

Chipollino: как лабораторная работа превратилась в инструмент для исследований

Непейвода А.Н., Дельман А.Д., Князихин Д.П., Макаров Э.А.,
Терентьева А.С., Терюха М.Р.

МГТУ им. Н.Э. Баумана

29.06.24

Происхождение и мотивация проекта

В свободном доступе представлено малое количество инструментов, наглядно иллюстрирующих понятия теории автоматов.

Цель состоит в том, чтобы сделать теоретические концепции более осязаемыми и понятными для студентов, путем создания интерактивного конвертера формальных языков.

Существующие решения

Академические:

- (Wolfram Automata, Grail...)
- Демо-реализации к статьям



- Точность
- Разнообразие
- Связность
- Дружелюбность
- Наглядность

Демонстрационные:

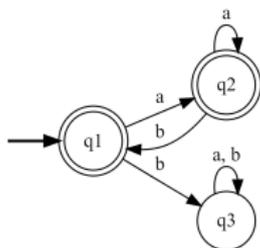
- парсинг, детерминизация, построение автомата



- Точность
- Разнообразие
- Связность
- Дружелюбность
- Наглядность

Существующие решения

Original Automaton Graph



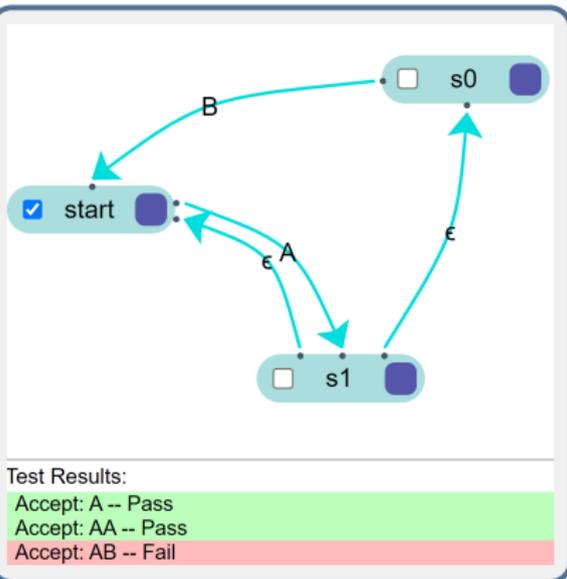
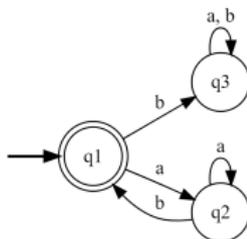
Computed Equivalences

0-Equivalence: $\{\{ "q1", "q2" \}, \{ "q3" \}\}$

1-Equivalence: $\{\{ "q3", "q1" \}, \{ "q2" \}\}$

2-Equivalence: $\{\{ "q3", "q1" \}, \{ "q2" \}\}$

Minimized Automaton Graph



Chipollino

Конвертер Chipollino — приложение, позволяющее генерировать, преобразовывать и анализировать различные представления формальных языков.

