



Санкт-Петербургский государственный университет
Кафедра системного программирования

Разработка приложения для визуального редактирования и симуляции конечных автоматов

Александр Евгеньевич Пłosкин, 19.Б10-мм

Научный руководитель: к.т.н. Ю.В.Литвинов, доцент кафедры системного программирования

Санкт-Петербург
2021

- Использование визуального моделирования в образовательных целях
- Вероятное повышение среднего балла на курсах „Программирование“, „Теория автоматов и формальных языков“ и т.д.
- Удобство при промышленной разработке

Цель работы — создание минимального полезного продукта

Требования:

- Поддержка ДКА, НКА, Эпсилон-НКА
- Визуальное задание автомата
- Сцена с поддержкой увеличения/уменьшения
- Ввод входной строки, мгновенное исполнение и пошаговая симуляция
- Автоматическое тестирование
- Сохранение/загрузка вычислителя и тестов

- JFLAP

- ▶ Поддерживает всевозможные формализмы
- ▶ Пошаговая симуляция
- ▶ Поддержка сохранений
- ▶ Устарел

- automatonsimulator.com

- ▶ Веб-приложение
- ▶ Поддерживает ДКА, НКА, МП-автоматы
- ▶ Красивая пошаговая симуляция и автоматические тесты

- web.cs.ucdavis.edu

- ▶ Только текстовое задание
- ▶ Пошаговое исполнение в табличном виде

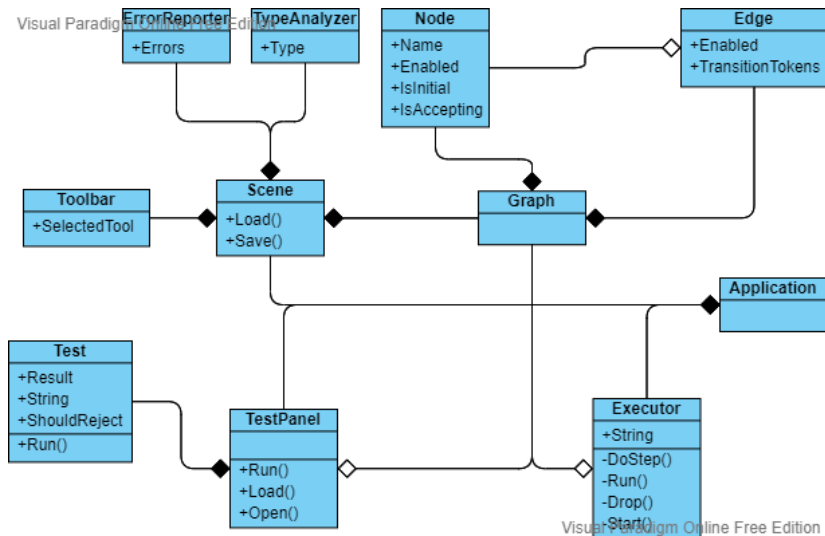
- .NET 5
- C#
- WPF
- Appveyor

Выбор библиотеки для визуализации графов

- MSAGL
 - ▶ Активно разрабатывается Microsoft
 - ▶ Позволяет манипулировать с объектами на сцене
 - ▶ Плохо задокументирована, есть проблемы при взаимодействии со сценой
- Graphviz
 - ▶ Красиво визуализирует графы
 - ▶ Не позволяет манипулировать с объектами
- Graph#
 - ▶ Использует Quickgraph, как логическое ядро
 - ▶ Поддерживает редактирование графа на сцене
 - ▶ Поддерживает различные алгоритмы визуализации

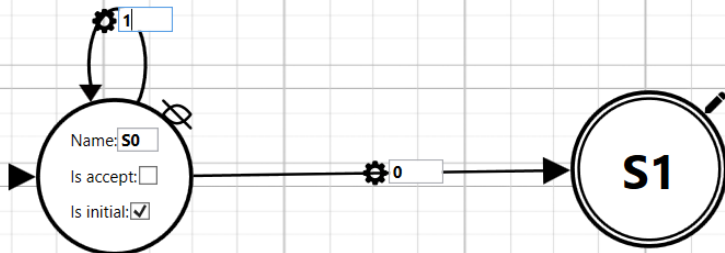
- Наследница Graph#
- Удобная работа со сценой
- Использовалась в REAL.NET

Архитектура



Реализация — визуальное редактирование

- Диагностика ошибок
- Определение типа



Реализация — симуляция исполнения

- Мгновенное исполнение
- Пошаговая симуляция с визуализацией на сцене
- Недетерминизм
- Блокировка при наличии ошибок
- Алгоритм исполнения

Реализация — автоматические тесты

Automata constructor (unsaved) *

File

Run all	2	N/A 0	2	Hide
String	Reject	Result	Run	Remove
1110	<input type="checkbox"/>			
1101	<input checked="" type="checkbox"/>			
1	<input type="checkbox"/>			
110	<input checked="" type="checkbox"/>			
Add new				

- Формат - XML
- Сохранение вычислителя
- Сохранение тестов

- Юнит-тесты
- Тестовые сценарии
- Апробация по методике System Usability Scale
- Оценка сложности операций пользователями по предложенным тестовым сценариям

Заключение

- Создан визуальный редактор вычислителей в графовом представлении
- Реализованы возможности симуляции работы вычислителя
- Разработана система автоматического тестирования
- Реализованы возможности сохранения и загрузки
- Разработан пользовательский интерфейс с локализацией
- Созданы тестовые сценарии
- Написана пользовательская документация

Ознакомиться с кодом можно в [репозитории](#) на неофициальной странице СП на GitHub'e