Лабораторная работа №1

Регулярные языки

Задание. Для заданного регулярного языка A над алфавитом $\Sigma = \{0, 1\}$

1. построить диаграмму состояний ДКА, распознающего A;
2. реализовать данный ДКА в виде программы, которая для произвольной входной строки w должна выводить историю вычислений ДКА на ней в виде последовательности состояний.

Варианты

1. $A = \{w\colon w$ начинается с 1 и заканчивается на 0$\}$;
2. $A = \{w\colon w$ содержит хотя бы три 1$\}$;
3. $A = \{w\colon w$ состоит хотя бы из 3-х символов и при этом третий символ – 0$\}$;
4. $A = \{w\colon w$ начинается с 0 и имеет чётную длину или начинается с 1 и имеет нечётную длину$\}$;
5. $A = \{w\colon w$ содержит 0101 в качестве подстроки$\}$;
6. $A = \{w\colon w$ не содержит 110 в качестве подстроки$\}$;
7. $A = \{w\colon w$ имеет длину не более пяти$\}$;
8. $A = \{w\colon w$ является любой строчкой, кроме 11 и 111$\}$;
9. $A = \{w\colon w$ на каждой нечётной позиции содержит 1$\}$;
10. $A = \{w\colon w$ содержит хотя бы два 0 и не более одной 1$\}$;
11. $A = \{w\colon w$ содержит чётное число 0 или ровно две 1$\}$;
12. $A = \{w\colon w$ содержит одинаковое число подстрок 01 и 10$\}$;
13. $A = \{w\colon w$ содержит ровно три 0 или ровно три 1$\}$;
14. $A = \{w\colon w$ содержит чётное число и 0, и 1$\}$;
15. $A = \{w\colon w$ содержит нечётное число и 0, и 1$\}$.

Лабораторная работа №2

Контекстно-свободные языки

Задание. Для заданного КС-языка B над алфавитом $\Sigma = \{0, 1\}$

1. построить диаграмму состояний МП-автомата, распознающего A;
2. реализовать данный МП-автомат в виде программы, которая для произвольной входной строки w должна выводить историю вычислений МП-автомата на ней в виде последовательности состояний и содержимого стэка памяти.

Варианты

1. $B = \{w\colon w$ содержит 0 не меньше, чем 1$\}$;
2. $B = \{w\colon w$ содержит 1 больше, чем 0$\}$;
3. $B = \{w\colon w$ содержит 1 в два раза больше, чем 0$\}$;
4. $B = \{0^n1^i0^j\colon n = i$ или $n = j\}$;
5. $B = \{0^{2n}1^n\colon n \geqslant 0\}$;
6. $B = \{w\colon w = 1^i0^n1^j\colon n = i$ или $n = j\}$;
7. $B = \{w\colon w$ не содержит 0 в три раза меньше, чем 1$\}$;
8. $B = \{w\colon w$ содержит 0 по крайней мере в два раза больше, чем 1$\}$;
9. $B = \{w\colon w$ содержит 0 не более, чем в два раза больше, чем 1$\}$;
10. $B = \{w\colon w$ после каждой подстроки 0^n, $n > 0$, сразу содержит подстроку $1^n\}$;
11. $B = \{w\colon w$ после каждой подстроки 1^n, $n > 0$, сразу содержит подстроку 0^i, $i > n\}$;
12. $B = \{w\colon w$ содержит различное число 0 и 1$\}$;
13. $B = \{w\colon w$ содержит подстроку 01 столько же раз, сколько в начале w расположено 0$\}$;
14. $B = \{w\colon w$ содержит подстроку 01 столько же раз, сколько в конце w расположено 1$\}$;
15. $B = \{w\colon w$ содержит подстроку 01 меньше раз, чем в начале w расположено 0$\}$.

Лабораторная работа №3

Разрешимость по Тьюрингу

Задание. Для заданного языка, в котором предполагается, что КС-грамматика, регулярное выражение или ДКА определены над алфавитом $\{0, 1\}$,

1. построить описание МТ, решающей его;
2. реализовать данную МТ в виде программы.

