Лекция 6. Автоматы с магазинной памятью

Главные вопросы, которые мы обсуждаем, представлены на СЛАЙДЕ 1. Покажем более сильную, нежели КА, абстрактную машину с вспомогательной памятью стекового типа. Покажем две различные версии таких машин с точки зрения их функционирования, а также рассмотрим, как ввести в эту машину детерминизм.

6.1 Неформальное описание автоматов с магазинной памятью

Автомат с магазинной памятью (PDA – *PushDown Automaton*), или магазинный автомат (далее – МПА) – это фактически ε-НКА, дополненный памятью магазинного (или стекового типа). В магазине (стеке) хранится последовательность «магазинных символов». Наличие магазина означает, что в отличие от КА, МПА помнит бесконечное количество информации. С другой стороны, МПА отличается от универсальной вычислительной машины тем, что в соответствии с принципом LIFO имеет доступ к информации только на одном конце магазина. СЛАЙД 2.

Как следствие существуют языки, распознаваемые некоторой программой и нераспознаваемые ни одним МПА. Как мы покажем далее, МПА распознают так называемые контекстно-свободные языки. Таким, например, является язык $L = \{a^n b^n : n > 0\}$

МПА можно рассматривать как устройство, представленное на СЛАЙДЕ 3. Элемент, обозначенный как «конечное управление», читает входные символы по одному. МПА может обозревать символ на вершине магазина и совершать переход на основе текущего состояния, входного символа и символа на вершине магазина. Он может выполнять и ε-переход. За один переход МПА совершает следующие операции (СЛАЙД 4):

1. Читает и пропускает входной символ, используемый при переходе. Если в качестве входа используется ε, то входные символы пропускаются.

2. Переходит в новое состояние, которое может не отличаться от предыдущего.

3. Заменяет символ на вершине магазина некоторой строкой, которая может быть пустой (означает снятие символа с вершины магазина). Это может быть тот же символ, что и был на вершине, тогда магазин не меняется. МПА может заменить магазинный символ, что равносильно изменению вершины без снятий и вталкиваний. Символ может быть заменен несколькими символами, что означает возможное изменение символа на вершине с последующим вталкиванием туда одного или нескольких новых символов.

Пример 34.

Рассмотрим язык $L_{wwr} = \{ww^R \mid w$ принадлежит $(0+1)^*\}$, который образован палиндромами четной длины над алфавитом $\{0, 1\}$. Можно дать следующее неформальное описание МПА, допускающего L_{wwr}. СЛАЙД 5.

1. Работа начинается в состоянии q_0, представляющем нашу догадку о том, что середина входной строки еще не достигнута . В этом состоянии символы считываются и их копии вталкиваются в магазин.

2. В любой момент можно предположить, что достигнута середина входной строки. В этот момент w находится в магазине, причем левый ее конец лежит на дне, правый – на вершине. Этот выбор можно отметить спонтанным переходом в состояние q_1. Т.к. наш МПА недетерминирован, в действительности предполагаются две возможности: или достигнута середина строки w, или можно входные символы читать дальше, вталкивая их

в магазин. В этом случае МПА остается в состоянии q_0.

3. В состоянии q_1 входные символы сравниваются с символами на вершине магазина. Если они одинаковые, то символ пропускается, магазинный удаляется, и работа продолжается. Если они не совпадают, то предположение о середине строки ошибочно. Иначе говоря, за предполагаемой строкой w нет w^R. Это тупиковая ветвь, она отбрасывается, хотя другие могут продолжаться и вести к тому, что строка допускается автоматом.

4. Если МПА опустошен, то на самом деле обнаружена строка w, за которой идет w^R. Следовательно, вся прочитанная строка допускается.

6.2 Формальное определение автомата с магазинной памятью

Формальная запись МПА содержит семь компонентов и выглядит следующим образом (СЛАЙД 6).

