Задание 3

Коновалов Андрей, 074

0	1	2	3	4	5	σ

Задача 0

(ii). Для начала научимся из множества состояний Q автомата $\mathcal{A}$ строить множество состояний Q_ε, каждый элемент которого достижим по ε-переходам из какого-то из состояний Q. Конкретно в данном случае мы будем добавлять элементы в Q в некотором порядке и в итоге получим множество Q_ε.

При построении будем использовать конструкцию "очередь". Это некоторый объект, над которым можно совершать 2 операции: положить туда элемент и достать оттуда элемент. Причем элементы достаются в том же порядке, в котором кладутся.

Для построения множества Q_ε будем использовать алгоритм, известный как поиск в ширину. Сопоставим каждому состоянию $\mathcal{A}$ элемент множества $\{true, false\}$. Состояния, помеченные $true$ будем называть посещенными, а Состояния, помеченные $false$ - не посещенными. Изначально пометим все состояния $\mathcal{A}$ как непосещенные.

Пусть дано множество Q. Пометим все состояния из Q как посещенные. Возьмем пустую очередь и последовательно положим туда все элементы из Q. Теперь, до тех пор пока очередь не пуста, будем доставать из нее элементы. Пусть мы достали из очереди элемент q. Просматривая все переходы из состояния q рассматриваем лишь ε-переходы. Пусть существует ε-переход из q в состояние p. Если p помечено как посещенное, то пропускаем его, иначе - помечаем как посещенное, добавляем его в множество Q и кладем в очередь. Если очередь пуста, то мы построили множество Q_ε.

Понятно, что таким образом в множестве Q_ε окажутся все вершины, достижимые по ε-переходам из какого-либо из состояний Q и никакие другие.

Оценим время работы алгоритма. Всего через очередь пройдет $O(|\mathcal{A}|)$ вершин. Для каждой из них мы просмотрим все переходы из нее, что для какой-то одной вершины занимает время $O(|\mathcal{A}|)$. Итого, общее время работы: $O(|\mathcal{A}|^2)$, где $|\mathcal{A}|$ - количество вершин в автомате $\mathcal{A}$.

Пусть мы построили автома $\mathcal{A}$ по заданному регулярному выражению R за $O(|R|)$, причем $|\mathcal{A}| = O(|R|)$. Проверим теперь, принимает ли $\mathcal{A}$ слово w. Для этого будем индуктивно вычислять множества V_i состояний, достижимых из начальных после прочтения префикса длины i слова w.

Посмотрим на множество из всех начальных состояний. Дополним его до множества состояний, достижимых из начальных по ε-переходам, используя описанный выше алгоритм и назовем полученное множество V_0. Далее, пусть построено множество $V_i, i < |w|$. Построим V_{i+1}. Пусть $V_{i+1} = \varnothing$. Пусть $(i+1)$-ая буква слова w есть a. Переберем все элементы множества V_i и для каждого из них просмотрим все переходы, помеченные буквой a. Состояния, в которые ведут эти переходы добавим в V_{i+1}. Далее дополним V_{i+1} состояниями, достижимыми по ε-переходам, используя алгоритм, описанный выше. Понятно, что в множестве V_{i+1} окажутся лишь те состояния, которые достижимы из какого-либо состояния множества V_i после обработки буквы a, а значит там окажутся только те состояния, которые достижимы из какого-либо из начальных состояний после обработки префикса длины $i+1$ слова w.

Получаем, что слово w принимается тогда и только тогда, когда $V_{|w|}$ содержит хотя бы одно финальное состояние.

Оценим время работы итогового алгоритма. Для построения множества V_i необходимо перебрать все переходы, которых $O(|\mathcal{A}|)$, для каждого из элементов V_{i-1}, которых $O(|\mathcal{A}|)$. Для каждого из множеств V_i мы используем поиск в ширину, сложность которого $O(|\mathcal{A}|^2)$. Всего множеств $O(|w|)$. Учитывая время, затраченное на построение $\mathcal{A}$ получаем $O(|R|^3|w|)$.

