

Федеральное государственное автономное образовательное учреждение высшего
образования "Национальный Исследовательский Университет ИТМО"
Мегафакультет Компьютерных Технологий и Управления
Факультет Программной Инженерии и Компьютерной Техники



Вариант №5
Домашняя работа 3
по дисциплине
'Разработка компиляторов'

Выполнил Студент группы Р33102
Лапин Алексей Александрович
Преподаватель:
Лаздин Артур Вячеславович

г. Санкт-Петербург
2024г.

Задание

Задание к лабораторной (домашней) работе «Конструирование LL(1) анализатора для КС-грамматики.» Для грамматики из соответствующего варианта необходимо:

1. Устранить левую рекурсию.
2. Провести левую факторизацию грамматики.
3. Для полученной преобразованной грамматики построить множества FIRST и FOLLOW для нетерминальных символов грамматики.
4. Для преобразованной грамматики реализовать синтаксический анализатор и реализовать программную реализацию этого анализатора.
5. Отчет должен включать:
 - (a) Исходную грамматику;
 - (b) Отдельно (для каждого правила) действия по устранению прямой левой рекурсии и отдельно действия для левой факторизации.
 - (c) Преобразованную грамматику
 - (d) Таблицы множеств FIRST и FOLLOW для нетерминалов;
 - (e) Таблица синтаксического анализатора;
 - (f) Реализацию синтаксического анализатора.
 - (g) Выводы.

Исходная грамматика

$$S \rightarrow ABBC$$
$$B \rightarrow bA \mid bB \mid bC \mid b$$
$$C \rightarrow CCa \mid Ca \mid a \mid c$$
$$A \rightarrow aA \mid aa$$

Устранение левой рекурсии

$$C \rightarrow C Ca \mid C a \mid a \mid c \Rightarrow \begin{cases} C \rightarrow a \mid c \mid aD \mid cD \\ D \rightarrow Ca \mid a \mid CaD \mid aD \end{cases}$$

Стало:

$$S \rightarrow ABBC$$
$$B \rightarrow bA \mid bB \mid bC \mid b$$
$$C \rightarrow CCa \mid Ca \mid a \mid c$$
$$A \rightarrow aA \mid aa$$
$$S \rightarrow ABBC$$
$$B \rightarrow bA \mid bB \mid bC \mid b$$
$$\Rightarrow C \rightarrow a \mid c \mid aD \mid cD$$
$$D \rightarrow Ca \mid a \mid CaD \mid aD$$
$$A \rightarrow aA \mid aa$$

Левая факторизация

$$\begin{aligned}
 S &\rightarrow ABBC \\
 B &\rightarrow \textcolor{blue}{b}A \mid \textcolor{blue}{b}B \mid \textcolor{blue}{b}C \mid \textcolor{blue}{b} \\
 C &\rightarrow \textcolor{green}{a} \mid \textcolor{pink}{c} \mid \textcolor{green}{a}D \mid \textcolor{pink}{c}D \\
 D &\rightarrow \textcolor{red}{Ca} \mid \textcolor{yellow}{a} \mid \textcolor{red}{Ca}D \mid \textcolor{yellow}{a}D \\
 A &\rightarrow \textcolor{orange}{a}A \mid \textcolor{orange}{a}a
 \end{aligned}$$

$$B \rightarrow \textcolor{blue}{b}A \mid \textcolor{blue}{b}B \mid \textcolor{blue}{b}C \mid \textcolor{blue}{b} \Rightarrow \begin{cases} B \rightarrow \textcolor{blue}{b}E \\ E \rightarrow A \mid B \mid C \mid \varepsilon \end{cases}$$

$$A \rightarrow \textcolor{orange}{a}A \mid \textcolor{orange}{a}a \Rightarrow \begin{cases} A \rightarrow \textcolor{orange}{a}F \\ F \rightarrow A \mid a \end{cases}$$

$$C \rightarrow \textcolor{green}{a} \mid \textcolor{pink}{c} \mid \textcolor{green}{a}D \mid \textcolor{pink}{c}D \Rightarrow \begin{cases} C \rightarrow \textcolor{green}{a}G \mid \textcolor{pink}{c} \mid \textcolor{pink}{c}D \\ G \rightarrow D \mid \varepsilon \end{cases} \Rightarrow \begin{cases} C \rightarrow \textcolor{green}{a}G \mid \textcolor{pink}{c}I \\ G \rightarrow D \mid \varepsilon \\ I \rightarrow D \mid \varepsilon \end{cases}$$

$$D \rightarrow \textcolor{red}{Ca} \mid \textcolor{yellow}{a} \mid \textcolor{red}{Ca}D \mid \textcolor{yellow}{a}D \Rightarrow \begin{cases} D \rightarrow \textcolor{red}{Ca}H \mid \textcolor{yellow}{a} \mid \textcolor{yellow}{a}D \\ H \rightarrow D \mid \varepsilon \end{cases} \Rightarrow \begin{cases} D \rightarrow \textcolor{red}{Ca}J \mid \textcolor{yellow}{a}J \\ H \rightarrow D \mid \varepsilon \\ J \rightarrow D \mid \varepsilon \end{cases}$$

