Некоторые вопросы, рассмотренные на практических занятиях

1 Лемма о накачке

Теорема 1 («Лемма о накачке» / «Лемма о разрастании»). *Пусть L — регулярный язык. Тогда существует такая константа $n \in \mathbb{N}$, что для любого слова $w \in L$, такого что $|w| \geqslant n$, существует такое разбиение $w = xyz$ слова w, что:*

(1) $y \neq \varepsilon$;

(2) $|xy| \leqslant n$;

(3) $\{xy^k z \mid k \geqslant 0\} \subset L$.

Доказательство. Так как L регулярный язык, то существует детерминированный конечный автомат $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ с $|Q| = N$ состояниями, распознающий язык L. Пусть $w \in L$ и $|w| = N + 1$. Подадим на вход автомату $\mathcal{A}$ слово w. Очевидно, существует состояние $q \in Q$, в котором автомат окажется дважды, читая это слово (принцип Дирихле / принцип голубятни). Разобьём слово w на три части $w = xyz$, так что:

$$(q_0, xyz) \vdash^* (q, yz) \overset{\triangle}{\vdash^*} (q, z) \vdash^* (q_F, \varepsilon),$$

где $q_F \in F$. Покажем, что для любого целого $k \geqslant 0$ автомат распознает слово $xy^k z$. Действительно, последовательность переходов при чтении цепочек x и z остаётся такой же, как для слова w. Часть y^k читается k-кратным повтором последовательности переходов $\triangle$. Таким образом, $\{xy^k z \mid k \geqslant 0\} \subset L$ и выполнено условие (3).
Части x, y слова w удовлетворяют условиям (1)–(2) по построению. Полагая $n = N + 1$, получаем выполненными все условия теоремы. ■

2 Удаление бесполезных символов

Определение. Пусть дана КС-грамматика $G = (\Sigma, N, \mathcal{P}, S \in N)$. Говорят, что символ $X \in \Sigma \cup N$

(1) *полезный в* G, если

$$\exists \alpha, \beta \in (\Sigma \cup N)^* \, \exists w \in \Sigma^* \colon \quad S \Rightarrow_G^* \alpha X \beta \Rightarrow_G^* w.$$

(2) *порождающий в* G, если

$$\exists w \in \Sigma^* \colon \quad X \Rightarrow_G^* w$$

(3) *достижимый в G*, если

$$\exists \alpha, \beta \in (\Sigma \cup N)^*: \quad S \Rightarrow_G^* \alpha X \beta$$

Бесполезным называется любой символ, не являющийся полезным.

Теорема 1. *Если к КС-грамматике G, такой что $L(G) \neq \varnothing$,* последовательно *применить два преобразования:*

(1) удалить символы, не являющиеся порождающими,

(2) удалить символы, не являющиеся достижимыми,

то будет получена грамматика, не содержащая бесполезных символов.

Доказательство. Обозначим грамматику, получившуюся после первого шага, $G_1 = (N_1, \Sigma_1, \mathcal{P}_1, S)$ и грамматику, получившуюся после второго шага, $G_2 = (N_2, \Sigma_2, \mathcal{P}_2, S)$ (символ S не будет удалён из грамматики после первого шага благодаря требованию $L(G) \neq \varnothing$ и после второго шага — по определению достижимого символа). Пусть $X \in N_2 \cup \Sigma_2$. Покажем, что X — полезный в G_2 символ. Для этого необходимо найти $\alpha, \beta \in (N_2 \cup \Sigma_2)^*$ и $w' \in \Sigma_2^*$, такие что

$$S \Rightarrow_{G_2}^* \alpha X \beta \Rightarrow_{G_2}^* w'. \tag{1}$$

Так как X не был удалён после первого шага, то он является порождающим в исходной грамматике G, то есть существует вывод $X \Rightarrow_G^* w$ для некоторого $w \in \Sigma^*$. Очевидно, что каждый символ этого вывода рано или поздно переходит в один из символов w или в ε, то есть также является порождающим в G, а значит, попадает в G_1. Потому этот вывод можно целиком перенести в G_1:

$$X \Rightarrow_{G_1}^* w, \tag{2}$$

где $w \in \Sigma_1^*$.

Так как X не был удалён после второго шага, то он достижим в грамматике G_1, то есть существует вывод $S \Rightarrow_{G_1}^* \alpha X \beta$ для некоторых $\alpha, \beta \in (N_1 \cup \Sigma_1)^*$. Очевидно, что каждый символ в этом выводе получен из S применением правил грамматики G_1, то есть является достижимым в G_1, а значит, попадает в G_2. Потому этот вывод можно целиком перенести в G_2:

$$S \Rightarrow_{G_2}^* \alpha X \beta, \tag{3}$$

где $\alpha, \beta \in (N_2 \cup \Sigma_2)^*$.