Варианты

1. $C_{CFG} = \{\langle G, k\rangle \colon G$ – КС-грамматика, $\mathrm{card}((L(G)) = k,\ k \in \mathbb{N} \cup \{0, \infty\}\}$;
2. $C = \{\langle G, x\rangle \colon G$ – КС-грамматика, $\exists y, z \in \Sigma^* \colon yxz \in L(G)\}$;
3. $A = \{\langle R\rangle \colon R$ – регулярное выражение, $\exists x, y \in \Sigma^* \colon x111y \in L(R)\}$;
4. $INFINITE_{DFA} = \{\langle A\rangle \colon A$ – ДКА и $L(A)$ – бесконечный язык$\}$;
5. $A_{\varepsilon,CFG} = \{\langle G\rangle \colon G$ – КС-грамматика и $\varepsilon \in L(A)\}$;
6. $ALL_{DFA} = \{\langle A\rangle \colon A$ – ДКА и $L(A) = \Sigma^*\}$;
7. $BAL_{DFA} = \{\langle M\rangle \colon M$ – ДКА, который допускает некоторую строку состоящую из одинакового числа 0 и 1$\}$;
8. $PAL_{DFA} = \{\langle M\rangle \colon M$ – ДКА, который допускает некоторый палиндром$\}$;
9. $E = \{\langle M\rangle \colon M$ – ДКА, который допускает некоторую строку, в которой 1 больше, чем 0$\}$;
10. $E_{DFA} = \{\langle A\rangle \colon A$ – ДКА и $L(A) = \varnothing\}$;
11. $A = \{\langle G\rangle \colon G$ – КС-грамматика и $1^* \subset L(G)\}$;
12. $S = \{\langle M\rangle \colon M$ – ДКА и $w \in L(M) \iff w^{\mathcal{R}} \in L(M)\}$;
13. $A = \{\langle M\rangle \colon M$ – ДКА, который не допускает строки, сожержащие нечетное число 1$\}$;
14. $EQ_{DFA} = \{\langle M_1, M_2\rangle \colon M_1, M_2$ – ДКА и $L(M_1) = L(M_2)\}$;
15. $A_{REX} = \{\langle R, w\rangle \colon R$ – регулярное выражение и $w \in L(R)\}$.

Лабораторная работа №4

Исследование временной сложности

Задание. Для заданного языка

1. построить описание МТ с полиномиальной временной сложностью, решающей его;
2. построить оценку сложности данной МТ;
3. реализовать данную МТ в виде программы;
4. провести тестовые исследования, демонстрирующие совпадение фактической временной сложности с теоретической.

Варианты

1. $CONNECTED = \{\langle G \rangle \colon G$ – связанный неориентированный граф$\}$;
2. $TRIANGLE = \{\langle G \rangle \colon G$ – неориентированный граф, содержащий 3-клику$\}$.
3. $MODEXP = \{\langle a, b, c, p \rangle \colon a, b, c, p$ – бинарные целые числа такие, что $a^b = c \bmod p\}$;
4. $PERM - POWER = \{\langle p, q, t \rangle \colon p, q$ – перестановки такие, что $p = q^t\}$, где перестановкой будем называть взаимнооднозначное отображение $p \colon \{1, \ldots, k\} \to \{1, \ldots, k\}$;
5. $UNARY{-}SSUM = \{\langle s, t \rangle \colon s = \{x_1, \ldots, x_k\}$, существует $\{y_1, \ldots, y_l\} \subset \{x_1, \ldots, x_k\} \colon \sum\limits_{i=1}^{l} y_i = t\}$, где все числа $x_1, \ldots, x_k, t$ представлены в унарном алфавите;
6. $PRIMES = \{m \colon m$ – бинарное простое число$\}$;
7. $SPATH = \{\langle G, a, b, k \rangle \colon G$ – неориентированный граф, содержащий простой путь не длиннее k из вершины a в вершину $b\}$;
8. $CNF_2 = \{\langle \varphi \rangle \colon \varphi$ – выполнимая булева формула в КНФ, каждая переменная в которой появляется не более, чем в 2 позициях$\}$;
9. $2SAT = \{\langle \varphi \rangle \colon \varphi$ – выполнимая булева формула в КНФ, в которой каждая скобка содержит не более двух литералов$\}$;
10. $RELPRIME = \{\langle x, y \rangle \colon x$ и y – взаимнопростые числа$\}$;
11. $UCYCLE = \{\langle G \rangle \colon G$ – неориентированный граф, который содержит простой цикл$\}$;
12. $CYCLE = \{\langle G \rangle \colon G$ – направленный граф, который содержит направленный цикл$\}$;
13. $BIPARTITE = \{\langle G \rangle \colon G$ – двудольный неориентированный граф$\}$, где двудольным называется граф вершины которого могут быть разбиты на два непересекающихся подмножества так, что не существует рёбер между вершинами из одного и того же подмножества;
14. $STRONGLY - CONNECTED = \{\langle G \rangle \colon G$ – сильно связанный ориентированный граф, т.е. для любых вершин a и b существуют направленные пути из a в b и из b в $a\}$;
15. $UPATH = \{\langle G, a, b \rangle \colon G$ – неориентированный граф, в котором нет пути из вершины a в вершину $b\}$.

Лабораторная работа №5

Верификаторы

Задание. Для заданного языка

1. построить описание верификатора с полиномиальной временной сложностью и соответствующего сертификата принадлежности;

2. реализовать данный верификатор в виде программы;

3. провести тестовые исследования, демонстрирующие совпадение фактической временной сложности с теоретической.

