

## 1 Лабораторная работа №2

Дана грамматика массивов в языке Kotlin - необходимо создать для нее парсер.

Пример выражения из грамматики - `"var x : Array<Int>;"`

Заведем все терминальные символы:

`"var"` — терминал обозначающий начало определения массива.

`"Array"` — терминал обозначающий начало задания типа массива.

`","` — терминал разделитель для перечисления дженериков.

`":"` — терминал разделитель между именем переменной и типом массива.

`"n/name"` — терминал обозначающий имя типа (для простоты просто последовательность букв).

`"<"` — терминал обозначающий начало определения параметризованных типов.

`">"` — терминал обозначающий конец определения параметризованных типов.

`";"` — терминал обозначающий конец определения массива.

`"$"` — терминал обозначающий конец строки.

Далее заведем список нетерминалов и составим для них правила выводящие нашу грамматику:

Нетерминал	Смысл
S	Определение массива в Kotlin
E	Корректный тип данных (возможно параметризованный)
T	Список параметров-дженериков.

Правило	Смысл
$S \rightarrow var\ n : Array < E >; \$$	Неизменная конструкция объявления
$S \rightarrow var\ Array : Array < E >; \$$	Случай когда имя переменной это Array
$E \rightarrow n$	Непараметризованный тип
$E \rightarrow n < T >$	Тип параметризованный списком типов
$E \rightarrow Array < E >$	Массив параметризованный одним типом
$T \rightarrow T, E$	Список это список и хвост из одного типа
$T \rightarrow E$	Список это один тип

(P.S. некоторое извращение приходится делать от того, что Array должен иметь ровно один параметризующий тип и Array может являться именем переменной)

Можно заметить что правила с нетерминалом T образуют непосредственную левую рекурсию за счет правила " $T \rightarrow T, E$ ", а правила с нетерминалом E правое ветвление за счет двух правил " $E \rightarrow n$ " и " $E \rightarrow n < T >$ "

Также формально правая рекурсия наблюдается и в правилах с S.

Все это необходимо исправить стандартным образом, после чего получится следующий набор нетерминалов и правил к ним:

Нетерминал	Смысл
S	Определение массива в Kotlin
S'	Определение массива в Kotlin после var
E	Корректный тип данных (возможно параметризованный)
T	Список параметров-дженериков.
E'	Параметризация (в том числе пустая)
T'	Продолжение списка параметров-дженериков

Правило	Смысл
$S \rightarrow var S'$	Неизменная конструкция объявления
$S' \rightarrow n : Array < E >; \$$	Случай когда имя переменной это не Array
$S' \rightarrow Array : Array < E >; \$$	Случай когда имя переменной это Array
$E \rightarrow Array < E >$	Массив параметризованный одним типом
$E \rightarrow nE'$	Корректный тип (мб параметризованный)
$E' \rightarrow \varepsilon$	Пустая параметризация
$E' \rightarrow < T >$	Параметризация списком
$T \rightarrow ET'$	Список из головы и продолжения
$T' \rightarrow \varepsilon$	Отсутствие продолжения
$T' \rightarrow , ET'$	Продолжение бьется на голову и продолжение

Осталось построить множества FIRST и FOLLOW, затем с помощью них проверить, что грамматика действительно LL(1), а потом построить нисходящий парсер.

Нетерминал	First	Follow
S	var	\$
S'	n Array	\$
E	n Array	> ,
T	n Array	>
E'	$\varepsilon$ <	> ,
T'	, $\varepsilon$	>