**Методический материал для лабораторной работы № 7 . Проектирование МП-автомата для КС-грамматики**

КС-языки задаются магазинными автоматами, которые являются расширением недетерминированного конечного автомата с ε-переходами. **Автомат с магазинной памятью** — это [конечный автомат](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82), который использует [стек](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA) для хранения состояний. Магазин (стек) –  
это бесконечное устройство хранения информации, работающее по принципу последний пришел – первый вышел (LIFO – last-in-first-out). Присутствие магазина  
означает, что в отличие от конечного автомата магазинный автомат может помнить бесконечное количество информации. Неформально магазинный автомат  
можно рассматривать следующим образом (рисунок 1 ). В чистом виде автоматы с магазинной памятью используются крайне редко. Обычно эта модель используется для наглядного представления отличия обычных конечных автоматов от [синтаксических грамматик](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0&action=edit&redlink=1). Реализация автоматов с магазинной памятью отличается от конечных автоматов тем, что текущее состояние автомата сильно зависит от любого предыдущего.

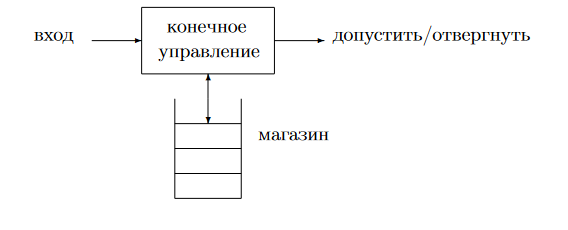


Рис. 1. Неформальное представление магазинного автомата

Конечное управление читает входные символы по одному. Магазинный автомат может читать символ с вершины магазина и совершать переход на основе  
текущего состояния, входного символа и символа на вершине магазина. Он также может выполнить спонтанный переход по символу ε. За один переход автомат  
совершает следующие действия.  
1) Читает и пропускает входной символ, используемый при переходе. Если в  
качестве входа используется ε, то входные символы не пропускаются.  
2) Переходит в новое состояние, которое может и не отличаться от предыдущего.  
3) Заменяет символ на вершине магазина некоторой цепочкой. Цепочкой может быть ε, что соответствует снятию с вершины магазина. Это может быть тот  
же символ, который был ранее на вершине магазина, т. е. магазин не изменяется.  
Автомат может заменить магазинный символ, что равносильно изменению вершины магазина без снятий и заталкиваний. Наконец, символ может быть заменен  
несколькими символами – это равносильно тому, что (возможно) изменяется символ на вершине, а затем туда помещаются один или несколько новых символов.

Существуют [детерминированные](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82) и [недетерминированные](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B4%D0%B5%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82) автоматы с магазинной памятью.

Для недетерминированных автоматов (в отличие от детерминированных) существует два эквивалентных критерия завершения работы:

1. пустой магазин,
2. достижение конечного состояния.

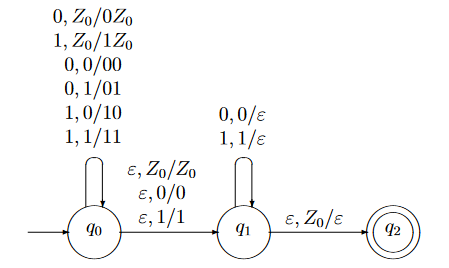
Детерминированный автомат завершает работу лишь тогда, когда достигает конечного состояния.

**Пример.** Рассмотрим язык палиндромов четной длины над символами 0 и  
1 (например, 010010, 110011, 1001) и дадим неформальное описание автомата,  
распознающего этот язык.  
1. Работа начинается в состоянии q0, представляющем догадку, что не достигнута середина входного слова. В этом состоянии символы читаются и их копии  
записываются в магазин.  
2. В любой момент можно предположить, что достигнута середина входного  
слова, т. е. левая половина слова находится в магазине. Этот выбор отмечается  
спонтанным переходом в состояние q1. Поскольку автомат недетерминированный,  
в действительности предполагаются обе возможности, т. е. можно также оставаться в состоянии q0 и продолжать читать входные символы и записывать их в  
магазин.  
3. В состоянии q1 входные символы сравниваются с символами на вершине  
магазина. Если они совпадают, то входной символ пропускается, магазинный удаляется, и работа продолжается. Если же они не совпадают, то предположение о  
середине слова неверно. Эта ветвь вычислений отбрасывается, хотя другие могут  
продолжаться и вести к тому, что слово допустимо.  
4. Если магазин опустошается и конец слова достигнут, то входное слово является палиндромом.