$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Их смысл:

- Q – конечное множество состояний.
- Σ – алфавит входных символов.
- Γ – конечный магазинный алфавит, т.е. те символы, который могут размещаться в магазине.

δ – функция переходов, которая управляет поведением МПА. Аргументами являются тройки $\delta(q, a, X)$, где q – состояние, a – входной символ или ε, X – магазинный символ. Выходом является пара (p, γ), где p – новое состояние, а γ – последовательность магазинных символов, замещающая X на вершине магазина. Если $\gamma = \varepsilon$, то магазинный символ снимается, если $\gamma = X$, то магазин не меняется, если $\gamma = YZ$, то X заменяется на Z, а Y помещается в магазин.

q_0 – начальное состояние МПА.

Z_0 – начальный магазинный символ (или **маркер дна**), который содержит МПА до начала работы.

F – множество заключительных состояний.

Пример 35.

Построим МПА P, допускающий язык L_{wwr} из примера 34. Магазинный символ Z_0 используется для отметки дна магазина, он должен присутствовать в магазине, чтобы удалив подстроку w и обнаружив на входе ww^R, можно было перейти в заключительное состояние q_2. Наш МПА можно представить в следующем виде (СЛАЙД 7).

$P = (\{q_0, q_1, q_2\}, \{0,1\}, \{0,1,Z_0\}, \delta, q_0, Z_0, \{q_2\})$

Функция переходов определяется такими правилами.

1. $\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\}, \delta(q_0, 1, Z_0) = \{(q_0, 1Z_0)\}$. Одно из этих двух правил применяется в начале работы МПА.

2. $\delta(q_0, 0, 0) = \{(q_0, 00)\}, \delta(q_0, 0, 1) = \{(q_0, 01)\}, \delta(q_0, 1, 0) = \{(q_0, 10)\}, \delta(q_0, 1, 1) = \{(q_0, 11)\}$. Все эти правила позволяют оставаться в состоянии q_0 и читать из входной строки символы, вталкивая их в магазин.

3. $\delta(q_0, \varepsilon, Z_0) = \{(q_1, Z_0)\}, \delta(q_0, \varepsilon, 0) = \{(q_1, 0)\}, \delta(q_0, \varepsilon, 1) = \{(q_1, 1)\}$. Эти правила позволяют спонтанно переходить из состояния q_0 в состояние q_1 без изменения вершины магазина.

4. $\delta(q_1, 0, 0) = \{(q_1, \varepsilon)\}, \delta(q_1, 1, 1) = \{(q_1, \varepsilon)\}$. В состоянии q_1 входные символы проверяются на равенство вершине магазина. При совпадении символы из магазина выталкиваются.

5. $\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$. Если в состоянии q_1 обнаружен маркер дна магазина, это означает наличие строки вида ww^R, поэтому МПА переходит в состояние q_2 и принимает строку.

Графическое представление МПА

Функцию переходов в виде списка, как в нашем примере, отследить нелегко. Более удобным способом может оказаться граф, похожий на диаграмму переходов КА. Введем в наш обиход диаграммы переходов МПА со следующими свойствами.

1. Вершины соответствуют состояниям МПА.

2. Треугольником обозначается начальное состояние, двойным кружком – заключительное состояние.

3. Дуги соответствуют переходам МПА. Их смысл таков. Дуга a, X;α, ведущая из состояния q в состояние p означает, что $\delta(q, a, X)$ содержит пару (p, α), возможно наряду с другими.

4. Если не оговаривается иное, то стартовым магазинным символом является Z_0 (примечание: в системе JFLAP маркер дна обозначается через Z).

Диаграмма переходов МПА из примера 35 приведена на СЛАЙДЕ 8.

Конфигурации МПА

Интуитивно понятно, что МПА переходит от конфигурации к конфигурации в соответствии с символами входной строки (или ε), но в отличие от КА конфигурация МПА включает помимо состояния (допустим, q) еще содержимое магазина (обозначим γ). Он может быть произвольно большим, то нередко он оказывается самой большой частью конфигурации. Кроме того, иногда бывает полезным в конфигурацию включить непрочитанную часть входной строки (допустим, w).

