Лекция 10

Теория формальных языков (II)

(Конспект: К. Ефремова)

10.1 Конечные автоматы (продолжение)

Определение 10.1. *Детерминированный конечный автомат* — это недетерминированный конечный автомат, у которого $\delta(q, \epsilon) = \emptyset$ и $|\delta(q, a)| \leqslant 1$. *Полностью определенным* детерминированным конечным автоматом называется автомат, у которого $|\delta(q, a)| = 1$. (Иначе говоря, можно считать, что у него $\delta : Q \times \Sigma \to Q$.)

Определение 10.2. Обозначение: $q_1 \xrightarrow{a} q_2$ означает $q_2 \in \delta(q_1, a)$.

Теорема 10.1. *Для любого недетерминированного конечного автомата $\mathcal{A} = (Q, \Sigma, q_s, F, \delta)$ можно построить полностью определенный детерминированный конечный автомат $\mathcal{A}' = (Q', \Sigma, q'_s, F', \delta')$, такой, что $L(\mathcal{A}) = L(\mathcal{A}')$.*

Доказательство. 1. Избавимся от ϵ-переходов. Найдем все пары состояний (q_1, q_2), такие, что q_2 достижимо из q_1 по ϵ-переходам: $q_1 \xrightarrow{\epsilon} q' \xrightarrow{\epsilon} \ldots \xrightarrow{\epsilon} q_2$. Для каждого *не*-$\epsilon$-перехода $q_2 \xrightarrow{a} q_3$ добавим переход $q_1 \xrightarrow{a} q_3$. Кроме того, если $q_2 \in F$, добавим и q_1 в F.

После этого удалим все ϵ-переходы. Заметим, что язык при этом не изменится. Действительно, раньше мы шли по пути $q_1 \to \ldots \to q_2 \to q_3$, считывая a, а теперь по другому пути, но все равно считываем a; и наоборот.

2. Пусть $Q' = 2^Q$; $q'_s = \{q_s\}$; $F' = \{q' \in Q' \mid q' \cap F \neq \emptyset\}$. Напишем новые переходы вида $q'_1 \xrightarrow{a} q'_2$, где $q'_2 = \{q \in Q \mid \exists r \in q'_1 : r \xrightarrow{a} q\}$. Построенный новый автомат, очевидно, детерминированный.

Докажем, что язык не меняется.

(a) Пусть старый автомат принимает строчку. Покажем,что новый автомат тоже примет эту строчку. Раньше мы совершали переход $q_s \xrightarrow{a_1} q_1$, теперь — переход $\{q_s\} \xrightarrow{a_1} \{q_1, \ldots\}$. Последующие переходы $q_{i-1} \xrightarrow{a_i} q_i$ превращаются в $\{q_{i-1}, \ldots\} \xrightarrow{a_i} \{q_i, \ldots\}$. Последнее состояние будет принадлежать F', ибо оно содержит конечное состояние старого автомата.

(b) Пусть строчка принимается новым автоматом. По определению, в состоянии, в котором он завершает работу, содержится некоторое конечное состояние старого автомата. Пусть последний переход нового автомата — $q' \xrightarrow{a} \{p, \ldots\}$ (где $p \in F$). По определению новой функции перехода, существует $q \in q'$, такое, что $q \xrightarrow{a} p$ в старом автомате. Далее рассмотрим предыдущий переход $r' \xrightarrow{b} \{q, \ldots\}$ в новом автомате, и т. д., пока не дойдем до стартового состояния. □

Теорема 10.2. *Множества языков, задаваемых*

(1) конечными автоматами,

(2) регулярными выражениями,

(3) праволинейными грамматиками

в одном и том же алфавите Σ, совпадают.

Доказательство. (2)⇒(1). Доказательство по индукции.
База:

1. Пустой язык. Автомат, принимающий пустой язык — автомат с пустым F.

2. Язык $\{\epsilon\}$. В автомате, принимающем его, $F = \{q_s\}$, а переходов в нем нет.

3. Язык $\{a\}$. Единственный переход такого автомата: $q_s \xrightarrow{a} q$; $F = \{q\}$.

Индукционный переход:

1. Рассмотрим конкатенацию языков L_1 и L_2. Пусть есть автомат, принимающий язык L_1; и другой автомат, принимающий L_2. Мы хотим построить автомат,принимающий $L_1 \cdot L_2$.

   Объединим эти два автомата в один, и слегка модифицируем его. Новым стартовым состоянием будет стартовое состояние первого автомата. Новым множеством конечных состояний будет множество конечных состояний второго автомата. Также добавим ϵ-переходы из (бывших) конечных состояний первого автомата в (бывшее) начальное состояние второго автомата.

2. $L_1 \cup L_2$. Снова объединим два автомата. Добавим новое стартовое состояние, а из него — ϵ-переходы в два старых стартовых состояния. На сей раз конечными состояниями построенного автомата будут конечные состояния обоих автоматов.

3. L^*. Добавим ϵ-переходы из конечных состояний — в начальное, а также добавим ϵ-переход из стартового состояния в *новое* (дополнительное) конечное состояние.

(1)⇒(3). Нетерминалы грамматики — состояния автомата. Если $q \xrightarrow{a} p$, то добавим правило $q \to ap$. Если еще и $p \in F$, то добавим еще правило $q \to a$.

