Задание 3

НКА и ДКА
Лемма о накачке
Автоматы и распознавание текстов

Ключевые слова [1]:*принцип мат. индукции, язык, регулярные выражения, конкатенация, объединение, итерация, конечные автоматы (КА), детерминированные и недетерминированные КА, регулярные языки. алгебра регулярных выражений, примеры нерегулярных языков; поиск подстрок, алгоритм Кнута-Морриса-Пратта.*

1 НКА и ДКА

Из алгоритма детерминизации НКА следует, что если НКА $\mathcal{A}$ имеет множество состояний $Q_{\mathcal{A}}$, то построенный по нему ДКА $\mathcal{B}$ имеет множество макросостояний $Q_{\mathcal{B}} \subseteq 2^{Q_{\mathcal{A}}}$, где $2^{Q_{\mathcal{A}}}$ – множество всех подмножеств множества $Q_{\mathcal{A}}$. Таким образом, на число состояний автомата $\mathcal{B}$ мы имеем верхнюю оценку $|Q_{\mathcal{B}}| \leqslant 2^{|Q_{\mathcal{A}}|}$. То есть число состояний ДКА ограниченно экспоненциальной функцией от числа состояний НКА, но существует ли язык, для которого эта оценка достигается? На самом деле, когда мы говорим об оценках такого рода, нам требуется рассматривать ни один какой-то язык, а последовательность языков, по которым мы и сможем установить экспоненциальную зависимость.

Задача 1. Определим язык $L_i = \{w \,|\, |w| = n, w[n-i] = 1\}$, то есть в язык L_i входят все слова, в которых 1 стоит на i-ом месте от конца[2]. Постройте НКА, распознающий язык L_3. По построенному НКА постройте ДКА.

Задача 2*. Докажите, что на языках L_i между НКА и построенными по ним ДКА достигается экспоненциальный разрыв.

Упражнение 1. Почему для доказательства экспоненциального разрыва необходима бесконечная последовательность языков, а не достаточно конечной?

[1] минимальный необходимый объём понятий и навыков по этому разделу)

[2] Во избежании путаницы, первый с конца символ – это последний символ слова.

2 Лемма о накачке[3]

В данном разделе мы поговорим о лемме о накачке – одном из способов доказательства нерегулярности языка.

Лемма 1. *Для любого регулярного языка L существует такая константа $p \geqslant 1$, что для любого слова w из L длиннее p, справедливо:*

- $w = xyz$
- $|y| \geqslant 1$
- $|xy| \leqslant p$
- $\forall i \geqslant 0, xy^iz \in L$.

Доказательство. Поскольку $L \in \mathsf{REG}$, то существует ДКА $\mathcal{A}$ распознающий L. Пусть $\mathcal{A}$ имеет N состояний. Возьмём $p = N + 1$. Тогда, если если слово w принадлежит L и $|w| \geqslant p$, то это означает, что при обработке w автомат $\mathcal{A}$ оказался в некотором сотоянии q дважды. Пусть в первый раз автомат оказался в q после прочтения префикса x, а второй раз, при прочтении префикса xy. Тогда $\delta(q, y) = q$, но поскольку $w = xyz$ принадлежит L, то это означает, что $\delta(q, z) = q_f \in F$, а значит все слова вида xy^iz, $i \geqslant 0$ лежат в L. □

Обратите внимание, что при доказательстве леммы, я использовал те же трюки, что и в доказательстве на семинаре того, что a^nb^n – нерегуляный язык. Также обратите внимание на то, что формула $w = xyz$ означает, что для слова w существует такое разбиение xyz, для которого выполняются следующие свойства и утверждение леммы: часто студенты это равенство ошибочно воспринимают как «для любого разбиения».

Пример 1. Используем лемму о накачке для доказательства христоматийного примера нерегулярности языка $L = \{0^n1^n \mid n \geqslant 0\}$.

