Лекция 5. Регулярные грамматики

Главные вопросы, которые мы обсуждаем, представлены на СЛАЙДЕ 1. Рассмотрим третий способ определения регулярных языков – регулярные грамматики.

5.1 Формальные грамматики

Прежде чем перейти к самим регулярным грамматикам, мы должны формально определиться с грамматиками как таковыми. Понятие грамматики изначально было формализовано лингвистами при изучении естественных языков. Предполагалось, что это может помочь при их автоматической трансляции. Однако наилучшие результаты в этом направлении достигнуты при описании не естественных языков, а языков программирования. СЛАЙДЫ 2-3.

5.2 Право- и леволинейные грамматики

В данном разделе мы рассматриваем грамматики, которые генерируют регулярные языки.

Грамматика $G = (V_T, V_N, P, S)$ называется **праволинейной** (далее – ПЛГ), если все ее продукции имеют две формы (СЛАЙД 4):

$A \text{ -> } xB \mid A \text{ -> } x$

С другой стороны, грамматика $G = (V_T, V_N, P, S)$ называется **леволинейной** (далее – ЛЛГ), если все ее продукции имеют две формы:

$A \text{ -> } By \mid A \text{ -> } y$

Всякая ЛЛГ и всякая ПЛГ является **регулярной грамматикой** (далее – РГ). Очевидно, что в РГ в правой части любой продукции не может быть больше одного нетерминального символа. Более этого, этот нетерминал может быть крайним левым или крайним правым символом правой части продукции.

Пример ПЛГ (СЛАЙД 5):

$G_1 = (\{a, b\}, \{S\}, \{S \text{ -> } abS \mid a\}, S)$.

Пример ЛЛГ:

$G_2 = (\{a, b\}, \{S, S_1, S_2\}, \{S \text{ -> } S_1ab, S_1 \text{ -> } S_1ab \mid a, S_2 \text{ -> } a\ \}, S)$.

Очевидно, что согласно определению РГ, данному выше, к этому классу относятся как G_1, так и G_2.

Например, грамматикой G_1 может быть порождена следующая цепочка:

$S \text{=>} abS \text{=>} ababS \text{=>} ababa$.

Из этой строчки легко предположить, что $L(G_1)$ эквивалентно определяется РВ $(ab)^*a$. И похожим образом $L(G_2)$ эквивалентно определяется РВ $aab(ab)^*$. Очевидно, что оба они являются РЯ.

Грамматика $G = (\{a, b\}, \{S, A, B\}, \{S \text{ -> } A, A \text{ -> } aB \mid \varepsilon, S \text{ -> } Ab\}, S)$ не является РГ. Хотя каждая продукция в ней принимает леволинейную либо праволинейную форму, сама грамматика ни ЛЛГ, ни ПЛГ. Следовательно, она – не РГ, но она является линейной грамматикой (далее – ЛГ). В ЛГ в правых частях продукций не может быть более одного нетерминала, но без ограничения на его местоположение.

Очевидно, ЛГ всегда РГ, но не всякая ЛГ является РГ.

Далее мы должны показать, что РГ связаны с РЯ, и что для каждого РЯ существует

РГ. Иными словами, мы попробуем доказать, что это еще один способ задания РЯ.

Первым делом мы покажем, что язык, генерируемый ПЛГ всегда РЯ (СЛАЙД 6). Для этого мы сконструируем НКА, который имитирует процесс порождения по ПЛГ. Важно, что все сентенциальные формы ПЛГ имеют такой вид, что в них ровно один нетерминал, и он появляется в крайней правой позиции. Предположим, выполняется такой шаг порождения:

$ab...cD => ab...cdE$,

который выполняется применением продукции D -> dE. В НКА это можно имитировать переходом из состояния D в состояние E по символу d. В этой схеме состоянию автомата соответствует нетерминал в сентенциальной форме, когда уже обработанная часть входной строки идентична терминальному префиксу сентенциальной формы. Эта простая идея служит базисом следующей теоремы.

Теорема 5.1. Если грамматика G является ПЛГ, то $L(G)$ – РЯ (СЛАЙД 7).