Таким образом, конфигурация МПА представляется тройкой (q, w, γ). Вершину магазина изображаем слева, а дно – справа. Такая тройка называется **конфигурацией** МПА, или его **мгновенным описанием** (далее – МО, *instantaneous description* – ID).

Для КА его МО – это всего лишь его состояние. Значит, для последовательностей конфигураций, через которые он проходил, можно использовать расширенную функцию переходов. Однако для МПА этого недостаточно, поэтому мы будем использовать пары конфигураций, связи между которыми представляют собой переходы. СЛАЙД 9.

Если $P(\ldots)$ – это МПА, то мы можем определить бинарное отношение $\vdash_P$, или просто $\vdash$, когда это понятно из контекста, следующим образом. Пусть $\delta(q, a, Z)$ содержит (p, α). Тогда для всех строк w из Σ^* и β из Γ^* полагаем $\delta(q, aw, X\beta) \vdash \delta(p, w, \alpha\beta)$.

Этот переход отражает мысль о том, что прочитывая на входе возможно пустой символ α и заменяя X на вершине строкой α, можно перейти из состояния q в состояние p. Оставшаяся часть входной строки и содержимое магазина под его вершиной не влияют на работу МПА. У нас они сохраняются для того, чтобы повлиять на дальнейшие события.

Для представления 0 или более переходов МПА можно воспользоваться обозначением $\vdash_P^*$. Теперь можно дать следующее определение с базисом обозначением $I \vdash^* I$.

В индуктивной части определения $I \vdash^* J$, если существует такое МО K, которое удовлетворяет условиям $I \vdash^* K$ и $K \vdash^* J$. По-другому, $I \vdash^* J$, если существует последовательность МО $K_1, \ldots, K_n$, у которой $I = K_1$, $J = K_n$, и $K_i \vdash K_{i+1}$ для $i = 1, \ldots, n$-1. СЛАЙД 10.

Пример 36.

Проследим работу МПА из примера 35 со входом 1111. Поскольку q_0 – начальное состояние, и Z_0 – стартовый символ, то начальное МО – $(q_0, 1111, Z_0)$. На этом входе МПА может несколько раз сделать ошибочные предположения. Вся последовательность МО, достижимых из начальной конфигурации, показана на СЛАЙДЕ 11. Стрелками представляется бинарное отношение $\vdash$.

Из начального МО можно выбрать два перехода. Первый предполагает, что середина палиндрома не достигнута, и ведет к МО $(q_0, 111, 1Z_0)$. В магазин вталкивается символ 1.

Второй вариант выбора предполагает, что середина уже достигнута. Без считывания очередного символа МПА переходит в q_1, что приводит к МО (q_1, 1111, Z_0). МПА может допустить вход, если он находится в состоянии q_1 и обозревает символ Z_0, он переходит в МО (q_2, 1111, Z_0). Это МО не является в точности заключительным, поскольку вход не дочитан до конца. Если бы входом вместо 1111 была ε, то та же последовательность привела бы к МО (q_0, ε, Z_0), показывающему, что ε допущена.

МПА может предположить, что он увидел середину уже после первой 1, т.е. МО (q_0, 111, 1Z_0). Очевидно, что и это предположение ведет к ошибке, поскольку весь вход не может быть прочитан до конца. Правильное предположение дает последовательность МО (q_0, 1111, Z_0) |— (q_0, 111, 1Z_0) |— (q_0, 11, 11Z_0) |— (q_1, 1, 1Z_0) |— (q_1, ε, Z_0) |— (q_2, ε, Z_0). Строка принята.

Принципиальными также являются следующие моменты. СЛАЙД 12.

1. Если последовательность конфигурации (иначе говоря – **вычисление**) является допустимой (*legal*) для МПА, то вычисление, образованное путем дописывания одной и той же строки к концам входных строк всех его конфигураций, также допустимо.

