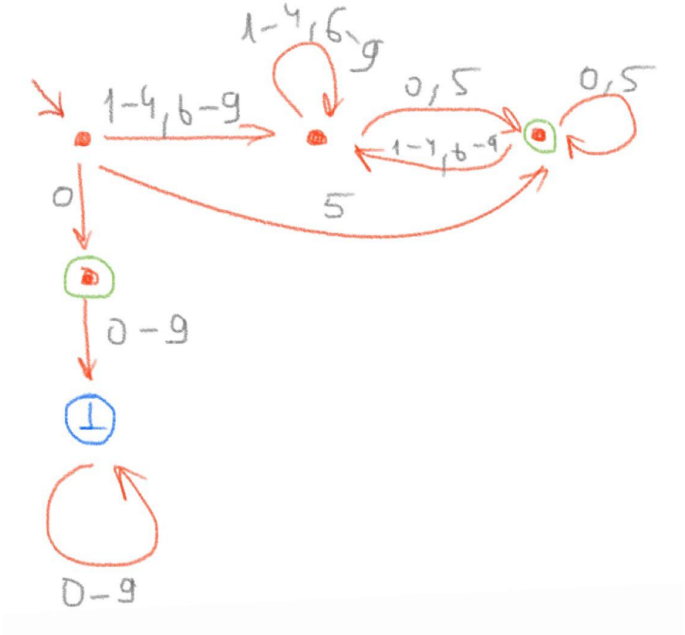
Михайлова Анна

Домашнее задание по ФЯ на 13.09.2021

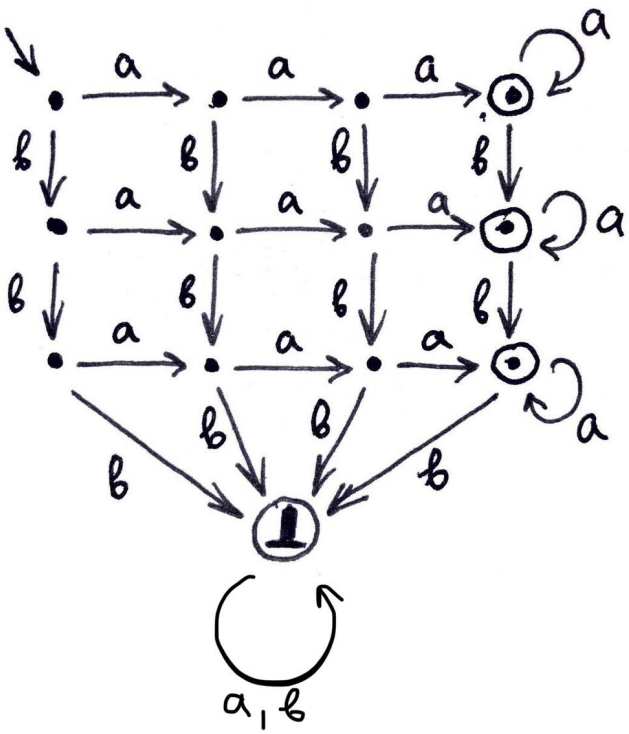
1. Построить полный конечный детерминированный автомат, распознающий язык неотрицательных чисел без лидирующих нулей, делящихся на 5.

Решение:



2.Построить полный конечный детерминированный автомат, распознающий язык строк над алфавитом $\{a, b\}$, в которых букв a не меньше 3, а букв b не больше 2.

Решение:



3. Найти документацию вашего второго самого любимого языка программирования. Найти особенности лексического синтаксиса, о которых вы раньше не знали. В отчете описать особенности и привести ссылку на спецификацию.

Решение:

Новости про C++: `[](){}();` – неожиданная штука, которая имеет смысл – это корректный код на C++, который скомпилируется. Здесь: объявили лямбда-функцию, которая ничего не захватывает, у нее нет параметров, у нее пустое тело, а потом вызвали эту лямбду.

ссылка

Еще для меня стало новостью, что в `Python3` можно добавлять `else` после `while` или после `for`, который выполнится, если выход из цикла случился естественным путем, а не с помощью `break`.

ссылка

4. Придумать язык для описания конечных автоматов. Привести описание этого языка на естественном языке. Привести три файла, описывающих какие-нибудь автоматы на вашем языке.

Решение:

Что нужно уметь описывать?

- алфавит
- множество состояний
- начальное состояние
- множество терминальных состояний
- функция переходов

Итак, пусть описание автомата будет содержать:

1. строка, в которой описан алфавит перечислением символов через пробел
2. строка, в которое задан список состояний – 0S для стартовой, число + 'M' для промежуточного состояния, число + 'T' для терминального
3. функция переходов задается так: состояние1 дефис состояние2 в квадратных скобках через пробел символы заданного алфавита, по которым происходит переход из состояния1 в состояние2

Пример 1:

Конечный автомат, получающий на вход число в десятичной системе счисления и определяющий, четное ли число получено на вход:

```
0 1 2 3 4 5 6 7 8 9
0S 1T 1M 2M 2T
0S - 1T [0]
0S - 2T [2 4 6 8]
0S - 2M [1 3 5 7 9]
```

```
1T - 1M [0 1 2 3 4 5 6 7 8 9]
1M - 1M [0 1 2 3 4 5 6 7 8 9]
2M - 2M [1 3 5 7 9]
2M - 2T [0 2 4 6 8]
2T - 2M [1 3 5 7 9]
```

Пример 2:

Конечный автомат, получающий на вход строку над алфавитом $\{a, b\}$ и определяющий, правда ли, что она содержит буквы a и содержит четное количество букв a.

```
a b
0S 1M 2M 1T
0S - 1M [b]
0S - 2M [a]
1M - 1M [b]
1M - 2M [a]
2M - 2M [b]
2M - 1T [a]
1T - 1T [b]
1T - 2M [a]
```

Пример 3:

Конечный автомат, получающий на вход строку над алфавитом $\{0, 1\}$ и определяющий, правда ли, что она содержит нечетное количество 0 и нечетное количество 1.

```
0 1
0S 1M 1T 2M
0S - 1M [0]
1M - 0S [0]
1M - 1T [1]
1T - 1M [1]
1T - 2M [0]
2M - 1T [0]
2M - 0S [1]
0S - 2M [1]
```

5. Подсветить какие-нибудь особенности лексического синтаксиса в среде разработки. В отчете указать, что вы хотели подсветить и как.

Решение:

Создала файл `.sublime-syntax` для редактора `Sublime`.

Хотела фиолетовым цветом подсвечивать символы (в первой строке), оранжевым цветом начальное и терминальные состояния, зеленым промежуточные состояния. Набор символов, по которым переходим в другое состояние, подсвечивается серым. Дефис белый.

Пример:

```
0 1
0S 1M 1T 2M
0S - 1M [0]
1M - 0S [0]
1M - 1T [1]
1T - 1M [1]
1T - 2M [0]
2M - 1T [0]
2M - 0S [1]
0S - 2M [1]
```

инструкция как пользоваться

Код отправила отдельным коммитом. Расширение файлов `.dfa`.