**Дадим теперь формальное определение магазинного автомата.**

**Магазинный автомат** – это семерка объектов M =< V, Q, Γ, ψ, q0, Z0, F >:  
V – конечное множество входных символов (как у конечного автомата);  
Q – конечное множество состояний (как у конечного автомата);  
Γ – конечный магазинный алфавит (не имеет конечноавтоматного аналога) –  
множество символов, которые можно помещать в магазин;  
ψ – функция переходов, управляющая поведением автомата: **ψ(x, Z, q) = (Z′, q′).**Здесь x ∈ ε∪X – символ входного алфавита либо пустое слово, Z ∈ Γ – магазиный  
символ, q ∈ Q – текущее состояние, q′ ∈ Q – новое состояние, γ ∈ Γ∗ – цепочка  
магазинных символов, замещающих Z на вершине магазина. Если γ = ε, то магазинный символ снимается, если γ = Z, магазин не меняется, если γ = Z1 . . . Zm, то Z заменяется на Zm, а символы Zm−1 . . . Z1 добавляются в магазин (Z1 теперь на вершине);  
q0 ∈ Q – начальное состояние;  
Z0 – начальный магазинный символ (маркер дна), в начале работы в магазине  
находится только этот символ;  
F – множество заключительных состояний

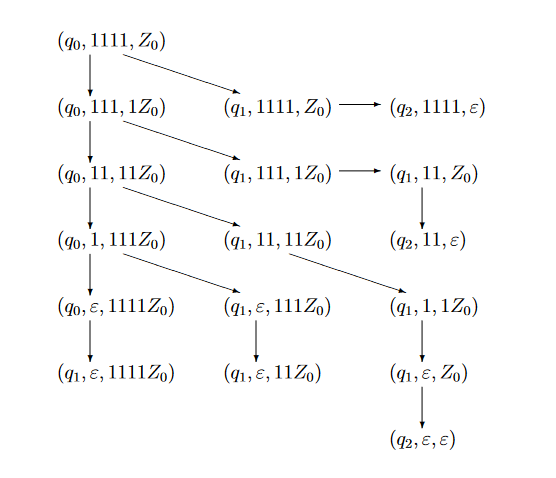
Пример. Автомат из предыдущего примера выглядит следующим образом.  
M =< {0, 1}, {q0, q1, q2}, {0, 1, Z0}, ψ, q0, Z0, {q2} > .  
Функция ψ определяется правилами:  
– ψ(0, Z0, q0) = (0Z0, q0), ψ(1, Z0, q0) = (1Z0, q0). Одно из этих правил применяется в начале работы, когда автомат находится в состоянии q0 и обозревает  
начальный символ Z0 на вершине магазина. Читается первый символ и помещается в магазин; Z0 остается под ним для отметки дна.  
– ψ(0, 0, q0) = (00, q0), ψ(0, 1, q0) = (01, q0), ψ(1, 0, q0) = (10, q0), ψ(1, 1, q0) =  
(11, q0). Эти правила позволяют оставаться в состоянии q0 и читать входные сим-  
волы, помещая каждый из них на вершину магазина над предыдущим верхним  
символом.  
– ψ(ε, Z0, q0) = (Z0, q1), ψ(ε, 0, q0) = (0, q1), ψ(ε, 1, q0) = (1, q1). Эти правила  
позволяют автомату спонтанно (без чтения входа) переходить из состояния q0 в  
состояние q1, не изменяя верхний символ магазина, каким бы он ни был.  
– ψ(0, 0, q1) = (ε, q1), ψ(1, 1, q1) = (ε, q1). В состоянии q1 входные символы про-  
веряются на совпадение с символами на вершине магазина. При совпадении по-  
следние выталкиваются.  
– ψ(ε, Z0, q1) = (ε, q2). Наконец, если обнаружен маркер дна магазина Z0 и  
автомат находится в состоянии q1, то обнаружен палиндром. Автомат переходит  
в заключительное состояние q2, магазин опустошается.  
Магазинный автомат также может быть представлен диаграммой. На дугах  
пишется (x, Z/γ) – вход и магазинный символ, по которому происходит переход,  
и через слэш – цепочка, замещающая верхний магазинный символ.

**Пример. Автомат из предыдущего примера имеет вид **

В отличие от конечного автомата, где для описания дальнейшего поведения  
автомата достаточно знать его состояние, для магазинного автомата значимым  
является и содержимое магазина. Полезно знать и непрочитанную часть входа.