2. Если вычисление допустимо для МПА, то вычисление, образованное дописыванием одних и тех же магазинных символов внизу магазина в каждой конфигурации, также допустимо.

3. Если вычисление допустимо для МПА, и в результате некоторый **суффикс** (*tail*) входной строки не прочитан, то вычисление, полученное удалением суффикса из входных строк каждой конфигурации, также допустимо.

Примем без доказательства две теоремы.

Теорема 6.1. Если $P = (\ldots)$ – МПА, и $(q, x, \alpha) \vdash_P^* (p, y, \beta)$, то для любой строки w из Σ^* и β из Γ^* верно утверждение:

$(q, xw, \alpha\gamma) \vdash_P^* (p, yw, \beta\gamma)$.

Для $\gamma = \varepsilon$ получается формальное утверждение принципа 1, а при $w = \varepsilon$ – принципа 2.

Обращение теоремы 6.1 неверно. Существуют действия, которые МПА мог бы совершить выталкиванием из стека, т.е. используя некоторые символы γ и заменяя их магазине, а это невозможно без обработки γ. Однако мы можем удалить неиспользуемый вход, т.к. МПА не может прочитать входные символы, а затем восстановить их на входе, согласно принципу 3.

Теорема 6.2. Если $P = (\ldots)$ – МПА, и $(q, xw, \alpha) \vdash_P^* (p, yw, \beta)$, то верно, что $(q, x, \alpha) \vdash_P^* (p, y, \beta)$. СЛАЙД 13.

6.3 Языки автоматов с магазинной памятью

До сих пор демонстрировался допуск автоматом входной строки **по заключительному состоянию**. Однако существует еще один способ определения языка МПА, имеющий важное практическое значение. Для любого МПА можно определить язык, допускаемый **по пустому магазину**, т.е. это множество строк, приводящих МПА в начальной конфигурации к опустошению магазина. Эти определения эквиваленты в смысле того, что для языка L найдется МПА, допускающий его по заключительному состоянию тогда и только тогда, когда для L есть автомат, допускающий его по пустому магазину.

При этом для конкретных МПА языки, допускаемые по заключительному состоянию и пустому магазину, обычно различны, но, очевидно, могут быть преобразуемы друг в друга.

Пусть $P = (\ldots)$ – МПА, тогда $L(P)$ – это язык, допускаемый P по заключительному состоянию, который может быть определен для некоторого состояния q и произвольной строки в магазине α, так что $\{w \mid (q_0, w, Z_0) \vdash_P^* (q, \varepsilon, \alpha)\}$.

Таким образом, начиная в стартовой конфигурации с w на входе, МПА P прочитывает эту строку и достигает заключительного состояния. Содержимое магазина в этот момент имеет только иллюстративное значение (СЛАЙД 14).

Пример 37. Пусть у нас есть МПА P, который допускает язык L_{wwr}. Попробуем доказать что это утверждение верно (СЛАЙДЫ 15 и 16). Сначала покажем допускающее вычисление P. Если $x = ww^R$, то справедливы следующие отношения.

$(q_0, ww^R, Z_0) \vdash^* (q_0, w^R, w^R Z_0) \vdash (q_1, w^R, w^R Z_0) \vdash^* (q_1, \varepsilon, Z_0) \vdash (q_2, \varepsilon, Z_0)$.

Иными словами, МПА имеет возможность прочитать w на входе и записать ее символы в магазин в обратном порядке. Затем он спонтанно переходит в состояние q_1 и проверяет совпадение w^R на входе с такой же строкой в магазине, и в конце спонтанно переходит в состояние q_2. Достаточность утверждения доказана.