(3)⇒(2). Можно построить систему уравнений

$$\begin{array}{rcl} A_1 & = & \alpha_{10} \cup (\alpha_{11}A_1) \cup (\alpha_{12}A_2) \ldots \cup (\alpha_{1n}A_n) \\ A_2 & = & \alpha_{20} \cup (\alpha_{21}A_1) \cup (\alpha_{22}A_2) \ldots \cup (\alpha_{2n}A_n) \\ & \ldots & \\ A_n & = & \alpha_{n0} \cup (\alpha_{n1}A_1) \cup (\alpha_{n2}A_2) \ldots \cup (\alpha_{nn}A_n), \end{array}$$

где $A_1, A_2, \ldots, A_n$ — все нетерминалы, они же неизвестные этой системы уравнений; α_{ij} — регулярные выражения в алфавите Σ, коэффициенты системы, строятся они следующим образом. Посмотрим все правила для нетерминала A_i, у которых в правой части A_j; пусть это $A_i \to a_1A_j, \ldots, A_i \to a_kA_j$. Тогда $\alpha_{ij} := \{a_1\} \cup \{a_2\} \cup \ldots \cup \{a_k\}$. (Если $j = 0$, рассматриваем правила, не содержащие нетерминала в правой части.)

Исключим A_1: перепишем первое неравенство в виде $A_1 = (\alpha_{11}A_1) \cup \beta$; теперь $A_1 = \alpha_{11}^*\beta$ (где $\beta \in (\Sigma \cup N)^*$) подставляем везде в последующие уравнения системы. Аналогично поступаем с $A_2, \ldots, A_n$, находя A_i из i-го уравнения. Найдя, таким образом, регулярное выражение (уже в алфавите Σ, а не $\Sigma \cup N$) для A_n, подставим его в предыдущие уравнения, и т. д. В результате будет получено решение, которое будет минимальным[1], т.е. не будет содержать других решений; значения для нетерминалов будут, очевидно, будет регулярными выражениями. Значение для стартового нетерминала — искомое[2] регулярное выражение. □

Задача 10.1. Завершить доказательство теоремы 10.2: доказать, что найденное так «решение» действительно будет минимальным решением, а также, что минимальное решение (точнее, соответствующее выражение для стартового нетерминала) действительно является языком, порождаемым данной грамматикой. □

[1] См. задачу 10.1.
[2] См. задачу 10.1.

Лемма 10.1 (лемма о разрастании для регулярных языков (pumping lemma)). *Пусть L — регулярный язык. Тогда существует константа c, такая, что любую строку $x \in L$ длины не менее c можно разбить на три части $x = u \cdot v \cdot w$, такие что $0 < |v| \leqslant c$ и $\forall i \geqslant 0$ $u \cdot v^i \cdot w \in L$.*

Доказательство. Рассмотрим детерминированный конечный полностью определенный автомат для языка L. Пусть $c = |Q| + 1$. Посмотрим, как он работает на цепочке x: $q_s \to q_1 \to \ldots \to q_k \in F$. На каждом шаге считывается некоторый символ. Поскольку $|x| \geqslant c$, мы должны были побывать в каком-то состоянии дважды, и в нашем пути есть циклы. Выберем несамопересекающийся цикл; пусть до первого прохождения по нему считывалась строка u, при прохождении по нему считывалась строка v (ее длина не превосходит c, поскольку цикл — несамопересекающийся), а после прохождения по нему (в том числе, если по нему пошли еще раз) — строка w. Очевидно, наш автомат примет любую строку вида uv^iw. □

С помощью этой леммы можно доказывать, что какой-нибудь язык не является регулярным.

Пример 10.1. $L = \{0^n1^n | n \in \mathbb{N} \cup 0\}$ не является регулярным.
Доказательство. Пусть регулярный. Рассмотрим варианты подстроки v из леммы.

1. В строку попадают только нули $\Rightarrow$ в uv^2w количество 0 увеличится, а количество 1 останется неизменным $\Rightarrow$ строка не будет принадлежать L.

2. В строку попадают только единицы — аналогично.

3. $v = 0^i1^j \Rightarrow$ в uv^2w после 1 будет идти 0 $\Rightarrow$ строка опять не будет принадлежать языку.

(Замечание: тем не менее, этот язык может быть порожден *бесконтекстной* грамматикой $S \to 0S1$, $S \to \epsilon$.) □

Полезные свойства регулярных языков:

1. Множество всех регулярных языков в данном алфавите замкнуто относительно операций, которые их порождают: $\cup$, $\cdot$, *.

2.

   Лемма 10.2. *Множество всех регулярных языков в данном алфавите замкнуто относительно дополнения.*

Доказательство. Рассмотрим детерминированный конечный полностью определенный автомат, задающий язык L. Поменяем местами его конечные состояния с остальными: $F' := Q \setminus F$. Полученный автомат задает язык $\overline{L}$. □

3.

Следствие 10.1. *…и замкнуто относительно пересечения.*

Разрешимые проблемы, связанные с регулярными языками:

1. Принадлежность. Рассмотрим детерминированный конечный полностью определенный автомат, принимающий данный язык. Чтобы узнать, принадлежит ли этому языку некоторая строка, запустим наш автомат на этой строке x. После $|x|$ переходов мы поймем, принадлежит ли она языку.

2. Пустота языка. Чтобы решить эту проблему, достаточно проверить, достижимо ли какое-нибудь конечное состояние автомата из начального. Эта задача, очевидно, алгоритмически разрешима.

3. Равенство языков. Чтобы построить алгоритм для этой задачи, достаточно заметить, что

   $$L_1 = L_2 \Leftrightarrow L_1 \cap \overline{L_2} = \emptyset \wedge L_2 \cap \overline{L_1} = \emptyset \text{ (или: } (L_1 \cap \overline{L_2}) \cup (L_2 \cap \overline{L_1}) = \emptyset).$$

4. Включение языков. Чтобы построить алгоритм для этой задачи, достаточно заметить, что

   $$L_1 \subseteq L_2 \Leftrightarrow (L_1 \cap \overline{L_2} = \emptyset) \text{ (или: } (L_1 \cap L_2 = L_1)).$$