Доказательство. Мы предполагаем, что $V = \{V_1, V_2, ...\}$, и продукции имеют форму V_0 -> v_1V_i, V_i -> v_2V_j, ...,V_n -> v_l, ... Если w – строка из $L(G)$, то вследствие формы продукций порождение будет иметь следующую форму:

$$V_0 => v_1V_i => v_1v_2V_j =>^* v_1v_2...v_kV_n => v_1v_2...v_kv_1 = w \qquad (5.1)$$

Автомат, который мы конструируем, будет репродуцировать это порождение путем поочередного «потребления» каждого из этих v. Стартовое состояние будет помечено нетерминалом V_0, а для каждого нетерминала V_i мы будем иметь состояние V_i. Соответственно, для каждой продукции вида V_i -> $a_1a_2...a_mV_j$ у автомата будут переходы, соединяющие V_i и V_j. Значит, расширенная функция переходов может быть определена так:

$$\hat{\delta}(V_i, a_1a_2\ldots a_m) = V_j .$$

Для каждой продукции вида V_i -> $a_1a_2...a_\mathrm{m}$ соответствующий переход автомата будет определяться функцией:

$\hat{\delta}(V_i, a_1a_2\ldots a_m) = V_f$, где V_f – это заключительное состояние.

Промежуточные состояния, которые нужны, чтобы сделать это, необязательны. Их можно задать произвольными метками. Полный автомат собирается из таких индивидуальных частей.

Теперь предположим, что строка w из $L(G)$ такая, что удовлетворяет выражению (5.1). В НКА есть путь из V_1 в V_i с меткой v_1, путь из V_i в V_j с меткой v_2, и так далее, тогда $V_f \in \hat{\delta}(V_0, w)$, и строка w принимается НКА.

И обратно, предположим, что строка w принимается НКА. Из-за наличия способа конструирования НКА для того, чтобы принять строку автомат должен пройти через последовательность состояний V_0, V_1,…,V_f с метками v_1, v_2, … Следовательно, строка должна иметь форму $w = v_1v_2...v_kv_l$ и порождение

$$V_0 => v_1V_i => v_1v_2V_j =>^* v_1v_2...v_kV_n => v_1v_2...v_kv_1$$

возможно. Следовательно, w принадлежит $L(G)$, и теорема доказана.

Пример 28. Создадим КА, принимающий язык, сгенерированный продукциями (V_0->aV_1, V_1->abV_0 | b). Мы начинаем построение графа переходов с вершин V_0, V_1, V_f. Первая продукция дает ребро от V_0 к V_1 с меткой a. Согласно второй продукции нам потребуется создать дополнительную вершину графа, чтобы появился путь от V_1 к V_0 с меткой ab. Наконец, нам нужно ребро с меткой b между V_1 к V_f. Язык, который принимается полученным КА и генерируется заданной РГ, можно описать РВ $(aab)^*(ab)$. СЛАЙД 8.

Чтобы показать, что любой РЯ можно сгенерировать ПЛГ, мы начинаем от ДКА для этого языка и изменяем на обратный порядок процесса конструирования, использованный в теореме 5.1.

Состояния ДКА становятся нетерминалами грамматики, а символы на переходах – терминалами в продукциях грамматики.

Теорема 5.2. Если L – РЯ на алфавите Σ, то существует ПЛГ G такая, что $L = L(G)$.

Доказательство. Пусть A – это ДКА, который принимает язык L. Полагаем $Q = \{q_1, q_2, \ldots, q_n\}$ и $\Sigma = \{a_1, a_2, \ldots, q_m\}$. Сконструируем ПЛГ $G = (\Sigma, V, P, S)$, где $V = \{q_1, q_2, \ldots, q_n\}$ и $S = q_0$. Для каждого перехода $\delta(q_i, a_j) = q_k$ мы добавляем в P продукцию

$$q_i \text{ -> } a_j q_k \qquad (5.2)$$

Если q_k принадлежит F, то в P добавляется продукция

$$q_k \text{ -> } \varepsilon \qquad (5.3)$$

Покажем, что грамматика G, созданная таким образом, может генерировать любую строку в L. Рассмотрим w из L, причем $w = a_i a_j \ldots a_k a_l$. Автомат A для приема этой строки должен двигаться через

$\delta(q_0, a_i) = q_p$, $\delta(q_p, a_j) = q_r$, …, $\delta(q_s, a_k) = q_t$, $\delta(q_t, a_l) = q_f$ из F.

Согласно нашему построению в грамматике будет одна продукция для каждой из этих δ. Следовательно, мы можем осуществить

$$q_0 \Rightarrow a_i q_p \Rightarrow a_i a_j q_r \Rightarrow^* a_i a_j \ldots a_k q_i \Rightarrow a_i a_j \ldots a_k a_l q_f \Rightarrow a_i a_j \ldots a_k a_l \qquad (5.4)$$

с использованием грамматики G и w из $L(G)$.

