Задание 4

Автоматы и распознавание текстов II
Лемма о накачке

Ключевые слова [1]:*конечные автоматы (КА), детерминированные и недетерминированные КА, регулярные языки. Примеры нерегулярных языков, лемма о накачке; поиск подстрок, алгоритм Ахо-Корасик.*

1 Алгоритм Ахо-Корасик

Алгоритм Ахо-Корасик позволяет проверять вхождение в текст t слов из заранее построенного словаря. Причём, алгоритм находит все вхождения за линейное время!

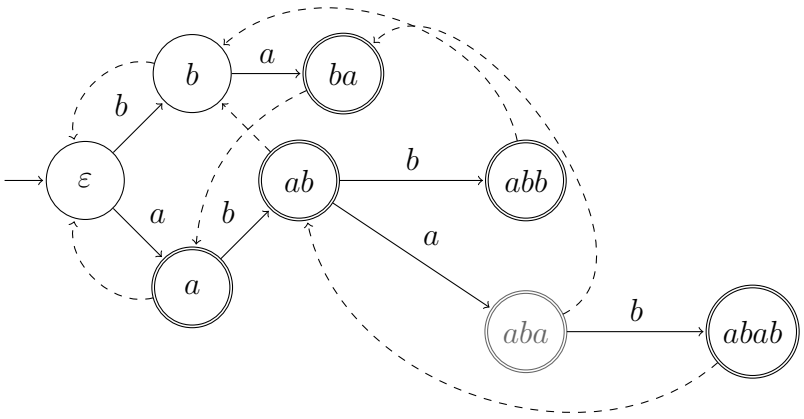

Рис. 1: Автомат Ахо-Корасик

Автомат для алгоритма Ахо-Корасик (рис. 1) построен следующим образом. Он содержит все состояния и переходы, что и соответсвующий

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

автомат для словаря (рис. 1 в задании 3), но помимо этого к принимающим состояниям относятся те состояния, некоторый суффикс которых является принимающим: состояние aba выделено серым, поскольку слово aba не лежит в словаре, однако в словаре лежит слово ba. Пунктирные переходы используются в случае сбоя: переход из состояния u (состояние = некоторое слово) по пунктирной линии осуществляется в случае сбоя: если из u нет перехода по a, то автомат переходит в состояние s, в которое ведёт пунктирная стрелка, и пытается перейти по a из s и т.д. Переход из u в s добавляется согласно следующему правилу. Слово s должно быть самым длинным суффиксом слова u ($u = ps$), который является состоянием автомата Ахо-Корасик. Такие переходы называют *суффиксными ссылками*.

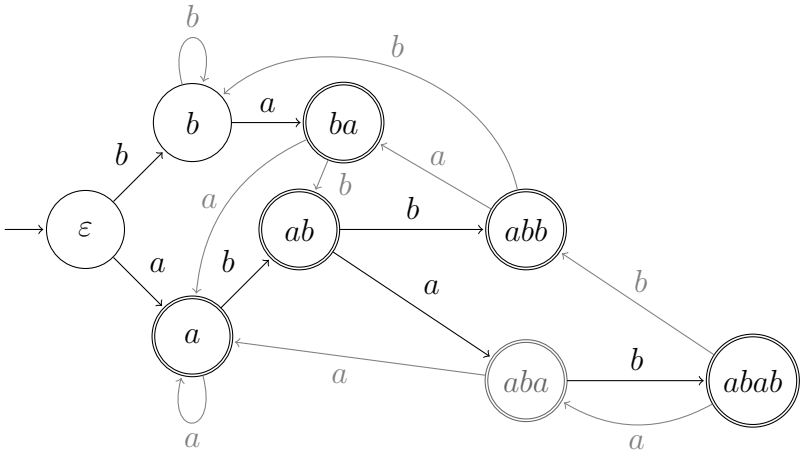

Рис. 2: ДКА Ахо-Корасик

Так, автомат Ахо-Корасик (рис. 1) по сути является ДКА (рис. 2), граф которого получается из автомата Ахо-Корасик добавлением явных переходов, которые были заданы суффиксными ссылками. ДКА (и соответсвенно автомат Ахо-Корасик) оказывается в принимающем состоянии тогда и только тогда, когда в тексте на входе автомата встретилось слово из словаря (возможно несколько одновременно). Заметим, что количество одновременно встретившихся подслов текста из словаря зависит только от состояния автомата.

Непосредственное построение ДКА Ахо-Корасик как правило неэф-

фективно для приложений – суффиксные ссылки позволяют существенно сократить таблицу переходов и уменьшить тем самым длину описания автомата.

Задача 1. Постройте для словаря $S = \{ac, acb, b, ba, c, cbb\}$ (который вы строили в предыдущем задании) автомат Ахо-Корасик. Посчитайте с его помощью количество различных вхождений слов из словаря S в подслово $acbacbb$.

2 Лемма о накачке[2]

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, существует разбиение $w = xyz$, такое что*

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

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 по разбиению $w = xyz$ идёт квантор существования: часто студенты это равенство ошибочно воспринимают как «для любого разбиения».

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

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

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 $xy^i z$, $i \geqslant 0$ принадлежат L.

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 $xy^2 z \in L$, 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 1. 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 случаях 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 2. Будут ли регулярными следующие языки?

1. $L = \{a^{2017n+5} \mid n = 0, 1, \} \cap \{a^{503k+29} \mid k = 401, 402, \ldots\} \subseteq \{a^*\}$.

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

3. Язык $L_3 = \{w \mid |w|_a = |w|_b\}$; через $|w|_a$ обозначают количество букв a в слове w.

4. Язык $L_3 = \{w \mid |w|_{ab} = |w|_{ba}\}$, через $|w|_{ab}$ обозначают количество подслов ab в слове w.