Конфигурация магазинного автомата представляется тройкой (q, γ, w) где q –  
состояние, γ – содержимое магазина, w – непрочитанная часть входного слова.  
Определим отношение (непосредственной выводимости) следующим образом: если ψ(a, X, q) = (α, p), то ∀w ∈ V∗, β ∈ Γ∗ (q, aw, Xβ) (p, w, αβ) ((p, w, αβ) непосредственно выводима из (q, aw, Xβ)).  
Это отношение отражает идею того, что, прочитывая на входе символ a (возможно, ε), и заменяя X на вершине магазина цепочкой α можно перейти из состояния q в состояние p. Заметим, что оставшаяся часть входа w и содержимое магазина под его вершиной β не влияют в данный момент на действия автомата, они просто сохраняются и, возможно, будут влиять на события в дальнейшем. Любую конфигурацию вида (q0, w, Z0) называют начальной, а любую конфигурацию вида (qf, ε, ε) – заключительной. Обратим внимание, что заключительная  
конфигурация – это конфигурация с пустым магазином. Таким образом, множество всех заключительных конфигураций находится во взаимно-однозначном  
соответствии с множеством всех заключительных состояний автомата. Из заключительной конфигурации не может быть выведена ни одна конфигурация. Не  
заключительную конфигурацию магазинного автомата, из которой не выводима  
ни одна конфигурация, называют тупиковой.  
Выводом на множестве конфигураций магазинного автомата M называют  
последовательность C0, C1, . . . , Cn, . . . (конечную или бесконечную) таких его конфигураций, что ∀i ≥ 0: Ci ` Ci+1 (если Ci+1 существует).  
Конфигурацию C′ называют выводимой из конфигурации C, (C `∗ C′) если  
существует связывающий их вывод. В частности, любая конфигурация выводима  
сама из себя.  
Таким образом, понятие выводимости для конфигураций автомата полностью  
аналогично таковому понятию для грамматики.

**Пример.** Рассмотрим работу автомата из предыдущего примера для входа  
1111.

****

Здесь мы получили единственную цепочку, приводящую к заключительной  
конфигурации (q2, ε, ε), остальные заканчиваются тупиковыми конфигурациями  
(или мы не попали в заключительное состояние, или прочитан не весь вход, или  
магазин не пуст).  
Входное слово w называют допустимым словом магазинного автомата, если на  
множестве конфигураций автомата существует вывод, связывающий начальную  
конфигурацию (q0, w, z0) с заключительной конфигурацией (qf , ε, ε).  
Язык, допускаемый автоматом L(M ) – это множество его допустимых слов.

**От грамматики к автомату**

По данной грамматике G =< V, T, P, A > строится магазинный автомат, ими-  
тирующий ее левые порождения. Любую левовыводимую цепочку, которая не является терминальной, можно записать в виде vU α, где U – крайний слева нетерминал, v – цепочка терминалов слева от U , α – цепочка терминалов и переменных справа. U α называется остатком этой левовыводимой цепочки. У терминальной левовыводимой цепочки остатком является ε. Идея построения магазинного автомата по грамматике состоит в том, чтобы магазинный автомат имитировал последовательность левовыводимых цепочек, используемых в грамматике для порождения данной нетерминальной цепочки w. Остаток каждой цепочки U α появляется в магазине с переменной U на вершине. При этом v представлен прочитанными на входе символами, а символы цепочки w после v считаются непрочитанными.  
Предположим, что автомат находится в конфигурации (q, y, U α), представляющей левовыводимую цепочку vU α. Он угадывает правило грамматики, используемое для расширения U , скажем, U → β. Переход автомата состоит в том,  
что U на вершине магазина заменяется цепочкой β, и достигается конфигурация  
(q, y, βα). Состояние при этом не меняется, и вообще, у этого автомата всего одно  
состояние q.  
Теперь (q, y, βα) может не быть представлением следующей левовыводимой  
цепочки, поскольку β может начинаться с терминальных символов. Также β может вообще не содержать переменных, а α может начинаться с терминалов. Все  
терминалы в начале цепочки βα нужно удалить до появления следующей переменной на вершине магазина. Эти терминалы сравниваются со следующими входными символами для того, чтобы убедиться, что наши предположения о левом  
порождении входной цепочки w правильны, и правило U → β выбрано верно; в  
противном случае эта ветвь вычислений отбрасывается.  
Если таким способом нам удается угадать левое порождение w, то в конце мы  
дойдем до левовыводимой цепочки w. В этот момент все символы в магазине или  
расширены (если это переменные) или совпали с выходными (если это терминалы). Магазин пуст, автомат допускает слово w по пустому магазину.  
Дадим формальное описание алгоритма.