И обратно, если w принадлежит $L(G)$, то ее порождение должно иметь форму (5.4). Это означает, что

$\hat{\delta}(q_0, a_i a_j \ldots a_k a_l) = q_f$. Что и требовалось доказать. СЛАЙД 9.

Дополнительно отметим, что наше ограничение относительно детерминированности A не существенно для доказательства теоремы 5.2. С небольшими изменениями мы можем сконструировать грамматику для НКА.

Пример 29. Создадим ПЛГ для языка $L(aab^*a)$. Функция переходов для НКА с соответствующими продукциями приведена в следующей таблице (СЛАЙД 10).

$\delta(q_0, a) = \{q_1\}$	$q_0 \text{ ->} aq_1$
$\delta(q_1, a) = \{q_2\}$	$q_1 \text{ ->} aq_2$
$\delta(q_2, b) = \{q_2\}$	$q_2 \text{ ->} bq_2$
$\delta(q_2, a) = \{q_f\}$	$q_2 \text{ ->} aq_f$
q_f из F	$q_f \text{ ->} \varepsilon$

Результат был получен согласно простой методике, описанной в теореме 5.2. Строку *aaba* можно породить следующим образом.

$q_0 \Rightarrow aq_1 \Rightarrow aaq_2 \Rightarrow aabq_2 \Rightarrow aabaq_f \Rightarrow aaba$.

Теоремы 5.1 и 5.2 устанавливают связь между РЯ и ПЛГ. Мы можем установить похожую связь между ЛЛГ и РЯ, тем самым показывая полную эквивалентность РГ и РЯ (СЛАЙД 11).

Теорема 5.3. Язык L – РЯ тогда и только тогда, когда существует ЛЛГ G такая, что $L = L(G)$.

Доказательство. Мы дадим лишь общую идею. Дана ЛЛГ с продукциями в форме A -> Bv и A -> v. Мы преобразуем ее в ПЛГ G_{rl} путем замены каждой продукции G, соответственно, на A -> $v^R B$ или A -> V^R. Потренировавшись на нескольких примерах, мы поймем, что $L(G) = (L(G_{rl}))^R$. Как известно, обращение любого РЯ так же является РЯ. Поскольку G_{rl} – ПЛГ, то $L(G_{rl})$ – является РЯ, но тогда такими также будут $L(G_{rl}))^R$ и $L(G)$. Что и требовалось.

Совмещая теоремы 5.2 и 5.3 мы подошли к эквивалентности РЯ и РГ.

Теорема 5.4. Язык L – РЯ тогда и только тогда, когда существует РГ G такая, что $L = L(G)$.

Без доказательства.

5.4 Свойства замкнутости регулярных языков

В этом параграфе мы приведем и попробуем доказать несколько теорем вида «если определенные языки регулярны, а язык L построен на их основе с помощью определенных операций, то язык L также РЯ». Эти теоремы часто в литературе называют **свойствами замкнутости** РЯ, т.к. в них утверждается, что класс РЯ замкнут относительно определенных операций. Эти свойства выражают идею того, что если один или несколько языков регулярны, то языки, определенным образом с ними связанные, также РЯ. Данные свойства, помимо прочего, служат иллюстрацией того, как эквивалентные представления РЯ подкрепляют друг друга в нашем понимании этих языков, поскольку обычно один способ представления лучше других подходит для доказательства некоторого свойства замкнутости.

Основные свойства замкнутости РЯ выражаются в том, что они замкнуты относительно следующих операций: объединение; пересечение; дополнение; разность; обращение; итерация; конкатенация; гомоморфизм; обратный гомоморфизм.

Замкнутость относительно булевых операций

Предположим, у нас имеется два РЯ – L и M над алфавитом Σ. Операция объединения двух языков нами рассматривалась ранее. СЛАЙД 12.

Пересечением называется язык $L \cap M$, который содержит все строки, принадлежащие обоим языкам.

Дополнением языка L называется язык L_{compl}, который содержит множество тех строк в алфавите Σ^*, которые не принадлежат L.

Класс РЯ замкнут относительно всех трех булевых операций (СЛАЙД 13).

Теорема 5.5. Если L и M – РЯ, то их объединение тоже РЯ.

