**Глава 1. Увод**

**1.1. Описание и идея на проекта**

Проектът е предназначен за създаване и управление на недетерминирани крайни автомати (NFA) с ℇ-преходи, над азбука, състояща се от цифрите и малките латински букви. Основната идея е да се предостави софтуер, който може да чете, сериализира, манипулира и визуализира такива автомати, както и да извършва различни операции върху тях, като проверка за празен език, крайност на езика, детерминизъм, разпознаване на думи и други.

**1.2. Цел и задачи на разработката**

Целта на разработката е да създаде програма, която да улесни работата с недетерминирани крайни автомати с ℇ-преходи и да предостави широк набор от функционалности за тяхното управление и анализ. Основните задачи на разработката включват:

Имплементация на структура за представяне на NFA с ℇ-преходи.

Реализация на механизъм за сериализация и десериализация на автоматите.

Въвеждане на уникални идентификатори за всеки прочетен автомат.

Изграждане на интерфейс за изпълнение на различни команди за управление на автоматите, включително:

list - списък с идентификаторите на всички прочетени автомати.

print <id> - извеждане на информация за всички преходи в автомата.

save <id> <filename> - запис на автомат във файл.

empty <id> - проверка дали езикът на автомата е празен.

finite <id> - проверка дали езикът на автомата е краен.

deterministic <id> - проверка дали автоматът е детерминиран.

recognize <id> <word> - проверка дали дадена дума е в езика на автомата.

union <id1> <id2> - намиране на обединението на два автомата и създаване на нов автомат.

concat <id1> <id2> - намиране на конкатенацията на два автомата и създаване на нов автомат.

un <id> - намиране на позитивната обвивка на автомат и създаване на нов автомат.

reg <regex> - създаване на нов автомат по даден регулярен израз.

draw <id> - визуализация на автомата с помощта на Graphviz или друго средство.

Създаване на документация и помощна информация за потребителите на програмата.

**1.3. Структура на документацията**

Документацията на проекта е организирана в няколко основни глави:

Глава 1. Увод - Представя описанието, целите и задачите на проекта, както и структурата на документацията.

Глава 2. Преглед на предметната област- Описва техническите аспекти на разработката, включително структурата на данните, алгоритмите и използваните технологии.

Глава 3. Проектиране - Подробно описание на проектиране на програмата, както и инструкции за използване на интерфейса и изпълнение на командите.

Глава 4. Тестове и валидация - Представя методите и резултатите от тестовете, извършени за проверка на коректността и ефективността на програмата.

Глава 5. Заключение - Обобщава постигнатото в проекта, изводите от разработката и възможностите за бъдещи подобрения и разширения.

**Глава 2.**

Технически Аспекти

Предметната област обхваща разработката на програма за манипулиране на недетерминирани краен автомати (NFA) с ℇ-преходи. Такава програма изисква имплементация на различни структури от данни и алгоритми, които позволяват създаването, модифицирането и анализа на автоматите.

Структура на Данните

Състояния (States): Всяко състояние на автомата е представено чрез обект от тип State, който съдържа информация за името на състоянието, дали е начално или финално, както и други характеристики.

Преходи (Transitions): Преходите между състоянията се представят чрез обекти от тип DeltaTransition, които съхраняват информация за началното и крайното състояние на прехода, както и символа, по който се осъществява този преход.

Недетерминиран Краен Автомат (NFA): Автоматът се съставя от идентификационен номер, множество състояния и множество преходи между тях.

Алгоритми

Проверка за Празен Език (Empty Language): Алгоритъмът за проверка дали езикът на автомата е празен използва търсене на пътища от началното състояние до финални състояния.

Проверка за Краен Език (Finite Language): За да се провери дали езикът на автомата е краен, може да се използва алгоритъм за обхождане на графа на автомата, който открива цикли.

Детерминизация (Determinization): Алгоритъмът за детерминизация на недетерминирани автомати преобразува недетерминирания автомат в еквивалентен детерминиран автомат.

Разпознаване на Думи (Word Recognition): Използва се алгоритъм за разпознаване на думи в автомат, който проследява преходите на автомата въз основа на входната дума.

Използвани Технологии

C++: Програмата е разработена на C++ заради бързата изпълнителна скорост и ефективното управление на паметта.

Graphviz: За визуализацията на автоматите може да се използва Graphviz или подобни библиотеки, които позволяват генериране на графични представяния на графи и автомати.

Структури от Данни: Използват се различни структури от данни като вектори, указатели и класове за представяне на състояния, преходи и недетерминирани крайни автомати.

Тази документация предоставя общ преглед на техническите аспекти на разработката на програмата за работа с недетерминирани краен автомат.

**Сложност на проекта:** В началото на създаването на проекта ми беше трудно, да си създам базовите класове. Но се консултирах с колега със същия проект и той ми показа неговия начин и така тръгна проекта. Също така понеже темата на проекта е свързана с предмет, който учим този семестър съм се допитвал до асистенти по ЕАИ и други колеги, изучаващи предмета. Като цяла проекта имаше доста голяма сложност според мен.

**Глава 3**

Инструкции за използване на клас Interface

Класът Interface предоставя команден интерфейс за работа с недетерминирани крайни автомати (NFA). Следвайки тези инструкции, можете да използвате функционалностите на програмата ефективно.

Инициализация и стартиране на интерфейса

Инициализация на интерфейса:

За да инициализирате командния интерфейс, използвайте метода Init. Този метод създава единствената инстанция на класа и връща препратка към нея.

Стартиране на интерфейса:

След инициализацията, стартирайте основния цикъл на командния интерфейс чрез метода Run. Този метод започва приемането и обработването на команди от потребителя.

Основни команди

help

Извежда списък с всички възможни команди и тяхното описание.

exit

Излиза от програмата.

open <filename>

Отваря файл с име <filename> и зарежда автоматите от него.

close

Затваря текущо отворения файл.

save

Запазва текущо отворения файл.

save <id> <filename>

Запазва автомата с идентификатор <id> във файл с име <filename>.

saveas <filename>

Запазва текущо отворения файл с ново име <filename>.

list

Извежда списък с идентификаторите на всички прочетени автомати.

print <id>

Извежда информация за всички преходи в автомата с идентификатор <id>.

empty <id>

Проверява дали езикът на автомата с идентификатор <id> е празен.

finite <id>

Проверява дали езикът на автомата с идентификатор <id> е краен.

deterministic <id>

Проверява дали автоматът с идентификатор <id> е детерминиран.

recognize <id> <word>

Проверява дали думата <word> е в езика на автомата с идентификатор <id>.

union <id1> <id2>

Намира обединението на автоматите с идентификатори <id1> и <id2>, създава нов автомат и отпечатва неговия идентификатор.

concat <id1> <id2>

Намира конкатенацията на автоматите с идентификатори <id1> и <id2>, създава нов автомат и отпечатва неговия идентификатор.

un <id>

Намира позитивната обвивка на автомата с идентификатор <id>, създава нов автомат и отпечатва неговия идентификатор.

reg <regex>

Създава нов автомат по даден регулярен израз <regex> и отпечатва неговия идентификатор.

**Глава 5**

Този проект предоставя мощен инструмент за работа с недетерминирани крайни автомати чрез команден интерфейс. Той включва разнообразни функции за управление на файлове и автомати, както и различни операции върху автоматите, което го прави подходящ за изследвания и обучение в областта на формалните езици и автомати.