

Diplomová práce

- Název: Konverze modelů bezkontextových jazyků
- Autor: Bc. David Navrkal

Obsah

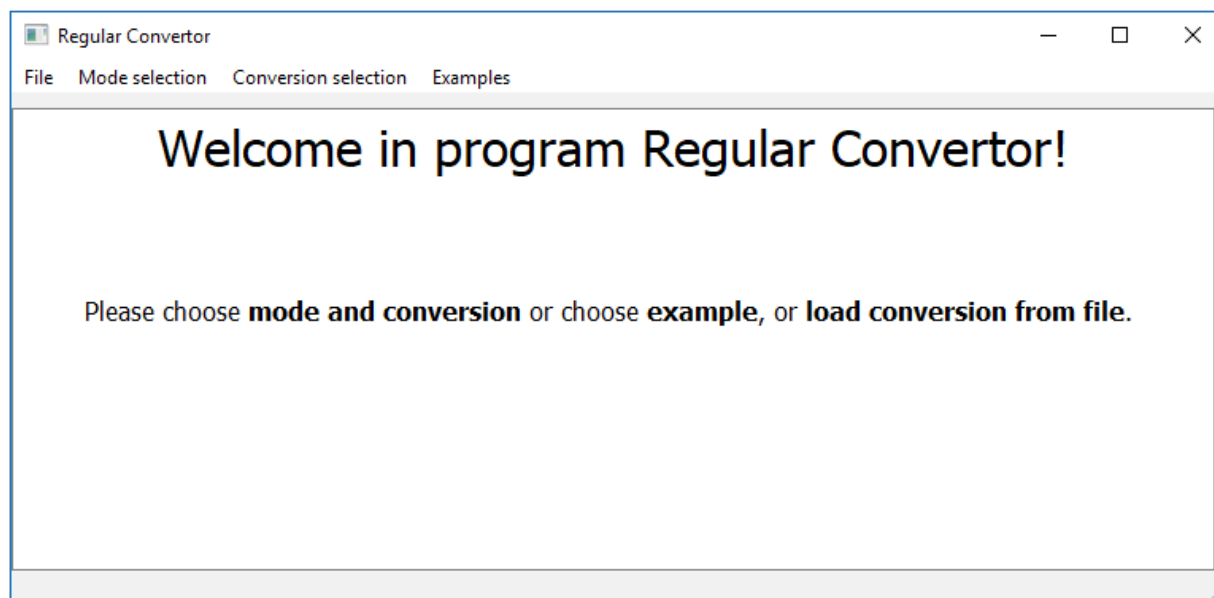
- Zadání
- Demonstrace aplikace
 - Úvodní obrazovka.
 - Výběr ze zabudovaných příkladů.
 - Okno konverze bezkontextové gramatiky (BZK) na zásobníkový automat (ZA).
 - Módy aplikace
 - Krokovací mód.
 - Mód samostatné práce.
 - Mód průběžné kontroly.
 - Ukládání a načítání ze souboru.
 - Editace BZK.
 - Editace ZA.
- Otázky oponenta.
- Porovnání s existujícími řešeními.

Zadání práce

1. Vypracujte literární rešerši na dané téma.
2. Implementujte ve zvoleném programovém prostředí převodní algoritmy mezi bezkontextovou gramatikou a zásobníkovým automatem.
3. Vytvořte uživatelské rozhraní pro převodní algoritmy.
4. Implementujte převodní algoritmy do dříve vytvořené programové aplikace v rámci Bc. práce.
5. Vytvořte databázi příkladů a otestujte v rámci vytvořeného prostředí.