Доказательство. Поскольку оба языка регулярны, то им соответствуют некоторые РВ. Пусть $L = L(R)$ и $M = L(S)$. Тогда $L \cup M = L(R+S)$ согласно определению операции объединения для РВ. Все оказалось не так страшно.

При определении свойства замкнутости относительно дополнения, также используют РВ. Разумеется, легким образом преобразовать РВ так, чтобы оно представляло собой дополнение заданного языка, у нас не получится. Однако это возможно, если следовать простой методике:

1. Преобразовать РВ в ε-НКА.
2. Преобразовать ε-НКА в ДКА с помощью конструкции подмножеств.
3. Дополнить заключительные состояния этого ДКА.
4. Преобразовать полученный ДКА обратно в РВ, используя известные способы.

Теорема 5.6. Если L – РЯ, то язык $L_{compl} = \Sigma^* - L$ тоже РЯ.

Доказательство. Пусть $L = L(A)$ для некоторого ДКА $A(Q, \Sigma, \delta, q_0, F)$. Тогда $L_{compl} = L(B)$, где В – это ДКА $(Q, \Sigma, \delta, q_0, Q - F)$. Иначе говоря, оба автоматы отличаются только тем, что заключительные состояния A стали незаключительными состояниями в B, и наоборот. Тогда w принадлежит $L(B)$, если и только если $\hat{\delta}(q_0, w)$ принадлежит $Q - F$, т.е. w не принадлежит $L(A)$.

Пример 30. Возьмем язык, содержащий строки из 0 и 1, которые всегда заканчиваются на 01 и определяются РВ (0+1)*01. Дополнением является язык, содержащий строки из 0 и 1, которые **не заканчиваются** на 01. Он описывается РВ (1+00*1(00*1)*1)*. К слову, оба ДКА отличаются только тем, что заключительные состояния стали незаключительными и наоборот. СЛАЙД 14.

Рассмотрим пересечение двух языков. Как известно, операции объединения, дополнения и пересечения не являются независимыми. Следовательно, пересечение можно выразить через объединение и дополнение с помощью одного из законов де Моргана, $L \cap M = (L_{compl} \cup M_{compl})_{compl}$.

Вместе с тем, мы можем непосредственно построить ДКА для пересечения двух РЯ (СЛАЙД 15).

Теорема 5.7. Если L и M – РЯ, то язык $L \cap M$ тоже РЯ.

Доказательство. Пусть ДКА $A_L(Q_L, \Sigma, \delta_L, q_L, F_L)$ и $A_M(Q_M, \Sigma, \delta_M, q_M, F_M)$ – это ДКА для заданных языков. Мы должны сконструировать ДКА A, который является комбинацией двух заданных автоматов. Формально,

$A = (Q_L \times Q_M, \Sigma, \delta, (q_L, q_M), (F_L \times F_M))$, где $\delta((p,q),a) = (\delta_L(p,a), \delta_M(q,a))$.

Достаточно легко с помощью индукции показать, что любая строка w принимается таким объединенным автоматом, если и только если ее допускают оба исходных автомата, т.е. $\hat{\delta}((q_L, q_M), w) = (\hat{\delta}_L(q_L, w), \hat{\delta}_M(q_M, w))$.

Существует и еще одна операция над множествами – разность. Если L и M – языки, то **разностью** $L - M$ называется множество строк, которые принадлежат L и не принадлежат M. СЛАЙД 16

РЯ замкнуты относительно этой операции.

Теорема 5.8. Если L и M – РЯ, то язык $L - M$ тоже РЯ.

Доказательство. Заметим, что $L - M = L \cap M_{compl}$. По теореме 5.6 язык M_{compl} – РЯ. По теореме 5.7 – $L \cap Mcompl$ – РЯ. Значит, (L – M) – РЯ.

Свойства замкнутости РЯ относительно конкатенации и итерации доказываются также, как и доказательство замкнутости относительно объединения, т.е. через операции над РВ.

Замкнутость относительно других операций

Обращением строки $a_1a_2\ldots a_n$ называется (СЛАЙД 17) строка, записанная в обратном порядке, и для произвольной строки w обозначается как w^R.

Обращение языка L, обозначаемое через L^R, состоит из всех строк, обратных строкам языка L.

Доказательство замкнутости РЯ относительно обращения может выполняться двумя способами, первый основан на автоматах (мы приведем его), второй – на РВ (можно ознакомиться в соответствующих источниках).

Если для автомата A есть язык L такой, что $L=L(A)$, то можно построить КА для L^R следующим образом (СЛАЙД 18).

