

Документация по использованию конвертера EBNF-to-CFG

Емельяненко Д.С.
Яровикова А.С.

Декабрь 2022

1 Введение

Конвертер [EBNF-to-CFG](#) используется для построения [КС-грамматики](#), эквивалентной введенной [РБНФ](#) в пользовательском синтаксисе.

2 Использование

2.1 Установка и запуск

1. Клонировать репозиторий проекта в выбранную папку

```
git clone https://github.com/ynastt/EBNF-to-CFG.git
```

2. Переходим в папку проекта

```
cd EBNF-to-CFG
```

3. В папке tests создаем папку для теста и создаем файлы syntax.txt (опционально), input.txt, CFGsyntax.txt (опционально)

```
cd tests
mkdir test1
cd test1
vim syntax.txt
vim input.txt
vim CFGsyntax.txt
```

4. Запускаем программу main.cpp из корневой папки проекта

```
g++ -Wall -o main main.cpp
./main
```

5. Результат программы будет в терминале и созданном файле output.txt

```
cd EBNF-to-CFG\tests\test1
vim output.txt
```

2.2 Формат входных данных

Программа обрабатывает входные данные следующих файлов:

- syntax.txt - файл с пользовательскими параметрами РБНФ грамматики;
- input.txt - файл с пользовательской грамматикой;

- CFGsyntax.txt - файл с пользовательскими параметрами для КС-грамматики

Входная РБНФ грамматика имеет вид:

```

[Grammar] → [Rules]
[Rules] → [Rule] | [Rule] [Delim] [Rules]
[Rule] → [Nterm] [Arrow] [RightPart]
[RightPart] → [Operation[]Start] [Term] [ConcOrAlt] [Rightpart] [Operation[]End] [ConcOrAlt] [Rightpart] |
               [Operation[]Start] [Nterm] [ConcOrAlt] [Rightpart] [Operation[]End] [ConcOrAlt] [Rightpart] |
               [Operation[]Start] [Rightpart] [Operation[]End] [ConcOrAlt] [Rightpart] |
               [Operation()Start] [Term] [ConcOrAlt] [Rightpart] [Operation()End] [ConcOrAlt] [Rightpart] |
               [Operation()Start] [Nterm] [ConcOrAlt] [Rightpart] [Operation()End] [ConcOrAlt] [Rightpart] |
               [Operation()Start] [Rightpart] [Operation()End] [ConcOrAlt] [Rightpart] |
               [Operation{}Start] [Term] [ConcOrAlt] [Rightpart] [Operation{}End] [ConcOrAlt] [Rightpart] |
               [Operation{}Start] [Nterm] [ConcOrAlt] [Rightpart] [Operation{}End] [ConcOrAlt] [Rightpart] |
               [Operation{}Start] [Rightpart] [Operation{}End] [ConcOrAlt] [Rightpart] |
               [Term] [ConcOrAlt] [Rightpart] |
               [Nterm] [ConcOrAlt] [Rightpart] |
               [Empty]
[ConcOrAlt] → [Concat] | [Alternative]
[Term] → [TermStart] [TermStr] [TermEnd]
[Nterm] → [NtermStart] [NtermStr] [NtermEnd]
[Alternative] → ' | '
[Concat] →
[NtermStr] → [A - Z]
[TermStr] → [a - z]
[Operation[]Start] → ' [ '
[Operation[]End] → ' ] '
[Operation{}Start] → ' { '
[Operation{}End] → ' } '
[Operation()Start] → ' ( '
[Operation()End] → ' ) '

```

Все параметры, которые встречаются в грамматике, но не имеют правил раскрытия, пользователь может задать самостоятельно в файле syntax.txt. По умолчанию программа задаёт эти параметры, опираясь на встроенный синтаксис. Ниже перечислены данные параметры. Пустые ячейки в таблице означают то, что значение параметра = пустая строка.

Название параметра	Применение	Значение по умолчанию
[Delim]	разделитель правил грамматики	'\n'
[Arrow]	разделитель левой и правой части правила	'->'
[TermStart]	символ начала терминала	
[TermEnd]	символ конца терминала	
[NtermStart]	символ начала нетерминала	,
[NtermEnd]	символ конца нетерминала	,
[Empty]	символ пустоты (обозначение для)	'#'

Таблица 1: Параметры пользовательской РБНФ грамматики.

Пользовательская РБНФ грамматика имеет следующие ограничения (при несоблюдении программа аварийно завершится):

- Пользователь должен вводить грамматику в input.txt, учитывая значения по умолчанию для тех параметров, значения которых он оставил пустыми в файле syntax.txt.

(Например, в РБНФ грамматике по умолчанию в качестве обрамляющих символов для нетерминалов используются одинарные кавычки 'S', если пользователь не задаст эти параметры или их оставит пустым (NtermStart= и NtermEnd=), то программа подставит значения по умолчанию, т.е. NtermStart= ' и NtermEnd= '. Если пользователь далее введет грамматику без учета значений параметров по умолчанию, то программа выдаст ошибку и завершится.)

- Пробел запрещен для обозначения каких-либо параметров пользовательской грамматики.
- Начальным считается нетерминал, правило переписывания которого введено первым в input.txt.

2.3 Формат выходных данных

В результате выполнения программы в директории tests/test<номер теста> будет создан файл output.txt с построенной КС-грамматикой. Также результат программы будет выведен в терминале.

КС-грамматика на выходе имеет следующий вид:

```
[Grammar] → [Rules]
[Rules] → [Rule] | [Rule] [Delim] [Rules]
[Rule] → [Nterm] [Arrow] [RightPart]
[RightPart] → [RightPart1] | [Empty]
[RightPart1] → [Term] | [Nterm] | [Nterm] [ConcOrAlt] [RightPart1] | [Term] [ConcOrAlt] [RightPart1]
[ConcOrAlt] → [Concat] | [Alternative]
[Term] → [TermStart] [TermStr] [TermEnd]
[Nterm] → [NtermStart] [NtermStr] [NtermEnd]
[NtermStr] → [A - Z]
[TermStr] → [a - z]
```

Все параметры, которые встречаются в грамматике, но не имеют правил раскрытия, пользователь может задать самостоятельно в файле CFGsyntax.txt. По умолчанию программа задаёт эти параметры, опираясь на встроенный синтаксис. Ниже перечислены данные параметры. Пустые ячейки в таблице означают то, что значение параметра = пустая строка.

Название параметра	Применение	Значение по умолчанию
[Delim]	разделитель правил грамматики	'\n'
[Arrow]	разделитель левой и правой части правила	'->'
[Empty]	символ пустоты (обозначение для)	
[TermStart]	символ начала терминала	
[TermEnd]	символ конца терминала	
[NtermStart]	символ начала нетерминала	'
[NtermEnd]	символ конца нетерминала	'
[Concat]	символ конкатенации	
[Alternative]	символ для альтернативы	' '

Таблица 2: Параметры КС-грамматики.

2.4 Пример использования

syntax.txt

```
Delim=
Arrow= :=
NtermStart= SS
```

NtermEnd= SS

input.txt

```
SSKSS := [a {SSSS}] | b | #  
SSSS := c | #
```

CFGsyntax.txt

```
Arrow= =>  
Empty= @  
Alternative= \\\
```

output.txt (и в терминале)

```
'K'=>'A0'  
'A0'=>'A2'\\ 'A1'  
'A1'=>b\\@  
'A2'=>a 'A3'\\@  
'A3'=>'S' 'A3'\\@  
'S'=>'A4'  
'A4'=>c\\@
```