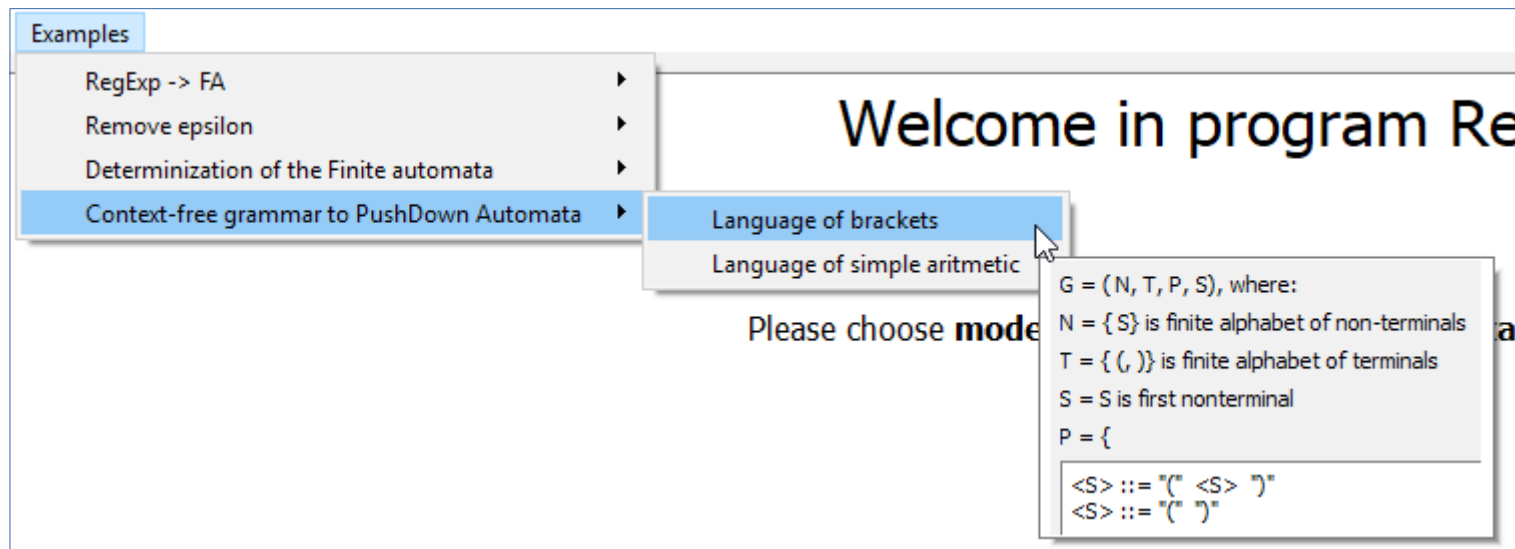
Úvodní obrazovka

- Na úvodní obrazovce se nachází instrukce, jak začít pracovat s programem:
 - Výběr módu a konverze.
 - Výběr z příkladů.
 - Načtení konverze ze souboru.



Výběr z příkladů

- Pro konverzi BZK na ZA jsou vestavěné 2 příklady:
 - 1) První má vstup gramatiku, která popisuje jazyk závorek.
 - 2) Druhá popisuje jazyk jednoduché aritmetiky.