Список входных
инструкций

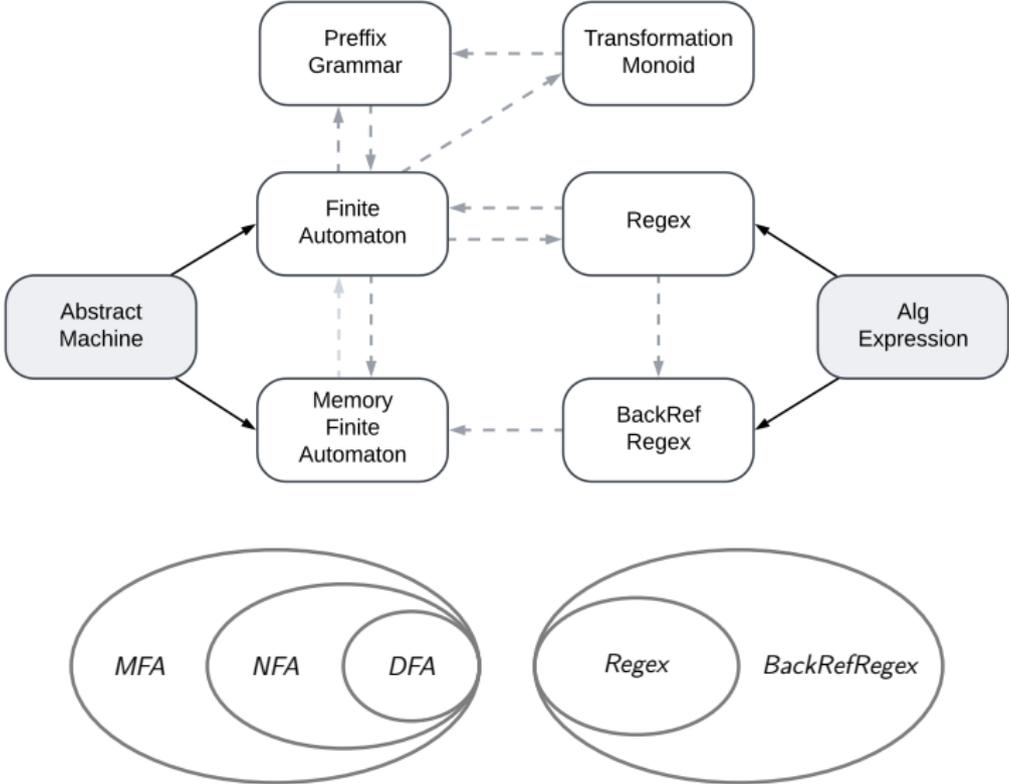


Подробный отчёт
о преобразованиях

Пример входных данных:

```
A = Complement.Annotate (Glushkov {a*})  
B = PrefixGrammar.Reverse A !!
```

Система типов

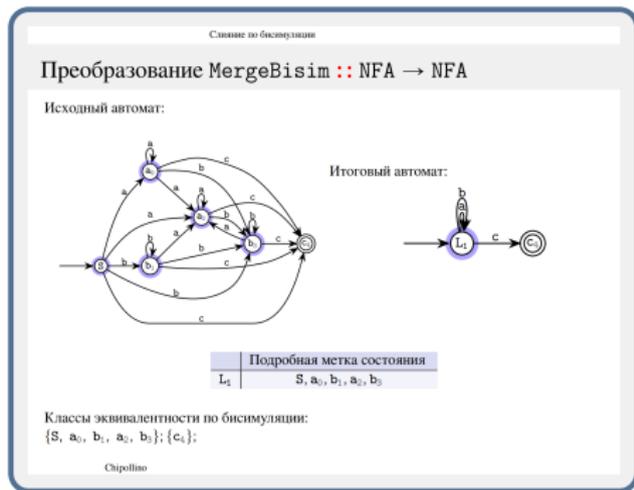


Шаблоны отчетов

```
\section{MergeBisim}
\begin{frame}{Пример слияния по бисимуляции}
  \textbf{Исходный автомат:}
  %template_oldautomaton

  \textbf{Итоговый автомат:}
  %template_result

  \textbf{Классы эквивалентности по
  бисимуляции:}
  %template_equivclasses
\end{frame}
```

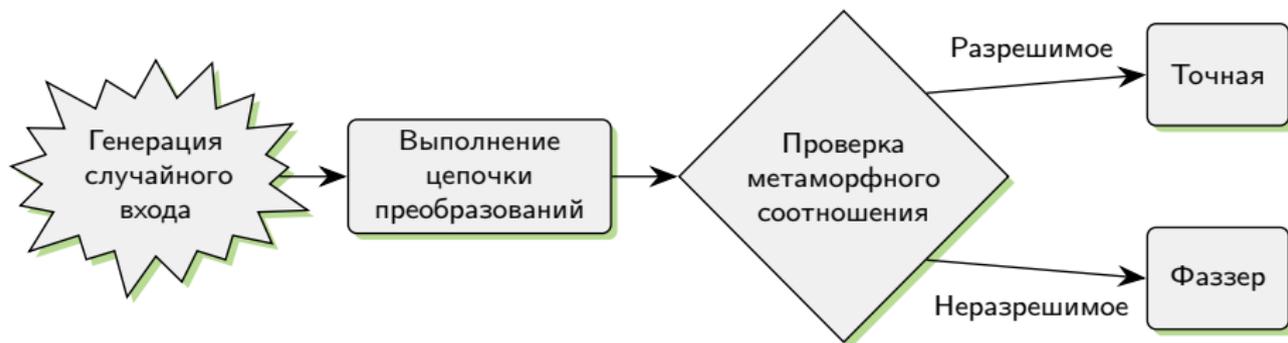


Автоматизированное расширение модулей

Чтобы обеспечить удобство масштабирования проекта реализованы инструменты, избавляющие разработчиков от шаблонного труда.