Задача 1

(i). Пусть задана диаграмма автомата $\mathcal{A}$. Построим по ней леволинейную грамматику G. Аксиомой объявляется новый нетерминал S. Начальное состояние q_0 порождает вывода $Q_0 \to \varepsilon$. Каждое финальное состояние q_i порождает правило $S \to Q_i$. Каждая дуга $(q_{init} \to q_{end})$, помеченная символом a, порождает правило $Q_{end} \to Q_{init}a$.

(ii). Пусть задан автомат $\mathcal{A} = (Q = \{q_0, q_1, \dots\}, \Sigma, \delta, q_0, F)$. Построим по нему леволинейную грамматику $G = (\mathcal{Q} = \{S, Q_1, Q_2, \dots\}, \Sigma, P, S)$, где $P = \{S \to Q_i \text{ для всех } q_i \in F\} \cup \{Q_i \to Q_j a, \text{ если } \delta(q_j, a) = q_i\} \cup \{Q_0 \to \varepsilon\}$.

Доказательство. Поскольку оба представления автомата эквиваленты, при доказательстве будем пользоваться представлением ввиде диаграммы. Заметим, что правило вывода $Q_i \to Q_j a$ входит в G тогда и только тогда, когда в $\mathcal{A}$ существует ребро из состояния q_j в состояние q_i, имеющее метку a.

1. Докажем $L(\mathcal{A}) \subseteq L(G)$. Возьмем произвольное слово $w \in L(\mathcal{A})$. Поскольку $w \in L(\mathcal{A})$, то существует некоторая последовательность состояний автомата $Z = (q_0, \dots, q_{n-1}, q_n)$, причем q_0 - начальное состояние, q_n - одно из финальных состояний. Так же для любой пары последовательных состояний (q_i, q_{i+1}) из Z в $\mathcal{A}$ существует ребро $q_i \to q_{i+1}$ с некоторой пометкой m, причем последовательная конкатенация пометок для каждой пары является словом w.

Построим, используя эту последовательность состояний, вывод слова w в грамматике G. Поскольку q_n - финальное состояние, то существует правило вывода $S \to Q_n$. Применим это правило. Поскольку для любой пары последовательных состояний (q_i, q_{i+1}) из Z в $\mathcal{A}$ существует ребро $q_i \to q_{i+1}$ с некоторой пометкой m, то в G существует правило вывода $Q_{i+1} \to Q_i m$. Применим последовательно правила вывода в следующем порядке:

$Q_n \to Q_{n-1}m_{n-1}$, $Q_{n-1} \to Q_{n-2}m_{n-2}$, $\ldots$, $Q_1 \to Q_0m_0$. Видно, что после этого будет выведено слово вида: Q_0w. Поскольку q_0 - начальное состояние, то существует правило вывода $Q_0 \to \varepsilon$. После применения этого последнего правило будет выведено слово w, а значит $w \in L(G)$.

2. Докажем $L(G) \subseteq L(\mathcal{A})$. Доказательство аналогично изложенному выше. Возьмем произвольное слово $w \in L(D)$. Для него существует вывод в G, состоящий из последовательного применения правил: $S \to Q_n, Q_n \to Q_{n-1}m_{n-1}$, $\ldots$, $Q_1 \to Q_0m_0, Q_0 \to \varepsilon$. Из существования такого вывода следует существования последовательно состояний $(q_0, \ldots, q_n)$, причем при обработке слова w автомат $\mathcal{A}$ пройдет через эту последовательность состояний начиная из начального состояний q_0, достигнет финальной вершины q_n и примет слово w.

3. $L(\mathcal{A}) \subseteq L(G) \wedge L(G) \subseteq L(\mathcal{A}) \Rightarrow L(G) = L(\mathcal{A})$.

