



## Лабораторная работа 3

### Теория: анализ структур ЯП

Проанализировать синтаксические структуры ЯП, на котором вы реализовывали 1 или 2 л.р.. Ваш вариант вычисляется как остаток от деления на 4 вашего номера в списке группы.

- Вариант 0 — найти синтаксическую структуру, которую можно распознать с помощью DPDA с допуском по пустому стеку, но не FA.
- Вариант 1 — найти синтаксическую структуру, которую можно распознать с помощью FA, но не DPDA с допуском по пустому стеку.
- Вариант 2 — найти синтаксическую структуру, которую нельзя распознать с помощью DPDA с допуском по пустому стеку и FA, но можно с помощью CFG.
- Вариант 3 — найти синтаксическую структуру, которую нельзя распознать с помощью CFG.



## Лабораторная работа 3

### Практика: CFG и Graphviz

- 1 Путём анализа накачек грамматики провести частичный анализ её регулярности (чет в списке группы). Результат должен быть представлен как в текстовом виде, так и в виде диаграммы Graphviz развертки нетерминалов, для которых найдены подозрительные накачки.
- 2 По PDA с допуском по пустому стеку построить CF-грамматику, порождающую тот же язык. Результат должен включать графическое представление исходного PDA в виде диаграммы Graphviz.



## Синтаксис входных данных

Чёрным обозначены элементы метаязыка, красным — элементы языка входных данных. Чтение данных осуществляется из файла. Расстановка пробелов произвольна. Синтаксис входных данных CFG. Считаем, что грамматика находится в квази-GNF:

$$\begin{aligned}\langle \text{grammar} \rangle &::= \langle \text{rule} \rangle^+ \\ \langle \text{rule} \rangle &::= \langle \text{nterm} \rangle \textcolor{red}{->} [a-z] \langle \text{term} \rangle^* \\ \langle \text{term} \rangle &::= \langle \text{nterm} \rangle \mid [a-z] \\ \langle \text{nterm} \rangle &::= [A-Z][0-9]^*\end{aligned}$$

Синтаксис PDA. Здесь **!!** — пустое слово, **\$** — конец строки:

$$\begin{aligned}\langle \text{PDA} \rangle &::= \textcolor{red}{<} \langle \text{state} \rangle, \langle \text{bottom} \rangle \textcolor{red}{>} \$ (\langle \text{transition} \rangle \$)^+ \\ \langle \text{transition} \rangle &::= \textcolor{red}{<} \langle \text{state} \rangle, (\langle \text{letter} \rangle \mid \textcolor{red}{!!}), \langle \text{stack\_s} \rangle \textcolor{red}{->} \textcolor{red}{<} \langle \text{state} \rangle, \langle \text{stack\_s} \rangle^* \textcolor{red}{>} \\ \langle \text{state} \rangle &::= [q-u][0-9]? \\ \langle \text{stack\_s} \rangle &::= [A-Z][0-9]? \\ \langle \text{letter} \rangle &::= [a-z]\end{aligned}$$



## Анализ накачек языка

Задача разбивается на следующие этапы:

- 1 Анализ регулярных подмножеств грамматики.  
Нахождение максимальных множеств  $M_i$  нетерминалов  $V_j$  таких, что все правые части правил вида  $V_j \rightarrow \dots$  содержат только нетерминалы из  $M_i$ , причём все эти части праволинейны.
- 2 Развёртка дерева левосторонних разборов исходной грамматики. Для каждого достижимого из стартового нетерминала  $A$  строим дерево развёртки до первых накачек вида  $\Phi_1 A \Phi_2$ , где  $\Phi_1$  — терминальная строка. Если оказалось, что  $\Phi_2$  состоит только из терминалов или регулярных нетерминалов (входящих в какое-нибудь из  $M_i$ ), тогда проверяем, входит ли  $\Phi_1$  в язык  $\Phi_2^+$ . Если не входит, тогда выводим дерево накачки нетерминала  $A$  как подозрительное на нерегулярную накачку.



## Продолжение

- 3 Если  $\Phi_1 \in L(\Phi_2^+)$ , тогда проверяем все кратчайшие конечные пути развёртки  $A$  до терминальной строки на вхождение в  $L(\Phi_2^+)$ . Если построенные на них строки также входят в  $L(\Phi_2^+)$ , сообщаем о возможной регулярности языка  $A$ . Если  $A \in M_i$ , сразу сообщаем о его регулярности.
- 4 Рекурсивно замыкаем множества регулярных и возможно регулярных нетерминалов. Если при переписывании нетерминала  $B$  все правые части содержат только регулярные нетерминалы, он регулярен. Если регулярные и возможно регулярные — возможно регулярен.
- 5 Если рекурсивное замыкание не дало никакой информации об исходном нетерминале  $S$ , но не было и подозрительных нерегулярных накачек  $S$ , сообщаем, что регулярность языка не удалось определить.



## Преобразование PDA в CFG

- 1 Для построения первого варианта CFG можно воспользоваться переборным алгоритмом порождения произведений путей.
- 2 Из результирующей CFG должны быть удалены все непродуктивные или недостижимые нетерминалы, а также правила, их включающие.
- 3 Оставшиеся нетерминалы должны быть переименованы в нетерминалы с именами вида  $[A-Z][0-9]^*$ . Стартовый нетерминал должен иметь имя S.



## Graphviz

- 1 В чётной задаче (анализ накачек) должны быть построены деревья развёртки всех нетерминалов до первой накачки. Деревья — это графы, ориентированные сверху вниз. В узлах деревьев, содержащих ветвления, стоят нетерминальные символы, в листьях — нетерминальные или терминальные. Если развертка нетерминала не содержит накачек (нет рекурсии), либо нетерминал попал в множество  $M_i$ , тогда такое дерево строить не нужно.
- 2 В нечётной задаче должна быть построена схема исходного PDA. Ориентация схемы — слева направо.