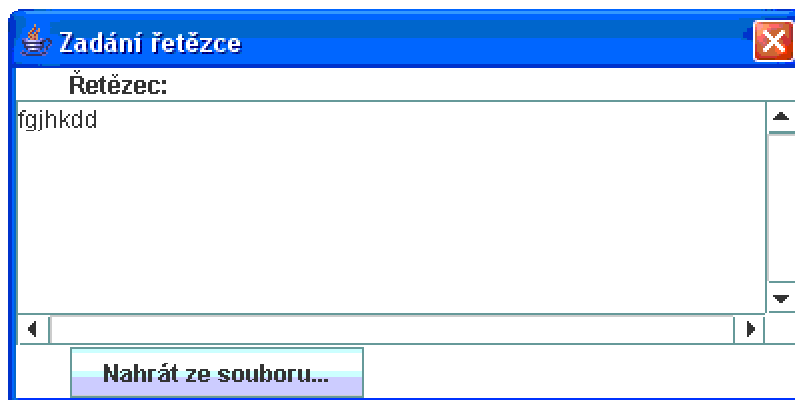


## 5.4 Simulace

### 5.4.1 Zadání vstupního řetězce

Vstupní řetězec se zadává po kliknutí na tlačítko "Řetězec" v panelu nástrojů. Řetězec lze buď načíst z textového souboru stisknutím tlačítka "Nahrát ze souboru", nebo přímo zadáním řetězce do vyhrazeného okénka. Při simulaci je přijímán řetězec, který je aktuálně v políčku "Řetězec". Toto okno lze zavřít stisknutím křížku, přičemž zadaný řetězec zůstane uložen. Toto okno ale může zůstat při simulaci otevřené - lze zde pozorovat postupné přijímání znaků.

### Okno pro zadání řetězce



### 5.4.2 Vlastní simulace

Chcete-li odsimulovat přijetí celého řetězce, je vhodné nejdříve celý automat resetovat a to tlačítkem "Reset" z panelu nástrojů. Dojde k nastavení počátečního stavu na aktivní. Dále stiskněte tlačítko "Start" na panelu nástrojů. Začne probíhat simulace, která se zastaví v okamžiku, je-li řetězec přijat nebo dojde-li k chybě a řetězec již nemůže být automaticky přijat. Zastavit simulaci lze také tlačítkem "Stop". V tuto chvíli, pokud si otevřete okno