Далее замечаем, что единственный путь достижения заключительного состояния состоит в том, чтобы находиться в q_1 и иметь только маркер дна в магазине. Кроме этого, любое допускающее вычисление начинается в состоянии q_0, совершает один переход в q_1 и никогда снова не приходит в q_0. То есть следует найти условия, налагаемые на x, для которых $(q_0, x, Z_0) \vdash^* (q_1, \varepsilon, Z_0)$, ведь именно такие строки допускает наш МПА по заключительному состоянию. Покажем индукцией по длине x следующее более общее утверждение.

Если $(q_0, x, \alpha) \vdash^* (q_1, \varepsilon, \alpha)$, то x имеет вид ww^R.

Базисом доказательства является $x = \varepsilon$, то x имеет вид ww^R с $w = \varepsilon$. В этом случае заключение верно и утверждение истинно.

Индуктивной частью доказательства является строка $x = a_1 \ldots a_n$, где $n > 0$. Существуют два перехода, которые МПА может совершить из МО (q_0, x, α).

1. $(q_0, x, \alpha) \vdash (q_1, x, \alpha)$. Теперь P может только выталкивать элемент из магазина. Он должен вытолкнуть символ из магазина с чтением входного символа, и $|x|>0$. Значит, если $(q_1, x, \alpha) \vdash (q_1, \varepsilon, \beta)$, то строка β короче строки α, и не может равняться ей.

2. $(q_0, a_1 \ldots a_n, \alpha) \vdash (q_1, a_2 \ldots a_n, a_1\alpha)$. Теперь последовательность символов может закончиться в $(q_1, \varepsilon, \alpha)$, только в том случае, когда последний переход является выталкиванием $(q_1, a_n, a_1\alpha) \vdash (q_1, \varepsilon, \alpha)$. В этом случае должно выполняться $a_1 = a_n$. Еще нам известно, что $(q_0, a_2 \ldots a_n, a_1\alpha) \vdash^* (q_1, a_n, a_1\alpha)$. По теореме 6.2 символ a_n можно удалить из конца входа, т.к. он не используется. Значит, получаем $(q_0, a_2 \ldots a_{n-1}, a_1\alpha) \vdash^* (q_1, \varepsilon, a_1\alpha)$. Поскольку вход этой последовательности короче, чем n, то можно применить индуктивное предположение и заключить, что $a_2 \ldots a_{n-1}$ имеет вид yy^R для некоторого y. Поскольку $x = a_1yy^Ra_n$, и мы знаем, что $a_1 = a_n$, то делаем вывод, что $x = ww^R$, в частности w $= a_1y$.

Эти рассуждения доказывают, что x допускается только того, когда он равен ww^R для некоторого w. Значит, необходимость доказана. Достаточность была доказана выше. Мы пришли к полному доказательству.

Теперь определим для каждого МПА $P = (\ldots)$ множество $N(P) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon)\}$, где q – произвольное состояние. Это множество всех входов w, которые P может прочитать, одновременно опустошив магазин. СЛАЙД 17.

Пример 38. К сожалению, МПА из примера 35 никогда не опустошает свой магазин, поэтому $N(P) = \varnothing$. Однако небольшое изменение позволяет допускать тот же язык и по пустому магазину, и по заключительному состоянию. Вместо перехода $\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$ мы используем $\delta(q_1, \varepsilon, Z_0) = \{(q_2, \varepsilon)\}$. Теперь МПА выталкивает последний символ из магазина, когда допускает, поэтому $L(P) = N(P) = L_{wwr}$. СЛАЙД 18.

Предположим, P_N – это МПА, допускающий язык L по пустому магазину, P_F – это МПА, допускающий L по заключительному состоянию. Примем без доказательства следующую теорему.

Теорема 6.3. Если $L = N(P_N)$ для некоторого P_N, то существует такой МПА P_F , у которого $L = L(P_F)$. СЛАЙД 19.

Пример 39. Построим некорректно работающий МПА, обрабатывающий последовательности ключевых слов *if* и *else* в программе на языке С, где *i* означает *if*, а *e* – *else*. В любом префиксе программы количество *e* не может превышать *i*, т.к. в противном случае *e* нельзя сопоставить предшествующему *i*. Магазинный символ *Y* используется для подсчета разницы между текущими количествами просмотренных *i* и *e*. Наш МПА состоит из одного состояния, его диаграмма переходов представлена на СЛАЙДЕ 20.

