

Лабораторная работа №5

Белецкий
Панов

Декабрь 2022

1 Описание

Преобразование синтаксиса, записанного в РБНФ в КС-грамматику

2 Запуск программы

1. Необходимо клонировать репозиторий проекта при помощи команды

```
git clone
```

2. В папке src находится файл main.kt с main функцией программы. Через файл main.kt происходит запуск программы.

3 Настройки файла syntax.txt

Перед вводом данных необходимо внести изменения в файл 'syntax.txt', лежащий внутри папки проекта. Внутри файла можно задавать конфигурацию следующих параметров:

- NStart - символ начала нетерминала. Значение по умолчанию: "_"
- NEnd - символ конца нетерминала. Значение по умолчанию: "_"
- 'Arrow' - строка-разделитель для имени нетерминала и его значения. Значение по умолчанию: "::="
- 'Epsilon' - символ пустого символа. Значение по умолчанию: "\$"
- IterStart - символ начала итерации. Значение по умолчанию: "{"
- IterEnd - символ конца итерации. Значение по умолчанию: "}"
- OptionalEnterStart - символ начала условного вхождения. Значение по умолчанию: "[" (Открывающая квадратная скобка)
- OptionalEnterEnd - символ конца условного вхождения. Значение по умолчанию: "]" (закрывающая квадратная скобка)
- NecessarilyStart - символ начала группировки элементов для обязательного вхождения. Значение по умолчанию: "("
- NecessarilyEnd - символ конца группировки элементов для обязательного вхождения. Значение по умолчанию: ")"
- Alternative - символ-разделитель правил или "альтернатива". Значение по умолчанию: "||"

Правила задания параметров:

1. Каждый новый параметр начинается с новой строчки. Значение задается через символ '='. Пример задания параметра:

```
Arrow = ::=
```

2. Если в файле указать несуществующий параметр, то он проигнорируется. Пример несуществующего параметра:

```
MY_CUSTOM_PARAM = @
```

3. Если неправильно записать параметр, в выводе появится соответствующая строка, для параметра будет использоваться значение по умолчанию. Пример сообщения об ошибке:

```
Error while parsing string: "Arrow_!=_—>"
```

4. Если не указать какой-либо существующий параметр, выберется значение по умолчанию.
5. Имена различных параметров не должны совпадать, за исключением тех, что представляют собой, например, открывающий и закрывающий символы. Так параметры NStart и NEnd могут совпадать, а NStart и NecessarilyStart нет.

4 Тесты

Вместе с файлами программы в проекте лежат файлы тестов в папках testN. Внутри есть необходимые параметры для файла syntax.txt, вводимые строки, соответствующие параметрам из файла syntax.txt, а также ожидаемый вывод программы при заданных входных данных и параметрах. Использование теста на примере теста из папки test2:

1. Скопировать содержимое файла test2/syntax.txt в src/syntax.txt

```
DEFAULT_SETTINGS = true
```

2. В строку ввода ввести содержимое файла input.txt

```
_Number_ ::= [ "+" | "-" ] _NaturalNumber_ [ "." [ _NaturalNumber_ ] ] [ ( "e" | "E" ) [ "+" | "-" ] _N
_NaturalNumber_ ::= _Digit_ { _Digit_ }
_Digit_ ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

3. Так как ввод строк завершается после ввода пустой строки, то вводим ее и получаем на выходе преобразование

```
[Nonterm0] -> + | - | e
[Nonterm1] -> [NaturalNumber] | e
[Nonterm2] -> . [Nonterm1] | e
[Nonterm3] -> e | E
[Nonterm4] -> [Nonterm3] [Nonterm0] [NaturalNumber] | e
[Number] -> [Nonterm0] [NaturalNumber] [Nonterm2] [Nonterm4]
[Nonterm5] -> [Digit] [Nonterm5] | e
[NaturalNumber] -> [Digit] [Nonterm5]
[Digit] -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Вместе с выводом дополнительно выводится отладочная информация, однако результат выполнения программы находится в самом конце вывода после строки "КС вид:"