Доказательство. Допустим, что язык L регулярный. Тогда, по лемме о накачке, существует константа p, что для любого слова w длиннее p, существует такое разбиение xyz, что $|xy| \leqslant p$ и слова xy^iz, $i \geqslant 0$ принадлежат L.

[3] Также известна как лемма о разрастании. Неудачные переводы неудачного термина «Pumping Lemma».

Рассмотрим $w = 0^p 1^p$. Если такое разбиение существует, то y имеет вид 0^k или 1^k, $k \geqslant 1$ – в противном случае, если $y = 0^k 1^l$, то $y^2 = 0^k 1^l 0^k 1^l$, но в L нет слов, в которых за 1 следует 0. Допустим, что $y = 0^k$. Тогда $x = 0^m$, $z = 0^l 1^p$, $k + m + l = p$. Но тогда, по лемме о накачке $xy^2z \in L$, а значит, слово $0^{m+2k+l}1^l \in L$, но $m + 2k + l > p$, т.к. $m + k + l = p$ и $k > 0$, поскольку $|y| \geqslant 1$. Аналогично приходим к противоречию когда $y = 1^k$. □

Замечание 1. *Для доказательства того, что некоторый язык является нерегулярным, при использовании леммы о накачке необходимо предъявить не одно конкретное слово, а последовательность слов, зависящих от p (константы леммы). Если вы предъявите всего одно слово, то кто-то очень умный сможет просто взять бóльшую константу.*

Упражнение 2. В предыдущем примере показано громоздкое доказательство: его можно сделать проще, убрав перебор случаев, воспользовавшись структурой слова. Постарайтесь получить доказательство в одну строчку или восстановить доказательство с семинара.

У этой леммы слишком много недостатков. Во-первых, она не всегда применима: утверждение леммы справедливо для некоторых нерегулярных языков. Во-вторых, неаккуратное применение леммы влечёт громоздкие выкладки. Даже для такого простого примера как $L = \{0^n 1^n \mid n \geqslant 0\}$, неподкованному в этой науке человеку потребуется применить много усилий, а в более сложных случаев перебор возможных y может оказаться ещё более громоздким. Фактически лемма о накачке работает в тех же случаях, когда срабатывает техника, обсуждённая на семинаре, поэтому некоторым возможно будет удобно применять эту технику напрямую. Тем не менее, у этой леммы есть очень важный плюс – учебный. Во-первых, лемма о накачке показывает структуру регулярного языка: разность длин двух последовательных слов из регулярного языка ограниченна линейной функцией. Во-вторых, существует ещё лемма о накачке для контекстно-свободных языков (которые мы изучим позже), для понимания которой стоит изучить более простую лемму о накачке для регулярных языков. В случае КС-языков, доказательство непринадлежности языка классу КС уже куда менее очевидно, так что лемма о накачке становится мощным и одним из основных инструментов.

Задача 3. Будут ли регулярными следующие языки?

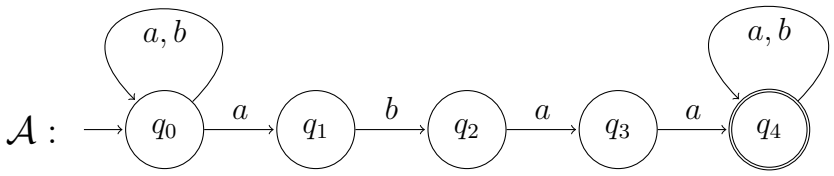
1. $L = \{a^{2013n+5} \mid n = 0, 1, \} \cap \{a^{503k+29} \mid k = 401, 402, \ldots\} \subseteq \{a^*\}$.

2. $L_2 = \{a^{200n^2+1} \mid n = 1000, 1001, \ldots\} \subseteq \{a^*\}$.