Задача 2

(i). Построим леволинейную грамматику $G = (N, T, P, S)$ используя алгоритм, описанный в задаче 1. Множеством нетерминальных символов будет являться множество $N = \{S, Q_0, Q_1, Q_2\}$. Множеством терминальных символов будет являться множество $T = \{0, 1\}$. Множество выводов P получаем, рассмотрев все ребра диаграммы.

$$\begin{gathered} S \to Q_1 \\ Q_0 \to Q_0 0 \\ Q_0 \to Q_1 1 \\ Q_0 \to \varepsilon \\ Q_1 \to Q_0 1 \\ Q_1 \to Q_2 0 \\ Q_2 \to Q_2 1 \\ Q_2 \to Q_1 0 \end{gathered}$$

То, что грамматика - искомая, следует из корректности построения грамматики по автомату, доказанной в задаче 1.

(ii). Используя описанную в теории последовательность действий построим определяющую систему $D(G)$ для грамматики G.

$$\begin{cases} Q_0 = Q_0 0 + Q_1 1 + \varepsilon \\ Q_1 = Q_0 1 + Q_2 0 \\ Q_2 = Q_1 0 + Q_2 1 \end{cases}$$

Для нахождения наименьшей неподвижной точки этой системы потребуется заметить, что наименьшей неподвижной точкой уравнений $X = X\alpha + \beta$ будет $X = \beta\alpha^*$. В этом можно убедится совершенно аналогичным описанному в теории способом. Найдем эту точку.

$$\begin{aligned}
&Q_2 = Q_2 1 + Q_1 0\\
&Q_2 = Q_1 0(1)^* = Q_1(0(1)^*)\\
&Q_1 = Q_0 1 + Q_1(0(1)^* 0)\\
&Q_1 = Q_0(1(0(1)^* 0)^*)\\
&Q_0 = Q_0 0 + Q_0(1(0(1)^* 0)^* 1) + \varepsilon\\
&Q_0 = \varepsilon((1(0(1)^* 0)^* 1) + 0)^* = (1(0(1)^* 0)^* 1 + 0)^*\\
&Q_1 = (1(0(1)^* 0)^* 1 + 0)^* (1(0(1)^* 0)^*)\\
&Q_2 = (1(0(1)^* 0)^* 1 + 0)^* (1(0(1)^* 0)^*)(0(1)^*)
\end{aligned}$$

В последних трех строчках вывода написано решение системы, которое является ее наименьшей неподвижной точкой.

Задача 3

Проверим, что оба языка L_1 и L_2 удовлетворяют сильному варианту ЛР:

$$\exists C\ \forall w \in L_1 : |w| \geq C,\ \exists x, y, z : w = xyz, |xy| \leq C, |y| \geq 0\ \rightarrow\ \forall i \geq 0\ xy^i z \in L_1$$

(i). Язык $L_1 = \{a^n b^{2m}, m, n \geq 0\}$. Возьмем $C = 2$. Рассмотрим произвольное слова $w \in L_1$. Это слово либо содержит ненулевое количество букв a, либо не содержит ни одной. Разберем эти 2 случая.

1. $w = a^n b^{2m} = xyz$. Слово w содержит ненулевое количество букв a $\Rightarrow n \geq 1$. Возьмем $x = \varepsilon$, $y = a$, $z = a^{n-1} b^{2m}$. Видно, что $|xy| = 1 \leq C = 2$, $|y| = 1 > 0$. Заметим, что $\forall i \geq 0\ xy^i z = \varepsilon a^i a^{n-1} b^{2m} = a^{n-1+i} b^{2m} \in L_1$. Поскольку $n \geq 1$, то $n - 1 + i \geq 0$, а значит $\forall i \geq 0\ xy^i z = a^{n-1+i} b^{2m} \in L_1$.

2. $w = b^{2m} = xyz$. $|w| \geq C = 2 \Rightarrow m \geq 1$. Возьмем: $x = \varepsilon$, $y = b^2$, $z = b^{2m-2}$. Видно, что $|xy| = 2 \leq C = 2$, $|y| = 2 > 0$. Заметим, что $\forall i \geq 0\ xy^i z = \varepsilon b^{2i} b^{2m-2} = b^{2m-2+2i}$. Поскольку $m \geq 1$, то $2m - 2 + 2i \geq 0$, а значит $\forall i \geq 0\ xy^i z = b^{2m-2+2i} \in L_1$.