Варианты

1. $ISO = \{\langle G, F \rangle : G, F$ – изоморфные графы$\}$, где два графа G и F называются изоморфными, если можно переобозначить вершины одного графа G так, чтобы G и F оказались идентичны;

2. $HALF-CLIQUE = \{\langle G \rangle : G$ – неориентированный граф с m вершинами, который содержит клику размером не менее $\frac{m}{2}\}$;

3. $LPATH = \{\langle G, a, b, k \rangle : G$ – неориентированный граф, содержащий простой путь не короче k из вершины a в вершину $b\}$;

4. $DOUBLE-SAT = \{\langle \varphi \rangle : \varphi$ – булева формула, допускающая по крайней мере 2 подстановки, обращающие её в истину$\}$;

5. $CNF_3 = \{\langle \varphi \rangle : \varphi$ – выполнимая булева формула в КНФ, каждая переменная в которой появляется не более, чем в 3 позициях$\}$;

6. $MAX-CUT = \{\langle G, k \rangle : G$ – неориентированный граф, который допускает разрез размером k или более$\}$, где разрезом графа называется разбиение его вершин на два непересекающихся подмножества, размером разреза называется число дуг, соединяющих вершины из различных подмножеств разделения;

7. $3COLOR = \{\langle G \rangle : G$ – неориентированный граф, вершины которого могут быть окрашены в три цвета так, чтобы никакие две смежные вершины не были окрашены одинаково$\}$;

8. $SET-SPLITING = \{\langle S, C \rangle : S$ – конечное множество, $= \{C_1, \ldots, C_N\}$, $C_i \subset S$, $N > 0$, при этом все элементы S могут быть окрашены в два цвета так, чтобы в составе C не было одноцветных подмножеств$\}$;

9. $DOMINATING-SET = \{\langle G, k \rangle : G$ – неориентированный граф, который имеет поглощающее множество размера $k\}$, где поглощающим множеством называется подмножество вершин графа такое, что каждая вершина графа является смежной хотя бы с одной вершиной из данного подмножества;

10. $CLIQUE = \{\langle G, k \rangle : G$ – неориентированный граф, содержащий k-клику$\}$;

11. $HAMPATH = \{\langle G, s, t \rangle : G$ – ориентированный граф, в котором есть гамильтонов путь из вершины s в вершину $t\}$;

12. $VERTEX-COVER = \{\langle G, k \rangle : G$ – неориентированный граф, содержащий покрытие размера $k\}$;

13. $FEEDBACK-VSET = \{\langle G, k\rangle \colon G$ – неориентированный граф, в котором удалив не более k вершин, можно устранить все циклы$\}$;

14. $FEEDBACK-VSET = \{\langle G, k\rangle \colon G$ – ориентированный граф, в котором удалив не более k рёбер, можно устранить все циклы$\}$;

15. $SET-PACKING = \{\langle S, C, k\rangle \colon S$ – конечное множество, $=\{C_1, \ldots, C_N\}$, $C_i \subset S$, $N > 0$, при этом существует набор попарно не пересекающихся подмножеств $C_{i_1}, \ldots, C_{i_k} \in C\}$.

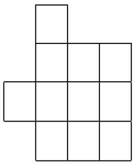
Лабораторная работа №6

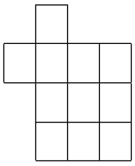
Игровые задачи

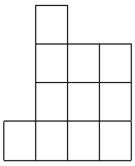
Задание. Для модифицированного поля игры в «крестики-нолики»

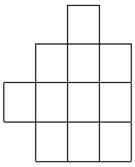
1. построить программными средствами ориентированный граф состояний игры;
2. определить существует ли выигрышная стратегия для первого игрока;
3. реализовать данную стратегию (если она существует) или стратегию приводящую к гарантированной ничьей (если выигрышной стратегии не существует) в виде интерактивной игры, в которой первый ход делает компьютер, а оппонентом выступает игрок.

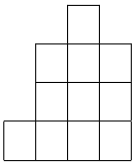
Варианты

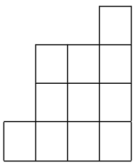
1. 

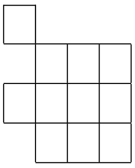
2. 

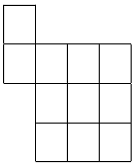
3. 

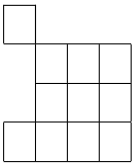
4. 

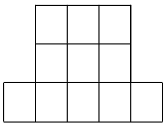
5. 

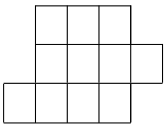
6. 

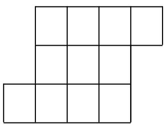
7. 

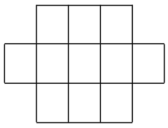
8. 

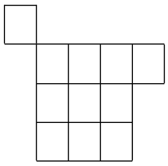
9. 

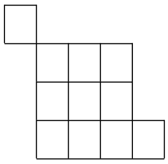
10. 

11. 

12. 

13. 

14. 

15. 

16.