Увидев *i*, МПА заталкивает *X*, а увидев *e* – выталкивает. Он начинает с одним *X* в магазине, то следует правилу, что если в магазине X^n, то символов *i* прочитано на *n*-1 больше, чем *i*, и входная последовательность недопустима в программе на С. Наш же МПА их допускает по пустому магазину.

Его формальное определение имеет вид

$$P_N = (\{q_0\}, \{i, e\}, \{X\}, \delta_N, q_0, Z_0, \varnothing)$$
$$\delta_N(q_0, i, X) = \{(q_0, XX)\} \qquad .$$
$$\delta_N(q_0, e, X) = \{(q_0, \varepsilon)\}$$

На основе этого автомата можно построить другой МПА P_F, который допускает этот же язык по заключительному состоянию. Его диаграмма переходов также показана на СЛАЙДЕ 21. Введены новые начальное и заключительное состояния.

Его формальное определение имеет вид

$$P_F = (\{q_0, q_1, q_2\}, \{i, e\}, \{X, Z_0\}, \delta_F, q_0, Z_0, \{q_2\})$$
$$\delta_F(q_0, \varepsilon, Z_0) = \{(q_1, XZ_0)\}$$
$$\delta_F(q_1, i, X) = \{(q_1, XX)\}$$
$$\delta_F(q_1, e, X) = \{(q_1, \varepsilon)\}$$
$$\delta_F(q_1, e, Z_0) = \{(q_2, \varepsilon)\}$$

Также примем без доказательства следующую теорему.

Теорема 6.4. Если $L = L(P_F)$ для некоторого P_F, то существует такой МПА P_N, у которого $L = N(P_N)$. СЛАЙД 21.

6.4 Детерминированные автоматы с магазинной памятью

Хотя МПА являются недетерминированными по определению, их детерминированный случай очень важен. В частности, многие синтаксические анализаторы в целом ведут себя как детерминированные МПА (далее – ДМПА), поэтому класс языков, допускаемый этими автоматами, углубляет понимание конструкций, пригодных для языков программирования.

Интуитивно понятно, что у ДМПА в любой ситуации нет возможности выбора переходов. А выбирать можно из двух вариантов. Если $\delta(q, a, X)$ содержит более одной пары, то МПА не является ДМПА, т.к. придется выбирать из двух пар. Если же $\delta(q, a, X)$ всегда одноэлементно, то возможность выбора остается – между чтением входного символа и совершением ε-перехода. Тогда МПА *P* определяется как ДМПА, если выполнены следующие условия:

1. $\delta(q, a, X)$ имеет не более одного элемента для всех *q* из *Q*, *a* из Σ или $a=\varepsilon$ и *X* из Γ.
2. Если $\delta(q, a, X)$ непусто для некоторого *a* из Σ, то $\delta(q, \varepsilon, X)$ должно быть пустым.

Пример 40.

Может показаться странным, но МПА из примера 35 не является ДМПА. Если мы слегка изменим язык путем помещения в него центрального маркера в виде символа *b* в середине слова, то получится язык $L_{wbwr} = \{wbw^R \mid w$ принадлежит $(0+1)^*\}$, который распознается ДМПА.

Этот автомат будет вталкивать 0 и 1 в магазин до появления маркера b. Затем ДМПА переходит в другое состояние, в котором сравнивает входные и магазинные символы, и выталкивает последние в случае их равенства. Находя его, он останавливается без допуска (по-другому – «умирает»). Его вход уже не может иметь вид wbw^R. Если удалением магазинных символов он достигает стартового символа, отмечающего дно магазина, он допускает свой вход. Этот автомат будет напоминать МПА из примера 35, но тот был недетерминированным. Автомат для языка L_{wbwr} приведен на СЛАЙДЕ 22 в виде диаграммы переходов.

