

Практическая работа № 4

Задания

1. Постройте НКА, распознающие следующие множества цепочек:

а) abc, abd и aacd. Входным алфавитом считать {a, b, c, d};

б) 0101, 101 и 011;

в) ab, bc и ca. Входным алфавитом считать {a, b, c}.

2. Преобразуйте НКА из (1) в ДКА

3. Преобразуйте следующий НКА в эквивалентный ДКА и опишите неформально язык, который он допускает

	0	1
$\rightarrow p$	{p,q}	{p}
q	{r,s}	{t}
r	{p,r}	{t}
*s	\emptyset	\emptyset
*t	\emptyset	\emptyset

4. Рассмотрите следующий ϵ -НКА

	ϵ	a	b	c
$\rightarrow p$	\emptyset	{p}	{q}	{r}
q	{p}	{q}	{r}	\emptyset
*r	{q}	{r}	\emptyset	{p}

а) найдите ϵ -замыкание каждого из состояний;

б) выпишите все цепочки, длина которых не более 3, допустимые данным автоматом;

5. Опишите обычными словами языки следующих регулярных выражений:

а) $(1 + \epsilon)(00^* 1)^* 0^*$;

б) $(0^* 1^*)^* 000(0 + 1)^*$;

в) $(0 + 10)^* 1^*$

6. Напишите регулярное выражение для описания телефонных номеров. Мобильных и городских.

7. Дано регулярное выражение $(0 + 1)^* 1(0 + 1) + (0 + 1)^* 1(0 + 1)(0 + 1)$.

С помощью дистрибутивных законов преобразуйте его в два различных, более простых, эквивалентных выражения.

8. Постройте конечный автомат для распознавания в тексте переменных. Для простоты будем считать: имена переменных, функций, методов должны содержать строчные и прописные буквы латинского алфавита, цифры от 0 до 9, знак `_`. При этом начинаться имя должно с буквы. Для расширения: используйте также выделение специальных зарезервированных «магических имён», которые играют особую роль в Python (возьмем, `__init__`, `__import__`, `__file__`).

9. Постройте конечный автомат для распознавания в тексте зарезервированных слов. Для простоты возьмем следующие слова: `class`, `for`, `while`, `if`, `elif`, `else`, `function`, `procedure`.

10. Напишите регулярное выражение для проверки корректности использования скобок — все скобки должны быть в паре: `()`, `[]`, `{}`.