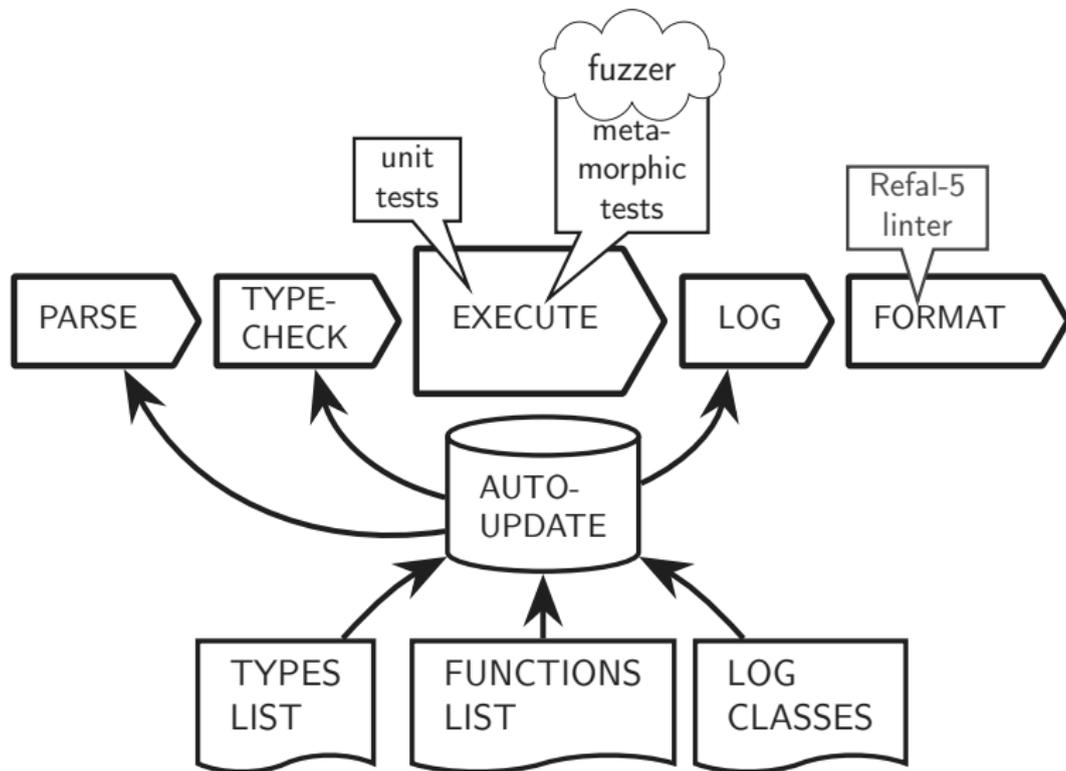


Метаморфное тестирование



- Автоматическая сборка и тестирование в пайплайнах;
- Экспериментальный проект — линтер для Refa15.

Архитектура проекта



Web-интерфейс

Input text:

```
N = RemEps.Thompson {ab|ac} !!  
Test {(aa)*|a*b} {a*} 1  
Verify (Equiv (Reverse.Reverse *NFA) *NFA)  
Verify (Equal (RemEps.Thompson *) (Antimirov *))  
Reverse.getNFA N1
```

Run

Function

Generate argument

Generate test

▼ Automata list

Reset Render

N1 Save Delete format: DSL

N2

A1

```
NFA  
0 initial_state label=S ;  
1 label=1 ;  
2 final label=2 ;  
...  
0 1 a ;  
1 2 b ;
```

+ Add Node Add Edge Change Finality

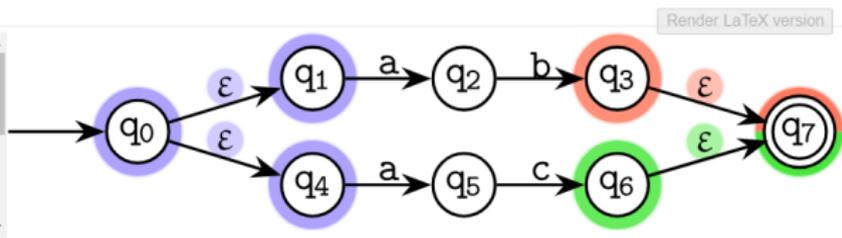
```
graph LR  
  Start(( )) --> S((S))  
  S -- a --> 1((1))  
  1 -- b --> 2(((2)))
```