1. Обратить все дуги на диаграмме переходов автомата A.

2. Сделать начальное состояние A единственным заключительным состоянием нового автомата.

3. Создать начальное состояние p_0 с ε-переходами во все заключительные состояния автомата A.

Будет получен КА, имитирующий A в обратном порядке, а значит, допускающий строку w тогда и только тогда, когда A допускает w^R.

Доказательство, использующее РВ, сводится к базисным правилам и индукции по трем РВ-операторам.

Базис: Если E равно ε, $\varnothing$ или a, где a – это некоторый символ, то E^R совпадает с E.

Окончание доказательства свойства замкнутости РЯ относительно обращения, основанного на РВ, мы оставляем в качестве самостоятельного упражнения. Вместо этого приведем следующий пример.

Пример 31.

Пусть язык L определяется РВ (0+1)0*. Тогда по правилу конкатенации L^R – это язык, описываемый выражением $(0^*)^R(0+1)^R$. Если применять правила итерации и объединения к двум частям этого выражения, а потом использовать базисное правило, то получим, что язык L^R определяется РВ (0+1)0*. СЛАЙД 19.

Гомоморфизм строк – это такая функция на множестве строк, которая подставляет определенную строку вместо каждого ее символа. СЛАЙД 20.

Пример 32.

Функция h, определенная как $h(0) = ab$ и $h(1) = \varepsilon$, является гомоморфизмом. В любой строке из 0 и 1 эта функция заменяет все нули подстрокой ab, а все единицы – пустой строкой. Если мы применим эту функцию h к строке 1100, то получим $abab$.

Гомоморфизм языка определяется с помощью его применения к каждой строке языка. Иными словами, если L – язык в алфавите Σ, а h – гомоморфизм на Σ, то $h(L) = \{h(w) \mid w$ принадлежит $L\}$. Рассмотрим язык РВ 10^*1, т.е. все строки, которые начинаются и заканчиваются единицами, а между ними – произвольное количество нулей. Пусть h – это гомоморфизм из примера 32. Тогда $h(L)$ – это язык РВ $(ab)^*$. Это легко объяснить тем, что гомоморфизм исключает все единицы, а вместо каждого нуля подставляет подстроку ab.

Гомоморфизм можно применять в обратном направлении (**обратный гомоморфизм**). Пусть h – это гомоморфизм над алфавитом Σ в строки, заданные в другом алфавите T. Пусть L – язык в алфавите T. Тогда $h^{-1}(L)$, читаемое как «обратное h от L», – это множество строк w из Σ^*, для которых $h(w)$ принадлежит L. СЛАЙД 21.

Пример 33.

Пусть L – язык РВ $(00+1)^*$, т.е. все строки из 0 и 1, где нули встречаются парами, и пусть h – это гомоморфизм $h(a) = 01$, $h(b)=10$. Тогда $h^{-1}(L)$ – это язык РВ $(ba)^*$, т.е. все строки, в которых повторяются пары ba.

Примем без доказательства две теоремы.

Теорема 5.9. Если L – РЯ в заданном алфавите, и h – гомоморфизм на этом алфавите, то язык $h(L)$ – также РЯ.

Теорема 5.10. Если h – гомоморфизм из алфавита Σ в алфавит T, L – РЯ в алфавите T, то язык $h^{-1}(L)$ – также РЯ. СЛАЙД 22.

5.5 Доказательство нерегулярности заданных языков

Не каждый язык является РЯ. В этом параграфе мы показываем методику доказательства нерегулярности языков, известная как *лемма о разрастании*.

Рассмотрим язык $L_{01} = \{0^n1^n \mid n \geq 1\}$. Это язык, содержащий определенное количество 0, за которыми следует точно такое же количество 1.

Неформально мы можем сказать, что если бы этот язык был РЯ, то допускался бы некоторым ДКА A, имеющим k состояний. Допустим на вход этого автомата поступает k нулей. Он находится в некотором состоянии после чтений k+1 префиксов входной строки. В нашем случае это ε, 0, 00, …, 0^k. Поскольку есть только k различных состояний, то прочитав два различных префикса, ДКА должен находиться в одном и том же состоянии, например, q.

Допустим, прочитав i и j нулей, A получает на вход 1. По прочтении i единиц он должен допустить строку, если ранее было получено i нулей и отвергнуть ее, если ДКА получил j нулей. Однако в момент поступления 1 автомат находится в состоянии q и не может вспомнить, какое именно количество нулей было принято. Значит, его можно перехитрить и заставить работать неправильно, т.е. допускать вход, когда он не должен этого делать и наоборот.