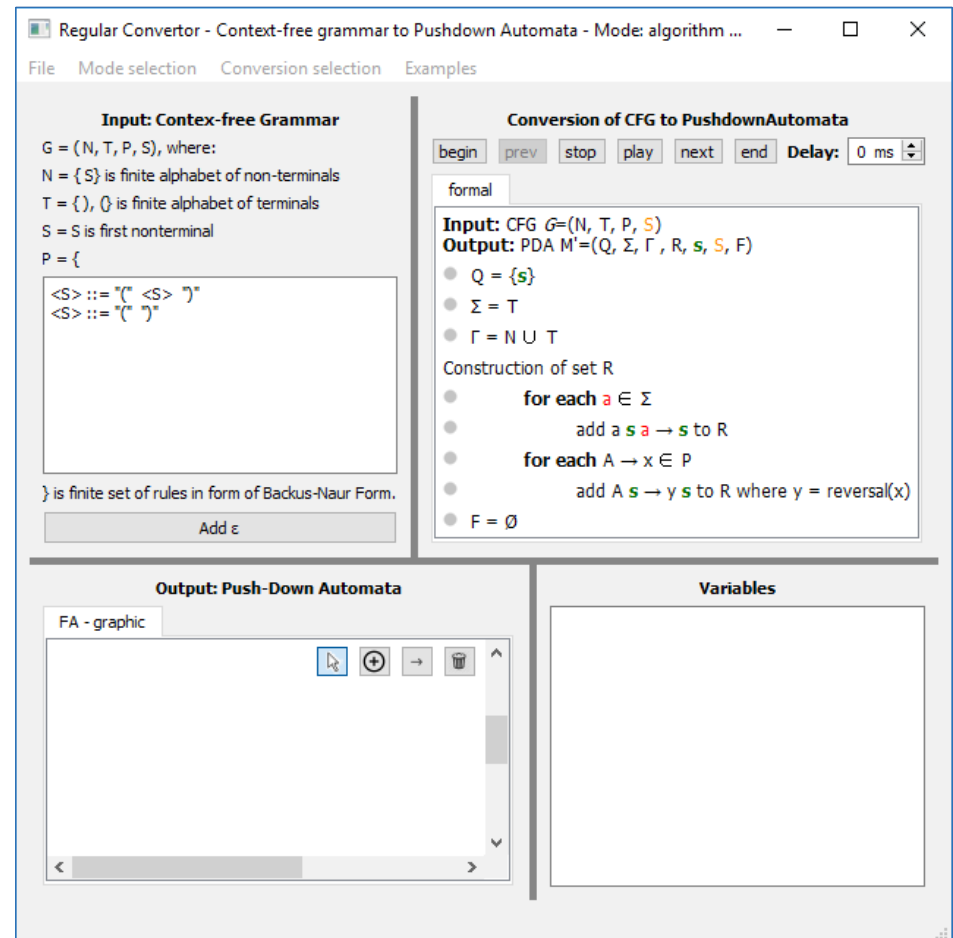


Okno konverze

Skládá se 4 částí (widgetů) pro:

- 1) Bezkontextovou gramatiku.
- 2) Algoritmus převodu.
- 3) Zásobníkový automat.
- 4) Zobrazení proměnných z algoritmického widgetu.

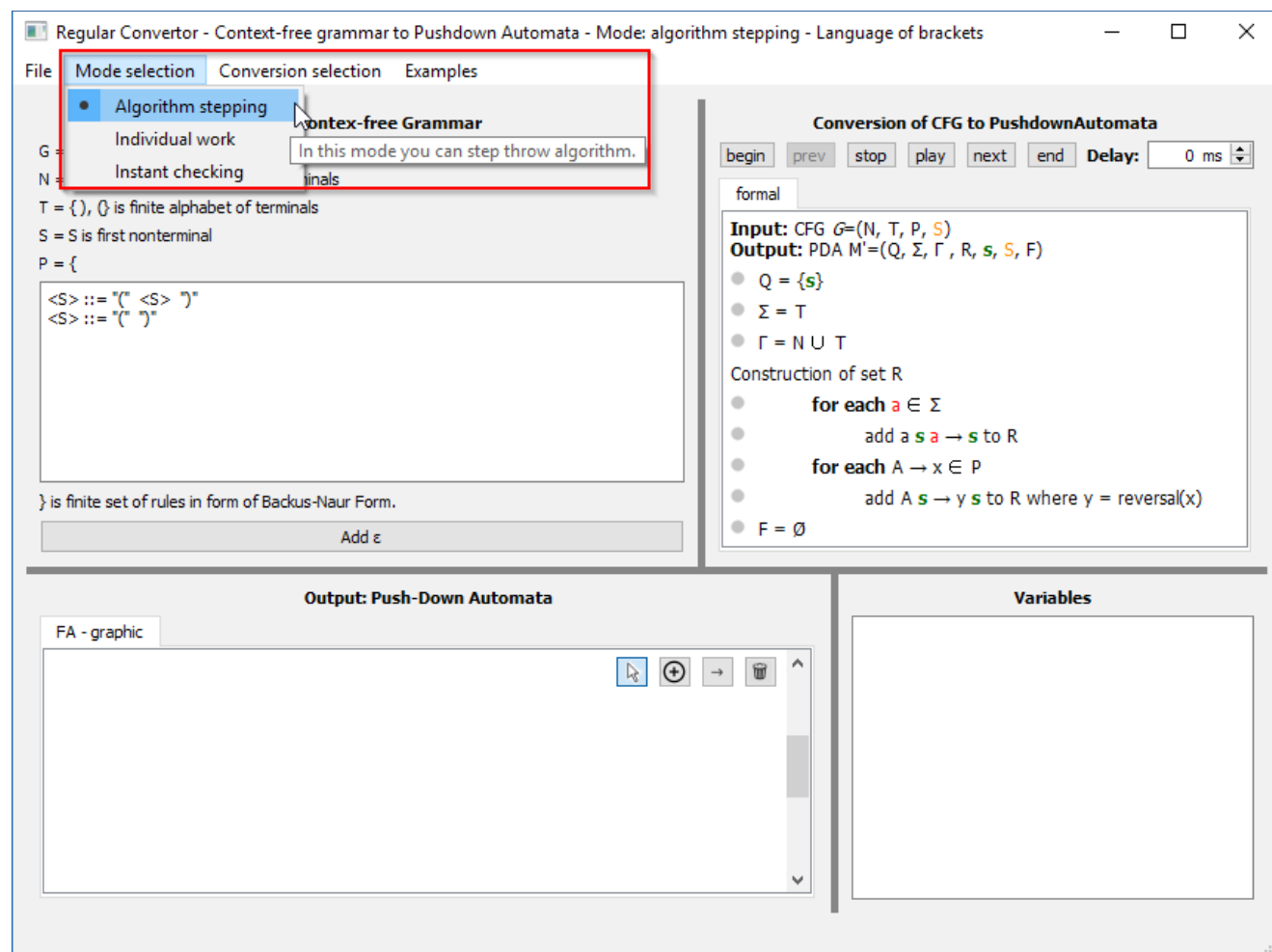
- Aktuálně je vybrán mód pro krokování algoritmu.
- Typ konverze, mód a název příkladu je uveden v záhlaví okna.