3. Язык L_3 всех слов в алфавите $\{0, 1\}$, которые представляют числа в двоичной записи, дающие остаток два при делении на три (слово читается со старших разрядов). Например, $001010\,(1010_2 = 10_{10} = 3{\times}3{+}1) \notin L_3$, а $10001\,(10001_2 = 17_{10} = 5 \times 3 + 2) \in L_3$.

3 Алгоритм Кнута-Морриса-Пратта и его связь с автоматами

НКА – очень удобный инструмент для описания автоматов, которые ищут слова в тексте. Например, автомат



проверяет имеет ли поданное на вход слово подслово $abaa$.

Задача 4. Постройте по НКА $\mathcal{A}$ ДКА $\mathcal{B}$.

Как мы уже обсуждали, для алгоритмической проверки принадлежности слова языку, распознаваемому НКА, по нему следует строить ДКА. Однако, в специальных случаях, используемых на практике, подобно описанному выше, есть более удобные алгоритмы и один из них – алгоритм Кнута-Морриса-Пратта. Этот алгоритм подробно описан в 10-ой главе книги А. Шеня «Программирование. Теоремы и задачи». Её можно в свободном доступе скачать здесь. Я рекомендую изучить КМП-алгоритм по этой книге, в этом разделе я лишь скажу пару слов о его связи с автоматами, а точнее дам на эту тему пару задач.

В основе этого алгоритма – использование для поиска слова вычисления префикс-функции.

Определение 1. Назовём префикс-функцией функцию $l()$, которая возвращает самый длинный собственный[4] префикс слова w, являющийся одновременно его суффиксом.

Пример 2. Приведём пример вычисления префикс-функции.

$$l(a^{n+1}) = a^n$$
$$l(ababa) = aba$$
$$l(abb) = \varepsilon$$

У префикс функции есть важное свойство – все собственные префиксы слова w, которые являются его суффиксами лежат в последовательности $l(w), l(l(w)), \ldots$

Задача 5*. Докажите, что в ДКА, распознающем язык $\Sigma^* w \Sigma^*$ не может быть меньше состояний чем элементов последовательности $l(w), l(l(w)), \ldots$

Зафиксируем слово w. Напомним, что $w[i, j]$ – это подслово $w_i w_{i+1} \ldots w_j$ слова $w = w_1 w_2 \ldots w_n$, здесь всюду $w_k \in \Sigma$. Для удобства будем считать, что $w[0, 0] = \varepsilon$, а $w[0, k] = w[1, k]$ при $k > 0$. Определим автомат, который будем называть *автоматом Кнута-Морриса-Пратта* или КМП-автоматом для слова w.

Определение 2. КМП автоматом $\mathcal{A}_w$ для слова $w \in \Sigma$ длины n, называется автомат, который задан набром $(Q,\ \Sigma,\ q_0,\ \delta,\ F)$, где

- $Q = \{\ w[0,0],\ w[0,1],\ w[0,2], \ldots,\ w[0,n]\ \}$;
- $q_0 = w[0,0]$;
- $\delta(w[0,k], a) = \begin{cases} w[0,k+1], & \text{при } w[0,k+1] = w[0,k]a \text{ и } k < n; \\ l(w[0,k]a), & \text{при } w[0,k+1] \neq w[0,k]a \text{ и } k < n; \\ w[0,n], & \text{при } k = n. \end{cases}$
- $F = \{w[0,n]\}$.

[4]То есть префикс, не совпадающий со всем словом w.

Замечание 2. *В качестве множества состояний КМП-автомата выступает множество слов, поэтому применение конкатенации при определении функции переходов корректно и осмысленно.*

Задача 6. Постройте КМП-автомат для слова *abaa*.

Задача 7*. Докажите, что КМП-автомат для слова w распознаёт язык $\Sigma^* w \Sigma^*$.

4 Дополнительные задачи

Задача 8. Приведите протокол работы КМП-алгоритма при поиске подслова *abba* в слове *abbbababbab*.