Это доказательство можно считать точным. Однако мы можем прийти к заключению о нерегулярности с помощью следующей теоремы (СЛАЙД 23).

Теорема 5.11 («Лемма о разрастании для РЯ»). Пусть L – РЯ, и существует константа n, для которой каждую строку w из L, удовлетворяющую неравенству $|w| \geq n$, можно разбить на три строки $w = xyz$ так, что выполняются условия:

1. $y \neq \varepsilon$.
2. $|xy| \leq n$.
3. Для любого $k \geq 0$ строка $xy^k z$ также принадлежит L.

Это значит, что всегда можно найти такую строку y недалеко от начала строки w, которая может разрастись. Если строку y повторить любое количество раз или удалить ее (k=0), то результирующая строка все равно будет принадлежать языку L.

Доказательство. Пусть L – РЯ, тогда L=$L(A)$ для некоторого A. Пусть A имеет n состояний. Рассмотрим произвольную строку w длиной не менее n, например, $w = a_1a_2 \ldots a_m$, где $m \geq n$ и каждый a_i есть входной символ. Для $i = 0,1,\ldots,n$ определим состояние p_i как $\hat{\delta}(q_0, a_1a_2...a_i)$, причем p_0=q_0.

Рассмотрим n+1 состояний p_i при i = 0,1,…,n. Поскольку у КА n различных состояний, то всегда найдутся два разных числа i и j ($0 \leq i < j \leq n$) при которых p_i=p_j.

Теперь разобьем строку w на xyz.

1. $x = a_1a_2 \ldots a_i$
2. $y = a_{i+1}a_{i+2} \ldots a_j$
3. $z = a_{j+1}a_{j+2} \ldots a_m$

Таким образом, x приводит КА в состояние p_i, y – из p_i обратно в p_i, а z – остаток строки w. Строка x может быть пустой при i=0, строка z – при $j = n = m$. А вот y не может быть пустой строкой из-за строгого неравенства.

Что же происходит, когда на вход поступает строка xy^kz для любого неотрицательного k. При k=0 наш автомат переходит из q_0 в p_i, прочитав x. Поскольку $p_i = p_j$, то z переводит A из p_i в заключительное состояние.

Если $k < 0$, то по x автомат переходит из q_0 в p_i, затем, читая y, он k раз циклически проходит через p_i , а затем по z переходит в заключительное состояние. Иначе говоря, для любого неотрицательного k строка xy^kz также принимается автоматом A, т.е. принадлежит языку L.

Литература к лекции 5

1. Гилл, А. Введение в теорию конечных автоматов / А. Гилл. – М.: Наука, 1966. – 272 с.
2. Кузнецов, А.С. Теория вычислительных процессов [Текст] : учеб. пособие / А. С. Кузнецов, М. А. Русаков, Р. Ю. Царев ; Сиб. федерал. ун-т. - Красноярск: ИПК СФУ, 2008. – 184 с.
3. Короткова, М.А. Математическая теория автоматов. Учебное пособие / М.А. Короткова. – М.: МИФИ, 2008. – 116 с.
4. Молчанов, А. Ю. Системное программное обеспечение. 3-е изд. / А.Ю. Молчанов. – СПб.: Питер, 2010. – 400 с.
5. Теория автоматов / Э. А. Якубайтис, В. О. Васюкевич, А. Ю. Гобземис, Н. Е. Зазнова, А. А. Курмит, А. А. Лоренц, А. Ф. Петренко, В. П. Чапенко // Теория вероятностей. Математическая статистика. Теоретическая кибернетика. — М.: ВИНИТИ, 1976. — Т. 13. — С. 109–188. — URL http://www.mathnet.ru/php/getFT.phtml?jrnid=intv&paperid=28&what=fullt&option_lang=rus
6. Серебряков В. А., Галочкин М. П., Гончар Д. Р., Фуругян М. Г. Теория и реализация языков программирования — М.: МЗ-Пресс, 2006 г., 2-е изд. - http://trpl7.ru/t-books/TRYAP_BOOK_Details.htm
7. Finite State Machine Generator - http://sourceforge.net/projects/genfsm/
8. Введение в схемы, автоматы и алгоритмы - http://www.intuit.ru/studies/courses/1030/205/info