Web-интерфейс

RemEps:

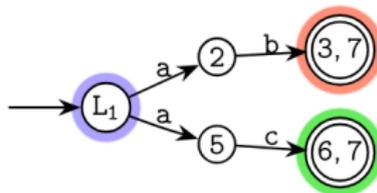
Автомат до преобразования:

```
Save to storage format: DSL
NFA
0 initial_state label=q0 ;
1 label=q1 ;
2 label=q2 ;
3 label=q3 ;
4 label=q4 ;
5 label=q5 ;
6 label=q6 ;
```



Автомат после преобразования:

```
Save to storage format: DSL
NFA
0 initial_state label=L1 ;
1 label=2 ;
2 label=5 ;
3 final label=6, 7 ;
4 final label=3, 7 ;
...
0 1 a :
```



```
format: CSV
;Подробная метка состояния
L1;0, 1, 4
```

	Подробная метка состояния
L1	0, 1, 4

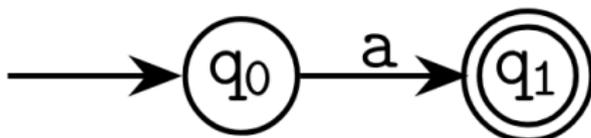
Мультиформатность

Thompson {a}

Save to storage format: DSL 

```
NFA
0 initial_state label=q0 ;
1 final label=q1 ;
...
0 1 a ;
```

[Render LaTeX version](#)



format:

- DSL
- DSL**
- DOT
- LaTeX
- GDF
- GML
- GEXF
- GraphML
- JSON



Save to storage format: DOT 

```
digraph {
  rankdir = LR
  node [shape=circle]
  dummy [label = "", shape = none]
  0 [label="q0"]
  dummy -> 0
  1 [label="q1", shape=doublecircle]
  0 -> 1 [label="a"]
}
```

Экспериментальный верификатор гипотез

Синтаксис:

Verify $\langle \text{Expr} \rangle$ n , где $\langle \text{Expr} \rangle$ – предикат, n – количество тестов.

Строится набор тестов, в котором на место \star в выражении подставляется случайный объект.

В качестве результата: доля успешных тестов и примеры, на которых гипотеза не выполнялась.

Генерируемые объекты верификатора:

- \star или $\star r$ – регулярное выражение
- $\star br$ – регулярное выражение с обратными ссылками
- $\star NFA$ – конечный автомат
- $\star DFA$ – детерминированный конечный автомат
- $\star MFA$ – автомат с памятью

Наши исследования

«*A Unified Construction of the Glushkov, Follow, and Antimirov Automata*»
C.Allauzen, M.Mohri: Math. Found. of Comp. Sci. 2006

Thompson → Glushkov

Verify (Equal (RemEps.Thompson ★) (Glushkov ★)) 5%

Thompson → Antimirov

Verify (Equal (RemEps.DeAnnotate.Minimize.RemEps.Annotate.Thompson ★)
(Antimirov ★)) 75%

«*Canonical regular expressions and minimal state graphs for definite events*»
J.A. Brzozowski: Mathematical Theory of Automata 1962

Минимизация Брззовски

Verify (Equal (Determinize.Reverse.Determinize.Reverse ★ NFA)
(Minimize ★ NFA)) 70%

Наши исследования

SYRCoSE23. *A. N. Nepeivoda, Yu. A. Belikova, K. K. Shevchenko, M. R. Teriukha, D. P. Knyazihin, A. D. Delman, A. S. Terentyeva.*
«**REDoS detection in “Domino” regular expressions by Ambiguity Analysis**»

SYRCoSE24. *A. N. Nepeivoda, A. D. Delman, A. S. Terentyeva.*
«**Bisimulations in Memory Finite Automata**»

Студенческая научная весна 2024, МГТУ им. Баумана.

А. Д. Дельман, А. С. Терентьева.

«**Метаморфное тестирование в изучении формальных языков**»

Планы на будущее

- Поддержка КС-языков в представлении PDA;
- Ускорение пайплайнов тестирования;
- Оптимизация алгоритмов.
- Создание API;
- Покрытие подробными теоретическими логиками большинства преобразований, дальнейшее совершенствование визуальной части отчетов.

Спасибо за внимание!



a_nevod@mail.ru
adelman2112@gmail.com
mathhyn@gmail.com