Módy aplikace

- Aplikace umožňuje nastavit 3 módy:

- 1) Krokování algoritmu. V tomto módu tvoří výstup algoritmu za uživatele aplikace.
- 2) Samostatná práce.
- 3) Mód průběžné kontroly.



Algoritmický widget v krokovacím módu

- Nahoře je jeho ovládání, které se skládá z tlačítek:
 - Přetoč algoritmus na začátek.
 - Posuň se o jeden krok vzad.
 - Zastav krokování.
 - Pust' krokování.
 - Posuň se o jeden krok vpřed.
 - Možnost nastavit zpoždění mezi kroky pro tlačítko spust' krokování.
- Aktuální krok je zvýrazněn žlutě.
- U každého kroku se dá zapnout/vypnout breakpoint na kterém, se zastavuje automatické přehrávání.

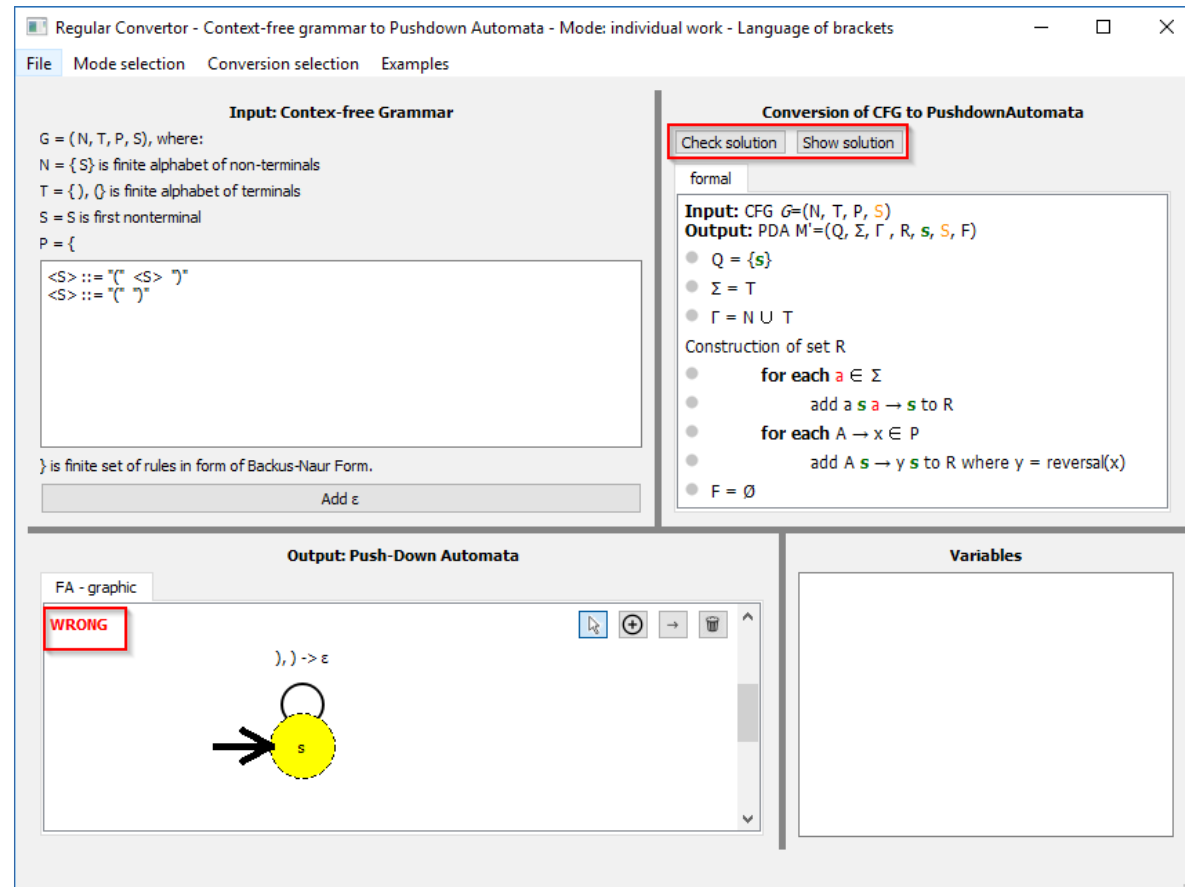
The screenshot displays the 'Regular Converter' application window, titled 'Regular Converter - Context-free grammar to Pushdown Automata - Mode: algorithm ...'. The interface is divided into several sections:

- Input: Context-free Grammar**: Contains the definition of the grammar $G = (N, T, P, S)$ with non-terminals $N = \{S\}$, terminals $T = \{(), \}$, and rules $P = \{ \langle S \rangle ::= "(\langle S \rangle)" \}$. An 'Add ϵ ' button is at the bottom.
- Conversion of CFG to PushdownAutomata**: A control panel with buttons 'begin', 'prev', 'stop', 'play', 'next', 'end', and a 'Delay' slider set to '0 ms'. Below these are radio buttons for 'formal' and 'graphical' views. The 'formal' view shows the conversion steps:
 - Input: CFG $G = (N, T, P, S)$
 - Output: PDA $M' = (Q, \Sigma, \Gamma, R, s, S, F)$
 - $Q = \{s\}$
 - $\Sigma = T$
 - $\Gamma = N \cup T$
 - Construction of set R:
 - for each $a \in \Sigma$: add $a s a \rightarrow s$ to R (highlighted in yellow)
 - for each $A \rightarrow x \in P$: add $A s \rightarrow y s$ to R where $y = \text{reversal}(x)$
 - $F = \emptyset$
- Output: Push-Down Automata**: A section with a 'FA - graphic' tab showing a state transition diagram with a single state 's' and a self-loop labeled $() ,) \rightarrow \epsilon$.
- Variables**: A text area showing the formal definition of the PDA components:

```
start state := s
Q = { s }
Σ = { (, ) }
Γ = { (, ), S }
a = )
R = {
    } s ) -> ε s
}
```


Algoritmický widget v módu samostatné práce

- Tento mód má 2 tlačítka:
 - *Zkontroluj řešení.*
 - *Zobraz správné řešení respektive vrať se zpět k uživatelskému řešení.*
- Při kontrole řešení nezáleží na pojmenování uzlu, nebo na pořadí přechodových pravidel.



Algoritmický widget v módu průběžné kontroly

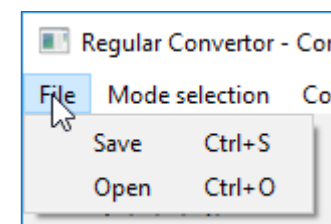
- V krátkém intervalu se automaticky kontroluje správnost řešení bez interakce s uživatelem.

The screenshot displays the 'Regular Converter' application window, titled 'Context-free grammar to Pushdown Automata - Mode: instant checking - Language of brackets'. The interface is divided into several sections:

- Input: Context-free Grammar**: This section defines the grammar $G = (N, T, P, S)$ where:
 - $N = \{S\}$ is the finite alphabet of non-terminals.
 - $T = \{(), \}$ is the finite alphabet of terminals.
 - S is the first nonterminal.
 - $P = \{$
 $\langle S \rangle ::= "(\langle S \rangle)"$
 $\langle S \rangle ::= "(\langle S \rangle)"$
 $\}$ is finite set of rules in form of Backus-Naur Form.
- Conversion of CFG to PushdownAutomata**: This section shows the formal conversion:
 - Input**: CFG $G = (N, T, P, S)$
 - Output**: PDA $M' = (Q, \Sigma, \Gamma, R, s, S, F)$
 - Parameters:
 - $Q = \{s\}$
 - $\Sigma = T$
 - $\Gamma = N \cup T$
 - Construction of set R**:
 - for each $a \in \Sigma$:
 - add $a \text{ s } a \rightarrow s$ to R
 - for each $A \rightarrow x \in P$:
 - add $A \text{ s } y \rightarrow s$ to R where $y = \text{reversal}(x)$
 - $F = \emptyset$
- Output: Push-Down Automata**: This section shows a graphical representation of the automaton with three states: 0, 1, and 2. State 0 is the start state, indicated by a double arrow. State 2 is the final state, indicated by a double circle. The word 'WRONG' is displayed in red text.
- Variables**: This section is currently empty.

Ukládání/načítání konverzí

- Program umožňuje ukládat a načítat konverze, ze souborů.
- Ukládá se typ konverze, mód a vstupní a výstupní model.
- Dá se využít ve výuce, tak že profesor může studentům předat zajímavou množinu příkladů, probíranou na přednáškách, nebo jako procvičení na doma.



Editace bezkontextové gramatiky

- Bezkontextová gramatika se zadává v Backus-Naur formě pomocí editačního boxu, který je předvyplněný příkladem gramatiky, pro snadnější orientaci uživatele.
- Pravidla, které mají stejnou levou stranu, se dají napsat na jeden řádek, kde jejich pravé strany se oddělují svislou čarou (znak „|“).
- Nevalidní vstup uživateli ukazuje změna pozadí na žlutou.

Input: Context-free Grammar

$G = (N, T, P, S)$, where:
 $N = \{\}$ is finite alphabet of non-terminals
 $T = \{\}$ is finite alphabet of terminals
 $S =$ is first nonterminal
 $P = \{\}$

Fill here Context-Free grammar in Backus-Naur Form.
E.g.
<StartNonTerminal> ::= <Nonterminal1> "terminal" | <Nonterminal2>
<Nonterminal1> ::= "terminal2" <Nonterminal2>
<Nonterminal2> ::= <StartNonTerminal>

} is finite set of rules in form of Backus-Naur Form.

Add ϵ

Input: Context-free Grammar

$G = (N, T, P, S)$, where:
 $N = \{\}$ is finite alphabet of non-terminals
 $T = \{\}$ is finite alphabet of terminals
 $S =$ is first nonterminal
 $P = \{\}$

TOTO JE NEVALIDNÍ VSTUP|

} is finite set of rules in form of Backus-Naur Form.

Add ϵ

Input: Context-free Grammar

$G = (N, T, P, S)$, where:
 $N = \{TOTO, VSTUP, VALIDNÍ\}$ is finite alphabet of non-terminals
 $T = \{JE\}$ is finite alphabet of terminals
 $S = TOTO$ is first nonterminal
 $P = \{\}$

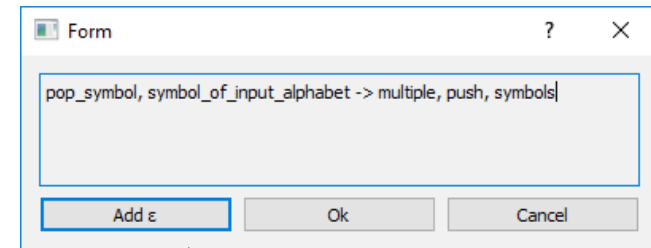
<TOTO> ::= "JE" <VALIDNÍ> | <VSTUP>

} is finite set of rules in form of Backus-Naur Form.

Add ϵ

Editace zásobníkového automatu

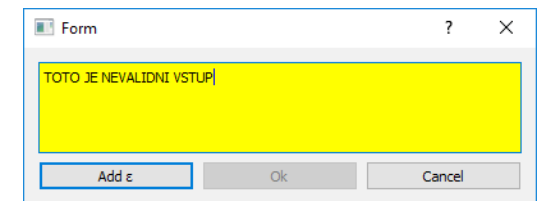
- Na obrázcích můžete vidět jakým způsobem se edituje zásobníkový automat.



Form

pop_symbol, symbol_of_input_alphabet -> multiple, push, symbols

Add ε Ok Cancel

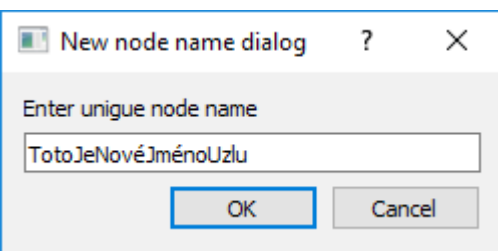


Form

TOTO JE NEVALIDNÍ VSTUP

Add ε Ok Cancel

Editace hrany dvojklikem



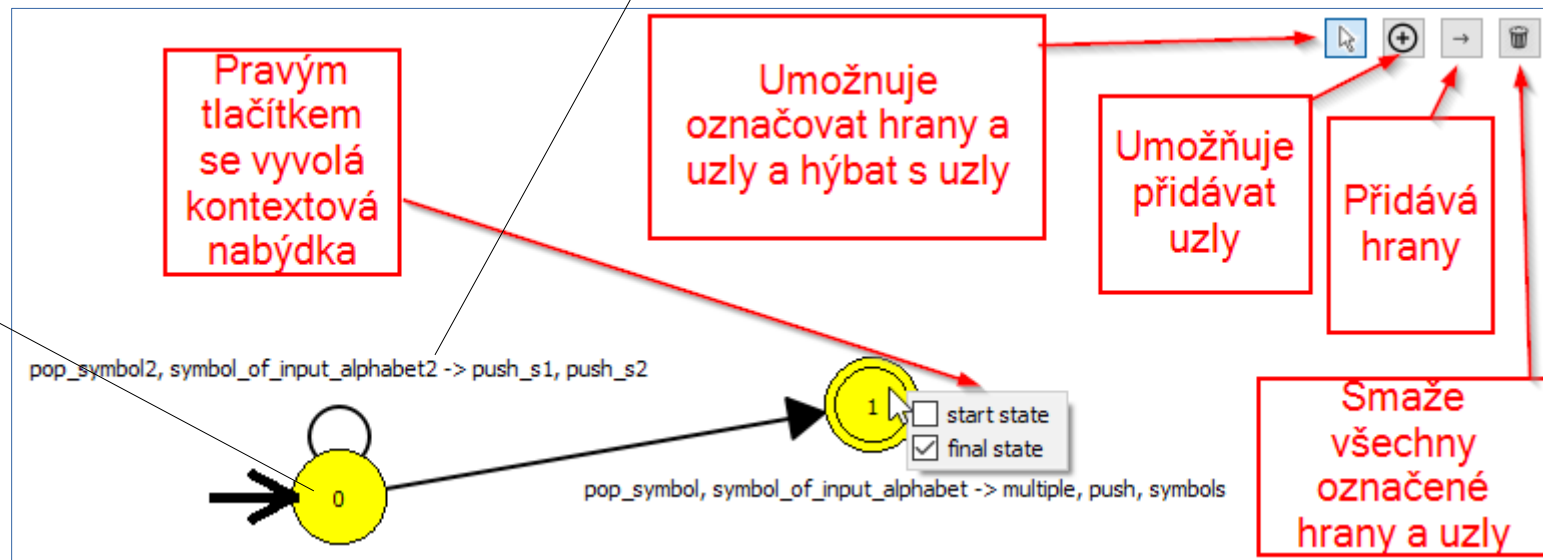
New node name dialog

Enter unique node name

TotoJeNovéJménoUzlu

OK Cancel

Editace jména uzlu dvojklikem



Děkuji za pozornost!

- Následuje přečtení posudků a otázky oponenta.

Otázky oponenta

- Proč jste se nezaměřil také na EBNF, která je spíše používaná při definici BKJ?
- Jelikož podobných pomůcek existuje již mnoho, prováděl jste také srovnání s již existujícími řešeními? Pokud ne, pokuste se o to alespoň krátce u obhajoby.

Srovnání s JFLAP

- JFLAP

- Podporuje velké množství konverzí a ostatních funkcionalit.
- Stejně jako RegularConvertor je Multiplatformí.
- Nutné instalovat Javu.
- Je free.
- Hůře pochopitelný, je nutné při prvním použití nahlízet do manuálu.
- Nemá zabudované příklady.
- Aplikace působí strohým dojmem.
- Obsahuje jenom jeden mód, který se vzdáleně podobá módu samostatné práce z RegularConvertoru.
- Neobsahuje vizualizaci algoritmu převodu.