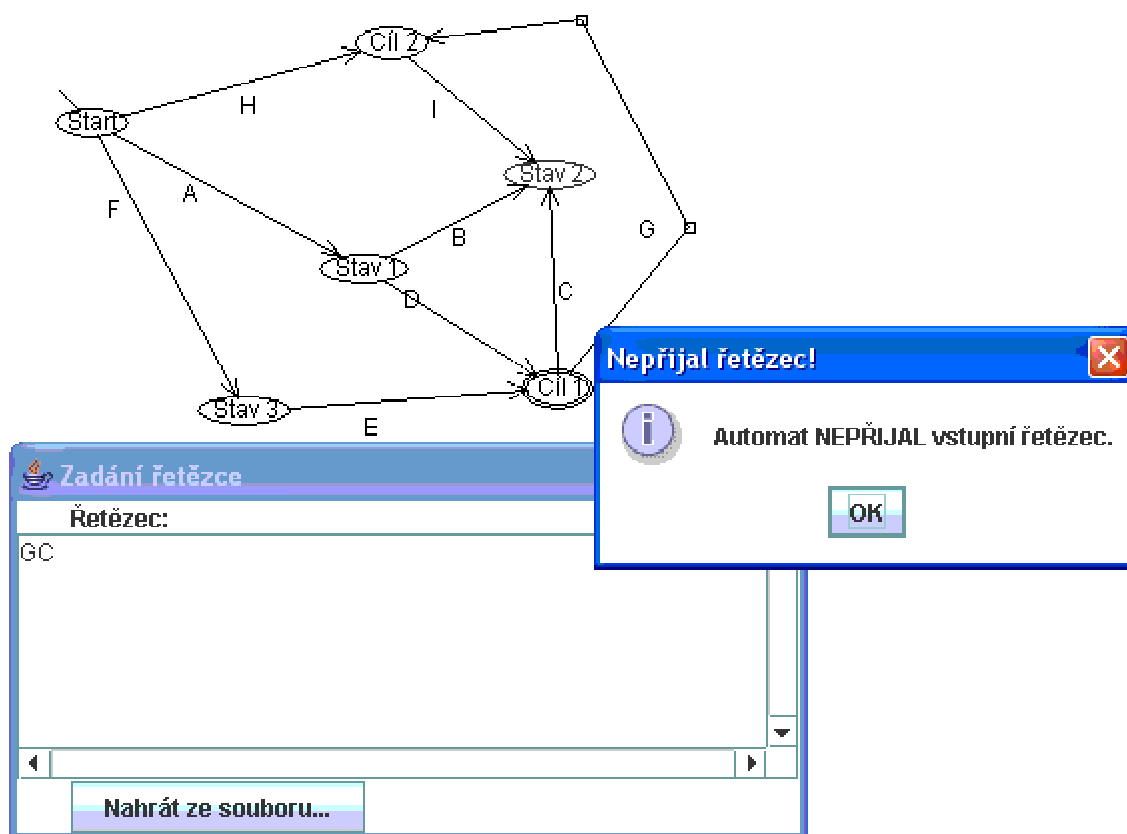
"Řetězec", tak vidíte, jaký řetězec ještě nebyl přijat. V grafu svítí červeně stav, ve kterém se automat nacházel v okamžiku přerušení, a přechod, kterým jsme se do tohoto stavu dostali. Opětovným stisknutím tlačítka "Start" bude simulace pokračovat.

### Panel nástrojů pro simulaci



Pokud simulace zastaví svou činnost a řetězec není přijat, lze zjistit, kde v řetězci se provádění zastavilo a to otevřením okna s řetězcem.

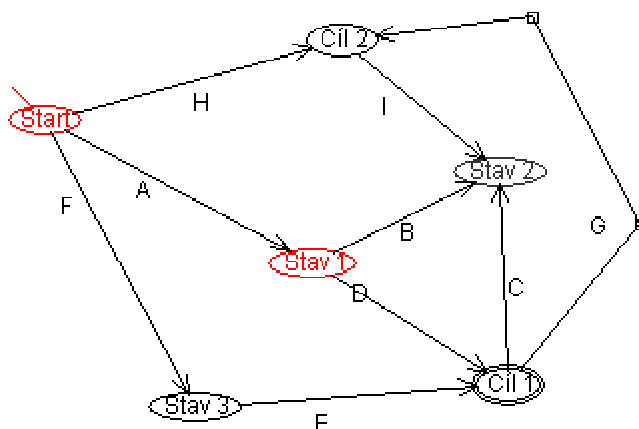
### Zastavení v případě nepřijetí řetězce automatem



### 5.4.3 Simulace odjinud než z počátečního stavu

Při simulaci od jiného stavu, než je počáteční, je nutné nastavit stav, odkud má simulace probíhat, na aktivní. U ostatních stavů je třeba aktivitu odstranit. Lze také označit více stavů za aktivní, simulace pak probíhá z více stavů.

### Příklad nastavení pro simulaci z více stavů



#### 5.4.4 Krokování

Pokud chcete řídit rychlost simulace sami, je možné simulaci krokovat tlačítkem "Krok" v panelu nástrojů. Stisknutím tohoto tlačítka proběhne přijetí jednoho znaku řetězce. Krokovat lze jak od počátečního stavu (před krokováním nutno nastavit počáteční stav na aktivní - lze stisknutím tlačítka "Reset" v panelu nástrojů), tak od kteréhokoliv jiného stavu (tento je třeba nastavit na aktivní manuálně)

**Při simulaci pozor na velká a malá písmena, program je case-senzitive a tedy rozlišuje velká a malá písmena!!!**