Получаем, что в обоих случаях ЛР выполняется.

(ii). Язык $L_2 = \{w \in \{a, b\}^* | a^k b^{2^n}, k, n \geq 0\} \cup \{b^i, i \geq 0\} = T_1 \cup T_2$. Возьмем $C = 1$. Рассмотрим произвольное слово $w \in L_2$. Это слово либо содержит ненулевое количество букв a, либо не содержит ни одной. Разберем эти 2 случая.

1. Поскольку w содержит a, то $w \in T_1$, а значит $w = a^k b^{2^n} = xyz$, причем $k \geq 1$. Возьмем: $x = \varepsilon$, $y = a$, $z = a^{k-1} b^{2^n}$. Видно, что $|xy| = 1 \leq C = 1$, $|y| = 1 > 0$. Заметим, что $\forall i \geq 0\ xy^i z = \varepsilon a^i a^{k-1} b^{2^n} = a^{k-1+i} b^{2^n}$. Поскольку $k \geq 1$, то $k - 1 + i \geq 0$, а значит $\forall i \geq 0\ xy^i z = a^{k-1+i} b^{2^n} \in T_1 \subseteq L_2$.

2. $w = b^n = xyz$. $|w| \geq C = 1 \Rightarrow n \geq 1$. Возьмем: $x = \varepsilon$, $y = b$, $z = b^{n-1}$. Видно, что $|xy| = 1 \leq C = 1$, $|y| = 1 > 0$. Заметим, что $\forall i \geq 0\ xy^i z = \varepsilon b^i b^{n-1} = b^{n-1+i}$. Поскольку $n \geq 1$, то $n - 1 + i \geq 0$, а значит $\forall i \geq 0\ xy^i z = b^{n-1+i} \in T_2 \subseteq L_2$.

Получаем, что в обоих случаях ЛР выполняется.

Задача 4

Заметим, что $L_2 = (a^*b^*c^*)^* = \{a,b,c\}^* = \{a,b,c\}^* \cdot \{\varepsilon\}$, а значит L_2 - простой.

Докажем, что язык $L_1 = (a^*b^*c)^*$ - не простой. Допустим, что он простой, то есть $\exists$ конечные $S_1, S_2 : \ L_1 = S_2^* S_1$. Множество слов $a^*c \subset L_1$, а значит $a^*c \subset S_2^* S_1$. Поскольку множество слов a^*c бесконечное, а множества S_1 и S_2 конечные, то $\exists n \geq 1 : \ a^n \in S_2$. Аналогично бесконечное множество слов $b^*c \subset L_1$, а значит $\exists m \geq 1 : \ b^m \in S_2$. Получаем, что существует слово w с префиксом $b^m a^n$, $w \in S_2^* S_1 = L_1$. Слово w имеет подслово ba, но $L_1 = (a^*b^*c)^*$ не может содержать слово с подсловом ba, поскольку между стоящей где-то буквой b и стоящий после нее буквой a должна стоять буква c. Получаем противоречие, значит L_1 - не простой.

Докажем, что язык $L_3 = (a^* + b^*c)^*$ - не простой. Допустим, что он простой, то есть $\exists$ конечные $S_1, S_2 : \ L_3 = S_2^* S_1$. Множество слов $a^* \subset L_3$, а значит $a^* \subset S_2^* S_1$. Поскольку множество слов a^* бесконечное, а множества S_1 и S_2 конечные, то $\exists n \geq 1 : \ a^n \in S_2$. Аналогично бесконечное множество слов $b^*c \subset L_1$, а значит $\exists m \geq 1 : \ b^m \in S_2$. Получаем, что существует слово w с префиксом $b^m a^n$, $w \in S_2^* S_1 = L_3$. Слово w имеет подслово ba, но $L_3 = (a^* + b^*c)^*$ не может содержать слово с подсловом ba, поскольку между стоящей где-то буквой b и стоящий после нее буквой a должна стоять буква c. Получаем противоречие, значит L_3 - не простой.