Уберем эквивалентные правила:

Исходные правила:

$$G \rightarrow D \mid \varepsilon$$

$$I \rightarrow D \mid \varepsilon$$

$$H \rightarrow D \mid \varepsilon$$

$$J \rightarrow D \mid \varepsilon$$

Упрощенное правило:

$$G \rightarrow D \mid \varepsilon$$

Стало:

$$S \rightarrow ABBC$$

$$B \rightarrow bA \mid bB \mid bC \mid b$$

$$C \rightarrow a \mid c \mid aD \mid cD$$

$$D \rightarrow Ca \mid a \mid CaD \mid aD$$

$$A \rightarrow aA \mid aa$$

$$S \rightarrow ABBC$$

$$B \rightarrow bE$$

$$E \rightarrow A \mid B \mid C \mid \varepsilon$$

$$\Rightarrow C \rightarrow aG \mid cG$$

$$D \rightarrow CaG \mid aG$$

$$G \rightarrow D \mid \varepsilon$$

$$A \rightarrow aF$$

$$F \rightarrow A \mid a$$

Множества FIRST и FOLLOW

FIRST & FOLLOW		
Грамматика:	FIRST	FOLLOW
$S = ABBC$	$FIRST(S) = \{a\}$	$FOLLOW(S) = \{\$ \}$
$B = bE$	$FIRST(B) = \{b\}$	$FOLLOW(B) = \{a, b, c\}$
$E = A \mid B \mid C \mid \varepsilon$	$FIRST(E) = \{a, b, c, \varepsilon\}$	$FOLLOW(E) = \{a, b, c\}$
$C = aG \mid cG$	$FIRST(C) = \{a, c\}$	$FOLLOW(C) = \{\$, a, b, c\}$
$D = CaG \mid aG$	$FIRST(D) = \{a, c\}$	$FOLLOW(D) = \{\$, a, b, c\}$
$G = D \mid \varepsilon$	$FIRST(G) = \{a, c, \varepsilon\}$	$FOLLOW(G) = \{\$, a, b, c\}$
$A = aF$	$FIRST(A) = \{a\}$	$FOLLOW(A) = \{a, b, c\}$
$F = A \mid a$	$FIRST(F) = \{a\}$	$FOLLOW(F) = \{a, b, c\}$

* * *

$$FIRST(S) = FIRST(A) = \{a\}$$

$$FIRST(B) = \{b\}$$

$$FIRST(E) = \begin{cases} FIRST(A) = a \\ FIRST(B) = b \\ FIRST(C) = a, c \\ \varepsilon \end{cases} = \{a, b, c, \varepsilon\}$$

$$FIRST(C) = \{a, c\}$$

$$FIRST(D) = \begin{cases} FIRST(C) = \{a, c\} \\ a \end{cases} = \{a, c\}$$

$$FIRST(G) = \begin{cases} FIRST(D) = \{a, c\} \\ \varepsilon \end{cases} = \{a, c, \varepsilon\}$$

$$FIRST(A) = \{a\}$$

$$FIRST(F) = \begin{cases} FIRST(A) = \{a\} \\ a \end{cases} = \{a\}$$

* * *

$$FOLLOW(S) = \{\$ \}$$

$$FOLLOW(B) = \begin{cases} FIRST(B) = \{b\} \\ FOLLOW(E) = \{a, b, c\} \end{cases} = \{a, b, c\}$$

$$FOLLOW(E) = FOLLOW(B) = \{a, b, c\}$$

$$FOLLOW(C) = \begin{cases} FOLLOW(S) = \{\$ \} \\ FOLLOW(E) = \{a, b, c\} \end{cases} = \{\$, a, b, c\}$$

$$FOLLOW(D) = FOLLOW(G) = \{\$, a, b, c\}$$

$$FOLLOW(G) = FOLLOW(C) = \{\$, a, b, c\}$$

$$FOLLOW(A) = \begin{cases} FIRST(B) = \{b\} \\ FOLLOW(E) = \{a, b, c\} \end{cases} = \{a, b, c\}$$

$$FOLLOW(F) = FOLLOW(A) = \{a, b, c\}$$

Множество Nullable

A	B	C	D	E	F	G	S
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE

Таблица синтаксического анализатора

NTs	Input Symbols			
	a	b	c	\$
S	$S \rightarrow ABBC$			
B		$B \rightarrow bE$		
E	$E \rightarrow A \mid C \mid \varepsilon$	$E \rightarrow B \mid \varepsilon$	$E \rightarrow C \mid \varepsilon$	
C	$C \rightarrow aG$		$C \rightarrow cG$	
D	$D \rightarrow CaG \mid aG$		$D \rightarrow CaG$	
G	$G \rightarrow D \mid \varepsilon$	$G \rightarrow \varepsilon$	$G \rightarrow D \mid \varepsilon$	$G \rightarrow \varepsilon$
A	$A \rightarrow aF$			
F	$F \rightarrow A \mid a$			

Так как A для пары (X, c) применимо более одного правила, то это не LL(1) грамматика.

