

Отчет по лабораторной работе №5

Диаграмма PDA (Вариант 3)

Баев Д.А. Федоров А.Н.

31 декабря 2022 г.

1 Задание

- Предложить грамматику PDA.
- Прочитать описание PDA в заданном синтаксисе
- Построить диаграмму PDA, где будут выделены состояния-ловушки, недетерминированные переходы и переходы, не влияющие на стек

2 Установка и запуск

1. Установка осуществляется путем клонируем репозиторий проекта

```
git clone https://github.com/Pur3Vessel/PDA_diagram
```

2. Переходим в папку проекта

```
cd PDA-diagram
```

3. Запускаем программу из корневой(!важно!) папки проекта, с указанием пути до файла с тестом и конфигом (опционально)

```
lua src/main.lua path/to/test.txt path/to/config.txt
```

4. Построенная диаграмма автоматически выведется на экран и сохранится в файле test.pdf в корне проекта

```
test.pdf
```

Интерпретация диаграммы:

- Зеленый круг (овал) - стартовое состояние
- Двойной круг - завершающее состояние
- Красный квадрат - ловушка
- Тонкая стрелка - недетерминированный переход
- Зеленая стрелка - стэконеависимый переход

3 Грамматика

Стартовый нетерминал: `[automata]`

`[automata]` \rightarrow `[initail - state][states][sep][transitions]` | `[initial - state][sep][transitions]`
`[initail - state]` \rightarrow `[state - name]` | `[state - name] - [flag]`
`[states]` \rightarrow `[state]`, `[states]` | `[state]`
`[state]` \rightarrow `[state - name]` | `[state - name] - [flag]`
`[state - name]` \rightarrow `[symbol]` | `[symbol][state - name]`
`[transitions]` \rightarrow `[transition]` | `[transition][sep][transitions]`
`[transition]` \rightarrow `[before][trans - sep][after]` | `[state - name]`, `[alphabet - symbol]`, `[stack - any][trans - sep][state - name]`
`[transition]` \rightarrow `[state - name]`, `[alphabet - symbol]`, `[stack - bottom][trans - sep][state - name]`, `[stack - symbols][stack - any]`
`[before]` \rightarrow `[state - name]`, `[alphabet - symbol]`, `[stack - symbol]`
`[after]` \rightarrow `[state - name]`, `[stack - symbols]` | `[state - name]`, `[empty]`
`[alphabet - symbol]` \rightarrow `[al - sym]` | `[empty]`
`[stack - symbols]` \rightarrow `[stack - symbol][stack - symbols]` | `[stack - symbol]`

`[symbol]` - символ в имени состояния. Может быть любым разрешенным символом.

`al-sym` - символ входного алфавита. Задается в `config` некоторой регуляркой (по умолчанию: `[a-z]`)

`stack-symbol` - символ стэкового алфавита. Задается в `config` некоторой регуляркой (по умолчанию: `[A-Y]`)

`flag` - строка, которая задает метку завершающего состояния. По умолчанию: `final`.

`sep` - символ, который задает разделитель между "строками". По умолчанию: `;`

`trans-sep` - строка, которая задает разделитель между двумя частями правила. По умолчанию: `->`

`empty` - символ, который задает пустую строку. По умолчанию:

`stack-any` - символ, который задает любой стэковый символ. По умолчанию: `*`

`stack-bottom` - символ, который задает дно стэка. Не должен распознаваться регуляркой, задающей стэковый алфавит. По умолчанию: `Z`

Ограничения:

- Имена состояний, которые записываются в описании переходов, должны быть объявлены в `[states]`
- В `[symbol]` запрещено использовать символы токенов, а также символы `,` `-` и пробельные символы

4 Заполнение `config`

Параметры указываются через запятую в формате: `[имя параметра] значение`

Пример: `[trans-sep] ::=, [empty] !`

В случае, если имя параметра указано неверно, этот параметр не будет изменен.

Ограничения:

- Нельзя параметризовать токены запятой и тире
- Для избежания неоднозначности все токены должны быть уникальными значениями
- Нельзя параметризовать токены пробельными символами
- Стэковый алфавит, входящий алфавит и дно стэка должны параметризоваться символами ASCII