

Отчет

Александр Андреев М3336

December 2021

1 Введение

Цель данной лабораторной работы — научиться разрабатывать грамматики для заданных неформально языков с учетом семантического

смысла и приоритета операторов, разрабатывать лексические анализаторы, разрабатывать вручную нисходящие синтаксические анализаторы, разрабатывать подробные тестовые наборы для анализаторов.

Форма отчетности: программа и текстовый отчет. Программа может быть написана на любом языке программирования общего назначения (рекомендуются Си++ и Java).

2 Описание заголовка функции в Kotlin

Заголовок функции в Kotlin. Заголовок начинается ключевым словом “fun”, далее идет имя функции, скобка, несколько описаний аргументов через запятую, затем может идти двоеточие и имя возвращаемого типа. Используйте один терминал для всех имен переменных. Используйте один терминал для ключевых слов fun и т. п. (не несколько ‘f’, ‘u’, ‘n’). 9 Пример: fun printSum(a: Int, b: Int): Unit

3 Грамматика для заголовка функции в Kotlin

```
S -> fun FUN_NAME ( PARAMS ) RETURN_VAL
PARAMS -> PARAM TAIL
PARAMS -> eps
PARAM -> VAR : TYPE
TAIL -> , PARAM TAIL
TAIL -> eps
RETURN_VAL -> : RETURN_TYPE
RETURN_VAL -> eps
FUN_NAME -> [a-zA-Z_][a-zA-Z0-9_]*
TYPE -> [a-zA-Z_][a-zA-Z0-9_]*
VAR -> [a-zA-Z_][a-zA-Z0-9_]*
RETURN_TYPE -> [a-zA-Z_][a-zA-Z0-9_]*
```

3.1 Описание грамматики:

Не терминал	Описание
S	Определение функции в Kotlin
PARAMS	Параметры функции
PARAM	Один не пустой параметр
TAIL	Остальные параметры
RETURN_VAL	Возвращаемое значение функции
FUN_NAME	Имя функции
TYPE	Имя типа переменной
VAR	Имя переменной
RETURN_TYPE	Тип возвращаемого значения

Нам повезло, в нашей грамматике нет левой рекурсии и правого ветвления

3.2 Построим множества First и Follow:

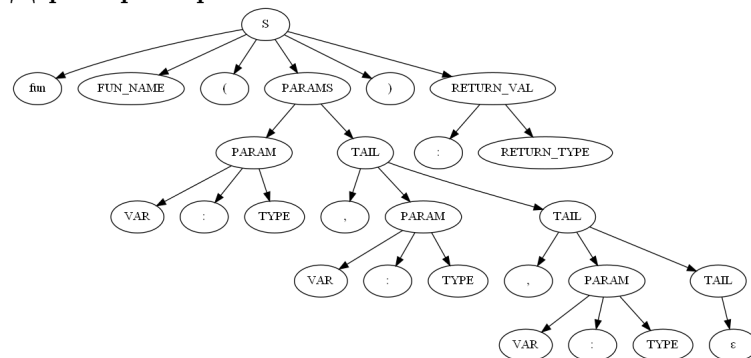
Не терминал	First	Follow
S	'fun'	\$
PARAMS	ϵ , [a-zA_Z_][a-zA_Z0-9_]*	'('
PARAM	[a-zA_Z_][a-zA_Z0-9_]*	',' , '('
TAIL	',' , ϵ	'('
RETURN_VAL	':' , ϵ	\$
FUN_NAME	[a-zA_Z_][a-zA_Z0-9_]*	'('
TYPE	[a-zA_Z_][a-zA_Z0-9_]*	',' , '('
VAR	[a-zA_Z_][a-zA_Z0-9_]*	':'
RETURN_TYPE	[a-zA_Z_][a-zA_Z0-9_]*	\$

4 Запуск

Входная строка:

```
"    fun name_112ppp(final: do    , a:Int, b :String): Int64 "
```

Дерево разбора:



5 Модификация

Добавлена поддержка функциональных типов.

```

S -> fun FUN_NAME ( PARAMS ) RETURN_VAL
PARAMS -> PARAM TAIL
PARAMS -> eps
PARAM -> VAR : TYPE
TAIL -> , PARAM TAIL
TAIL -> eps
RETURN_VAL -> : RETURN_TYPE
RETURN_VAL -> eps
FUN_NAME -> [a-zA_Z_][a-zA_Z0-9_]*
TYPE -> [a-zA_Z_][a-zA_Z0-9_]*
TYPE -> (TYPES) -> TYPE
VAR -> [a-zA_Z_][a-zA_Z0-9_]*
RETURN_TYPE -> [a-zA_Z_][a-zA_Z0-9_]*
RETURN_TYPE -> (TYPES) -> RETURN_TYPE
TYPES -> eps
TYPES -> TYPE TYPES_TAIL
TYPES_TAIL -> , TYPE TYPES_TAIL
TYPES_TAIL -> eps

```

5.1 Построим множества First и Follow:

Не терминал	First	Follow
S	'fun'	\$
PARAMS	ϵ , [a-zA_Z_][a-zA_Z0-9_]*)'
PARAM	[a-zA_Z_][a-zA_Z0-9_]*	',' , '('
TAIL	',' , ϵ)'
RETURN_VAL	',' , ϵ	\$
FUN_NAME	[a-zA_Z_][a-zA_Z0-9_]*	('
TYPE	[a-zA_Z_][a-zA_Z0-9_]*, '('	',' , '('
VAR	[a-zA_Z_][a-zA_Z0-9_]*	':'
RETURN_TYPE	[a-zA_Z_][a-zA_Z0-9_]*, '('	\$
TYPES	ϵ , [a-zA_Z_][a-zA_Z0-9_]*, '(')'
TYPES_TAIL	',' , ϵ)'