МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Домашняя работа №3**

Вариант 10

Выполнил:

Круглов Егор Ильич, P3324

Преподаватель:

Лаздин Артур Вячеславович

Санкт-Петербург

2025

**Выполнение**

1. S → abAC | acAB | aa

2. B → BBb | Bbb | b

3. A → aAA | aCC | bB | b

4. C → cc | cC

Убираем левую рекурсию для 2 правила (для B):

Сделаем левую факторизацию для S:

Сделаем левую факторизацию для A:

Сделаем левую факторизацию для C:

По итогу получаем:

1.

2.

3.

4.

5.

6.

7.

8.

9.

|  |  |  |  |
| --- | --- | --- | --- |
| Символ | nullable | FIRST | FOLLOW |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | true |  |  |
|  | false |  |  |
|  | false |  |  |
|  | true |  |  |
|  | false |  |  |
|  | false |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Видим, что у нас есть 3 конфликта, недающих грамматике быть LL(1). Изменим грамматику, чтобы их избежать:

1.

2.

3.

4.

5.

6.

7.

8.

9.

|  |  |  |  |
| --- | --- | --- | --- |
| Символ | nullable | FIRST | FOLLOW |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Мы убрали эпсилон правила, а также заменили:

1. на
2. на

Также заметим, что у нас образовалось . Уберем , просто заменив его везде на .

1.

2.

3.

4.

5.

6.

7.

8.

|  |  |  |  |
| --- | --- | --- | --- |
| Символ | nullable | FIRST | FOLLOW |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |
|  | false |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Теперь у нас действительно таблица LL(1) анализатора.

**Код:**

Ссылка на github: <https://github.com/KruglovEgor/Compilers/blob/main/lab3/main.py>

def ll1\_parser(input\_string):  
 parse\_table = {  
 'S': {'a': ['a', 'S\_1']},  
 'S\_1': {'a': ['a'], 'b': ['b', 'A', 'C'], 'c': ['c', 'A', 'B']},  
 'B': {'b': ['b', 'B\_1']},  
 'B\_1': {'a': ['a', 'b', 'B\_1'], 'b': ['B', 'b', 'B\_1'], 'c': ['c']},  
 'A': {'a': ['a', 'A\_1'], 'b': ['b', 'B']},  
 'A\_1': {'a': ['A', 'A'], 'b': ['A', 'A'], 'c': ['C', 'C']},  
 'C': {'c': ['c', 'C\_1']},  
 'C\_1': {'a': ['a'], 'c': ['C']}  
 }  
  
 stack = ['$', 'S']  
 input\_string = list(input\_string) + ['$']  
 pointer = 0  
  
 print(f"Parsing '{''.join(input\_string[:-1])}':")  
  
 while stack:  
 top = stack[-1]  
 current\_input = input\_string[pointer]  
  
 print(f"Stack: {stack}, Input: {input\_string[pointer:]}")  
  
 # Если верхний символ — терминал или конец строки  
 if top in {'a', 'b', 'c', '$'}:  
 if top == current\_input:  
 if top == '$':  
 print("Success! Input accepted.")  
 return True  
 stack.pop()  
 pointer += 1  
 else:  
 # Ошибка: промах по терминалу  
 print(f"Error: Expected '{top}', but found '{current\_input}' at position {pointer}")  
 return False  
  
 # Если верхний символ — нетерминал  
 elif top in parse\_table:  
 if current\_input in parse\_table[top]:  
 stack.pop()  
 production = parse\_table[top][current\_input]  
 # Добавляем правило в стек в обратном порядке  
 for symbol in reversed(production):  
 stack.append(symbol)  
 else:  
 # Ошибка: промах по нетерминалу  
 expected = list(parse\_table[top].keys())  
 print(  
 f"Error: No rule for '{top}' with input '{current\_input}' at position {pointer}. Expected one of {expected}")  
 return False  
 else:  
 print(f"Error: Unknown symbol '{top}' in stack")  
 return False  
  
 print("Error: Stack not empty after input exhaustion")  
 return False  
  
  
# Тестирование  
positive\_cases = [  
 "aa",  
 "abacaccaca",  
 "acbbcbabc"  
]  
  
negative\_cases = [  
 "abcc",  
 "acaab",  
 "abc"  
]  
  
print("Testing positive cases:")  
for test in positive\_cases:  
 ll1\_parser(test)  
 print("---------------------------------")  
  
print("\nTesting negative cases:")  
for test in negative\_cases:  
 ll1\_parser(test)  
 print("---------------------------------")

Вывод:





