Задание 2
НКА и алгоритмы поиска подстрок

Литература:

1. *Хопкрофт Д., Мотвани Р., Ульман Д.*
   Введение в теорию автоматов, языков и вычислений.
   М.: Вильямс, 2002.

2. *Ахо А., Ульман Д.*
   Теория синтаксического анализа, перевода и компиляции
   М.: Мир, 1978. Гл. 0, 2.

3. *Серебряков В.А., Галочкин М.П., Гончар Д.Р., Фуругян М.Г.*
   Теория и реализация языков программирования.
   М.: МЗ-пресс, 2006.

4. *Шень. А. Х.*
   Программирование: теоремы и задачи
   М.: МЦНМО, 2004.

5. *Журавлёв Ю.И., Флёров Ю.А, Вялый М.Н.*
   Дискретный Анализ. Формальные системы и алгоритмы.
   М.: МЗ-пресс, 2010.

Ключевые слова [1]:*язык, регулярные выражения, конкатенация, объединение, итерация, конечные автоматы (КА), детерминированные и недетерминированные КА, регулярные языки.*

Упражнения из этого задания вовсе не обязательно техать. Задачи, помеченные звёздочкой указывают на трудность задачи, но не переводят их в разряд необязательных. Стоит хотя бы попытаться их решить.

1 Построение по регулярному выражению конечного автомата

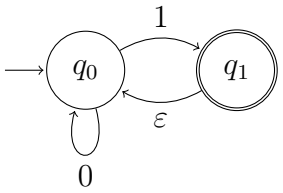
На семинаре мы разобрали алгоритм построения детерминированного конечного автомата по регулярному выражению. Однако, его нельзя на-

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

звать простым. Построить недетерминированный автомат по регулярному выражению гораздо проще. В каком-то смысле, если вы имеете дело с регулярным выражением, вы имеете дело с НКА.

Напомним, что помимо обычных переходов недетерминированные автоматы, имеют также ε-переходы, т.е. переходы вида $\delta(q_i, \varepsilon) = q_j$. Наличие таких переходов означает, что попав в состояние q_i, автомат может перейти в состояние q_j не обрабатывая следующий символ слова.

Пример 1.



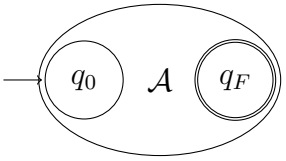
Легко видеть, что данный автомат принимает язык, состоящий из слов, оканчивающихся на 1. При прочтении 1, автомат переходит из состояния q_0 в q_1, дальше, если во входном слове ещё остались необработанные символы, автомат делает ε-переход из состояния q_1 в q_0 и продолжает обработку слова.

Для построения НКА по РВ будем использовать определение регулярного языка. Напомним определение класса регулярных языков REG.

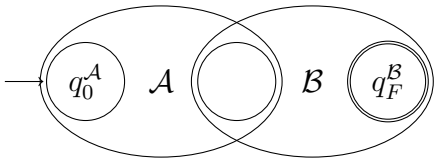
- $\varnothing \in \mathsf{REG}$.
- $\forall \sigma \in \Sigma : \ \{\sigma\} \in \mathsf{REG}$.
- $\forall X, Y \in \mathsf{REG} \ : \ X \cdot Y, \ X|Y, \ X^* \in \mathsf{REG}$.
- Больше нет регулярных языков.

Мы будем строить НКА по РВ из каждого пункта данного определения. С первыми двумя пунктами проблем нет – их я оставляю как лёгкое упражнение. Перейдём сразу к третьему пункту. Допустим уже построены автоматы $\mathcal{A}$ и $\mathcal{B}$ для регулярных языков X и Y соответственно. Мы будем предполагать, что оба автомата имеют всего одно принимающее состояние. Если в автомате несколько принимающих состояний, то можно построить эквивалентный ему автомат с единственным принимающим

состоянием, добавив к множеству состояний состояние q_F, которое будет единственным принимающим, и добавив ε-переходы в q_F из старого множества F: $\forall q \in F : \delta(q, \varepsilon) = q_F$. Будем схематично обозначать автоматы эллипсами, и помечать в них только начальное и принимающее состояние. Таким образом, автомат $\mathcal{A}$ имеет вид



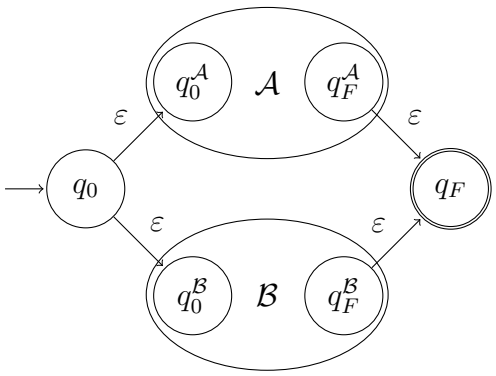
В дальнейшем, мы будем предполагать, что начальное состояние на схеме находится слева, а принимающее справа. Построим явно автомат распознающий $X \cdot Y$, $L(\mathcal{A}) = X, L(\mathcal{B}) = Y$.



Для этого по автомату $\mathcal{A}$ распознающему язык X и автомату $\mathcal{B}$, распознающему язык Y мы строим автомат, распознающий $X \cdot Y$ объединяя множества состояний $\mathcal{A}$ и $\mathcal{B}$ так, что $q_0 = q_0^{\mathcal{A}}$, $q_F^{\mathcal{A}} = q_0^{\mathcal{B}}$, $F = \{q_F^B\}$. Опять получили автомат с единственным принимающим состоянием.

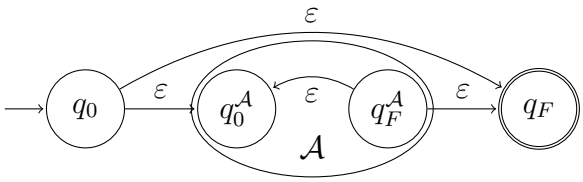
Упражнение 1. Доказать, что построенный автомат распознаёт язык $X \cdot Y$.

Для построения языка $X|Y$ используем следующую конструкцию:



Упражнение 2. Доказать, что построенный автомат распознаёт язык $X|Y$.

И наконец перейдём к построению автомата для языка X^*:



Упражнение 3. Доказать, что построенный автомат распознаёт язык X^*.

Задача 1. Постройте НКА по регулярному выражению $a(ab|b)^*b$.

2 НКА и ДКА

Как мы обсудили на семинаре, 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 $|Q_{\mathcal{B}}| \leqslant 2^{|Q_{\mathcal{A}}|}$. Таким образом, число состояний в ДКА ограниченно экспоненциальной функцией от числа состояний в 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.

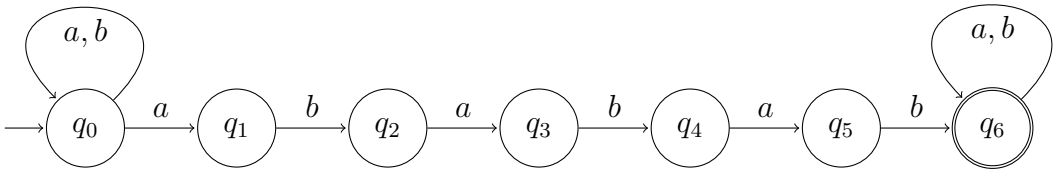
Задача 2. Определим язык $L_i = \{w \,|\, |w| = n, w[n-i] = 1\}$, то есть в язык L_i в ходят все слова, в которых 1 стоит на i-ом месте от конца[2]. Постройте НКА для языка L_3. По построенному НКА постройте ДКА.

Задача 3^* **.** Докажите, что на языках L_i между НКА и построенными по ним ДКА достигается экспоненциальный разрыв.

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

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 $ababab$.

Задача 4. Постройте по данному автомату детерминированный.

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 несобственный[3] префикс слова w, являющийся одновременно его суффиксом.

Пример 2. Приведём пример вычисления префикс-функции.

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

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

У префикс функции есть важное свойство – все несобственные 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 $l(w), l(l(w)), \ldots$

Задача 6^* **.** Приведите алгоритм построения ДКА для языка $\Sigma^* w \Sigma^*$, использующий префикс-функцию.

4 Дополнительные задачи

Задача 7. Приведите протокол работы КМП-алгоритма при поиске подслова *abba* в слове *abbbababbab*.