Так как X достижим в G_1, то и каждый символ вывода (2) достижим в G_1, а значит, этот вывод целиком переносится в G_2:

$$X \Rightarrow_{G_2}^* w, \tag{4}$$

где $w \in \Sigma_2^*$.

Так как $\alpha, \beta \in (N_2 \cup \Sigma_2)^*$, то к цепочкам α и β применимы те же рассуждения, что к символу X, то есть можно получить результат, аналогичный (4):

$$\exists u \in \Sigma_2^*\colon \alpha \Rightarrow_{G_2}^* u, \tag{5}$$

$$\exists v \in \Sigma_2^*\colon \beta \Rightarrow_{G_2}^* v. \tag{6}$$

Объединяя результаты (3)–(6), получаем:

$$S \Rightarrow_{G_2}^* \alpha X \beta \Rightarrow_{G_2}^* uwv,$$

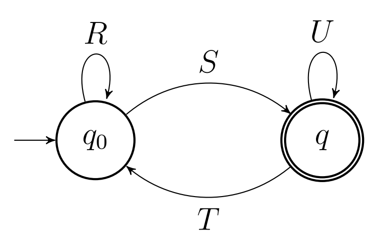
то есть выполнено (1). ■

3 Нахождение языка конечного автомата методом исключения состояний

Метод исключения состояний подразумевает последовательное удаление вершин графа переходов автомата, которое протоколируется с помощью записи на оставшихся дугах регулярных выражений (можно считать, что в изначальном графе на дугах простейшие регулярные выражения — однобуквенные или пустые множества для отсутствующих дуг).
Процедура исключения состояния s: для каждых двух (необязательно различных, но несовпадающих с s) состояний p и q, таких что существует фрагмент графа переходов автомата $p \xrightarrow{R_1} s \xrightarrow{R_2} q$, где R_1, R_2 — некоторые регулярные выражения (метки дуг переходов), прибавить к метке дуги $p \xrightarrow{R_3} q$ выражение $R_1 R^* R_2$, где R это метка петли на вершине s (если петля на s и/или дуга $p \to q$ отсутствовали в исходном графе, то можно считать, что их метки равны $\varnothing$) — таким образом получена дуга с меткой: $p \xrightarrow{R_3 + R_1 R^* R_2} q$. Удалить все просмотренные дуги $p \to s$ и $s \to q$, инцидентные вершине s. После этого s стала изолированной или (неориентированно) висячей и её можно удалить (с входящими или выходящими из неё дугами, если таковые имеются).

Алгоритм (нахождение языка автомата методом исключения состояний)**.**
Вход: Конечный автомат $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, заданный графом переходов.
Выход: Регулярное выражение, описывающее язык автомата.

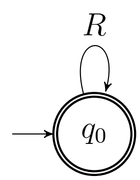
Шаг 1. Для каждого финального состояния $q \in F$, отличного от начального q_0, применять процедуру исключения состояний до тех пор, пока не останутся две вершины: q_0 и q. В результате получится подобный автомат:



Допускаемый им язык описывается так:

$$(R + SU^*T)^*SU^*.$$

Шаг 2. Если начальное состояние q_0 является финальным ($q_0 \in F$), применять процедуру исключения состояний, пока не останется единственная вершина q_0. В результате получится подобный автомат:



Допускаемый им язык описывается так: R^*.

Шаг 3. Язык исходного автомата определяется как сумма всех регулярных выражений, полученных на шагах (1)–(2).

4 Свойства замкнутости класса регулярных языков

Теорема 1. *Класс регулярных языков замкнут относительно операций: объединения, конкатенации, итерации, дополнения, пересечения, разности.*

Доказательство. Для регулярных языков L_1, L_2 рассмотрим описывающие их регулярные выражения $\mathrm{RE}(L_1)$, $\mathrm{RE}(L_2)$. Язык объединения (соответственно, конкатенации, итерации) описывается выражением $\mathrm{RE}(L_1)+\mathrm{RE}(L_2)$ (соответственно, $\mathrm{RE}(L_1)\,\mathrm{RE}(L_2)$, $\mathrm{RE}(L_1)^*$), а значит, регулярен.

Для регулярного языка L построим распознающий его детерминированный конечный автомат $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ ($L = L(\mathcal{A})$). Легко видеть, что автомат $\widetilde{\mathcal{A}} = (Q, \Sigma, \delta, q_0, Q \setminus F)$ допускает дополнение $\overline{L} = \Sigma^* \setminus L$ языка L, которое является, таким образом, регулярным языком.

Поскольку справедливо $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$, язык $L_1 \cap L_2$ регулярен по доказанному выше. Аналогичное можно заключить из равенства $L_1 \setminus L_2 = L_1 \cap \overline{L_2}$. ■