Насильно преобразуем к LL(1).

Удаляем правила:

$E \rightarrow C \mid \varepsilon$

$D \rightarrow CaG \mid \varepsilon$

$G \rightarrow D$

$F \rightarrow A \mid \varepsilon$

Грамматика:

$S \rightarrow ABBC$

$B \rightarrow bE$

$E \rightarrow A \mid b$

$C \rightarrow a \mid c$

$A \rightarrow aF$

$F \rightarrow a$

FIRST & FOLLOW		
Грамматика:	FIRST	FOLLOW
$S \rightarrow ABBC$	$FIRST(S) = \{a\}$	$FOLLOW(S) = \{\$ \}$
$A \rightarrow aF$	$FIRST(A) = \{a\}$	$FOLLOW(A) = \{a, b, c\}$
$B \rightarrow bE$	$FIRST(B) = \{b\}$	$FOLLOW(B) = \{a, b, c\}$
$C \rightarrow a \mid c$	$FIRST(C) = \{a, c\}$	$FOLLOW(C) = \{\$ \}$
$E \rightarrow A \mid B$	$FIRST(E) = \{a, b\}$	$FOLLOW(E) = \{a, b, c\}$
$F \rightarrow a$	$FIRST(F) = \{a\}$	$FOLLOW(F) = \{a, b, c\}$

NTs	Input Symbols			
	a	b	c	\$
S	$S \rightarrow ABBC$			
B		$B \rightarrow bE$		
E	$E \rightarrow A$	$E \rightarrow b$		
C	$C \rightarrow a$		$C \rightarrow c$	
A	$A \rightarrow aF$			
F	$F \rightarrow a$			

Программа-распознаватель

Код

```

1 parse_table = {
2     "S": {
3         "a": ["A", "B", "B", "C"]
4     },
5     "B": {
6         "b": ["b", "E"]
7     },
8     "E": {
9         "a": ["A"],
10        "b": ["b"]
11    },
12    "C": {
13        "a": ["a"],
14        "c": ["c"]
15    },
16    "A": {
17        "a": ["a", "F"],
18    },
19    "F": {
20        "a": ["a"],
21    },
22 }
23 class LL1Parser:
24     def __init__(self, parse_table, start_symbol, end_symbol):
25         self.parse_table = parse_table
26         self.start_symbol = start_symbol
27         self.end_symbol = end_symbol

```

```

28
29 def parse(self, input_string):
30     input_queue = list(input_string) + [self.end_symbol]
31     parse_stack = [self.end_symbol, self.start_symbol]
32
33     while parse_stack:
34         top = parse_stack.pop()
35         current_input = input_queue[0]
36         if top == current_input:
37             input_queue.pop(0)
38         elif top in self.parse_table:
39             if current_input in self.parse_table[top]:
40                 rule = self.parse_table[top][current_input]
41                 if rule != 'ε':
42                     parse_stack.extend(reversed(rule))
43                 print(f"{top} → {''.join(rule)}")
44             else:
45                 raise SyntaxError(
46                     f"Unexpected symbol:
47                         {current_input}.\nSentence {input_string}
48                         does not belong to this grammar.")
49         else:
50             raise SyntaxError(
51                 f"Unexpected symbol: {top}.\nSentence
52                     {input_string} does not belong to this
53                     grammar.")
54
55     if input_queue and input_queue[0] != self.end_symbol:
56         raise SyntaxError(
57             f"Parsing did not complete properly.\nSentence
58                 {input_string} does not belong to this grammar.")
59
60     print(f"Parsing completed successfully.\nSentence
61         {input_string} belongs to this grammar.")
62
63
64 def main():
65     print("LL(1) analyzer for CS grammar")
66
67     parser = LL1Parser(parse_table, "S", "$")
68
69     sentence: str = input("Enter your sentence: ")
70
71     try:
72         parser.parse(sentence)
73     except SyntaxError as e:
74         print(e)
75
76 if __name__ == "__main__":
77     main()

```

Вывод

```
> python3 main.py
LL(1) analyzer for CS grammar
Enter your sentence: aabbbbc
S → ABBC
A → aF
F → a
B → bE
E → b
B → bE
E → b
C → c
Parsing completed successfully.
Sentence aabbbbc belongs to this grammar.
```

```
> python3 main.py
LL(1) analyzer for CS grammar
Enter your sentence: acb
S → ABBC
A → aF
Unexpected symbol: c.
Sentence acb does not belong to this grammar.
```