Очевидно, что он – ДМПА. У него нет выбора перехода в одном и том же состоянии и при использовании одних и тех же входных и магазинных символов. Что касается выбора между использованием входного или пустого символов, то у нас единственным ε-переходом, который он совершает, является переход из q_1 в q_2 с Z_0 на вершине магазина. Однако в состоянии q_1 с Z_0 на вершине – других переходов нет.

РЯ и ДМПА

ДМПА допускают класс языков, который занимает промежуточное положение между РЯ и так называемыми контекстно-свободными языками (они будут рассмотрены в следующем разделе лекционного курса).

Теорема 6.5. Если L – РЯ, то $L = L(P)$ для некоторого ДМПА P. СЛАЙД 23.

Доказательство. По большому счету ДМПА позволяет сымитировать ДКА. МПА содержит некоторый символ Z_0 в магазине, т.к. он должен иметь магазина, но в действительность он игнорирует магазин, используя только состояние.

Пусть $A = (Q, \Sigma, \delta_A, q_0, F)$ – КА, построим МПА $P = (Q, \Sigma, \{Z_0\}, \delta_P, q_0, Z_0, F)$, определив функцию переходов $\delta_P(q, a, Z_0) = \{(p, Z_0)\}$ для всех состояний p и q из Q, при которых $\delta_A(q, a) = p$.

Мы утверждаем, что $(q_0, w, Z_0) \vdash_P^* (p, \varepsilon, Z_0)$ тогда и только тогда, когда $\hat{\delta}(q_0, w) = p$, т.е. P имитирует A, используя его состояние. Доказательства в обе стороны просты и проводятся индукцией по $|w|$, поэтому оставляем это в качестве самостоятельного упражнения. Поскольку и A, и P допускают, достигнув какого-либо состояния из F, то мы приходим к выводу, что их языки равны. Этого мы и добивались.

Если мы хотим, чтобы ДМПА допускал по пустому магазину, то обнаруживаем, что возможность по распознаванию весьма ограничены. Считается, что язык L имеет **префиксное свойство**, или **свойство префиксности**, если в L нет двух различных строк x и y, где x является префиксом y. СЛАЙД 24.

Пример 41.

Язык L_{wbwr} из примера 40 имеет префиксное свойство, т.е. в нем не может быть двух разных строк wbw^R и xbx^R, одна из которых является префиксом другой. Допустим, wbw^R является префиксом для xbx^R и $w \neq x$. Тогда w должна быть короче x. Значит, b в w приходится на позицию, в которой x имеет 0 или 1, а это противоречит нашему предположению относительно того, что wbw^R является префиксом для xbx^R. СЛАЙД 24.

Есть простые языки без префиксного свойства. Рассмотрим $\{0\}^*$, т.е. набор строк из 0, в том числе пустых. В этом языке есть пары строк, одна из которых является префиксом другой, так что у этого языка нет свойства префиксности. И в самом деле, из любых двух строк одна является префиксом другой, хотя это условие сильнее, чем то, которое нужно для отрицания префиксного свойства.

Надо отметить, что язык $\{0\}^*$ регулярен, поэтому неверно утверждать, что каждый РЯ есть $N(P)$ для некоторого ДМПА P.

Теорема 6.6. Язык L есть $N(P)$ для некоторого ДМПА P тогда и только тогда, когда L имеет префиксное свойство и L есть $L(P')$ для некоторого ДМПА P'. СЛАЙД 25

Доказательство.

Дадим лишь общую идею. Сначала доказывается, что если $L = N(P)$ для ДМПА P, то L обладает префиксным свойством. Затем нужно доказать, что если $L = N(P)$ для ДМПА P, то существует ДМПА P', для которого $L = L(P')$. Отсюда, если L обладает префиксным свойством, и $L = L(P')$ для P', то существует ДМПА P, для которого $L = N(P)$.

Литература к лекции 6

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