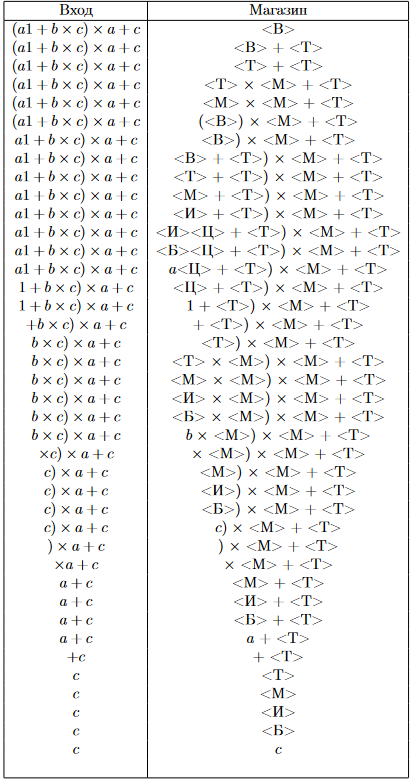
**Пример.**Построить автомат, допускающий язык, порожденный грамматикой с продукциями  
S → aSbb | a.  
**Преобразуем грамматику к нормальной форме Грейбах**:  
S → aSA | a,  
A→bB,  
B→b.  
Соответствующий автомат будет иметь три состояния  
{q0, q1, q2}, q0 – начальное, q2 – финальное состояние. Вначале за-  
носим в стек начальный символ грамматики S командой  
θ(q0, ε, z) = {(q1, Sz)}.  
Продукция S → aSA моделируется автоматом посредством извлечения S из стека и заменой на SA, когда читается входной символ a. Аналогично продукция S → a соответствует извлечению символа S из стека в результате чтения символа a из входной строки. Таким образом, обе эти продукции представляются в НМА соотношением  
θ(q1, a, S) = {(q1, SA), (q1, ε)}.  
Аналогичным образом остальным продукциям соответствуют команды:  
θ(q1, b, A) = {(q1, B)},  
θ(q1, b, B) = {(q1, ε)}.  
Появление в вершине стека начального символа стека z означает  
завершение вывода, и НМА переходит в заключительное состояние:  
θ(q1, ε, z) = {(q2, ε)}

**Алгоритм построения магазинного автомата по контекстно-свободной  
грамматике**

Пусть G =< V, T, P, A > – КС-грамматика. Построим магазинный автомат  
M =< T, {q}, V, ψ, q, S >, который допускает L(G) по пустому магазину. Функция  
переходов ψ определяется следующим образом:  
– ψ(ε, U, q) = {(β, q)|U → β ∈ P }, ∀U ∈ V \ T ;  
– ψ(x, x, q) = {(ε, q)}, ∀x ∈ T ;  
других правил перехода нет.  
Пример. Порождающая грамматика для арифметических выражений, ис-  
пользующих операции сложения и умножения, скобки и идентификаторы.  
<Выражение> → <Терм>  
<Выражение> → <Выражение> + <Терм>  
<Терм> → <Множитель>  
<Терм> → <Терм> × <Множитель>  
<Множитель> → <Идентификатор>  
<Множитель> → (<Выражение>)  
<Идентификатор> → <Буква>

<Идентификатор> → <Идентификатор><Буква>  
<Идентификатор> → <Идентификатор><Цифра>  
<Буква> → a . . . <Буква> → z  
<Цифра> → 0 . . . <Цифра> → 9  
Сократим запись:  
<В> → <Т>  
<В> → <В> + <Т>  
<Т> → <М>  
<Т> → <Т> × <М>  
<М> → <И>  
<М> → (<В>)  
<И> → <Б>  
<И> → <И><Б>  
<И> → <И><Ц>  
<Б> → a . . . <Б> → z  
<Ц> → 0 . . . <Ц> → 9  
Преобразуем эту грамматику в магазинный автомат. Входной алфавит X –  
это все терминальные символы грамматики, т. е.  
V = {a, . . . , z, 0, . . . , 9, (, ), +, ×}.  
Магазинный алфавит Γ – это терминальные символы и переменные:  
Γ = V ∪ {<В>,<Т>,<М>,<И>,<Б>,<Ц>}.  
Начальный символ Z0=<В>.  
Функция переходов определяется следующим образом:  
1) ψ(ε, <В>, q) = {(<Т>, q), (<В> + <Т>, q)};  
2) ψ(ε, <Т>, q) = {(<М>, q), (<Т> × <М>, q)};  
3) ψ(ε, <М>, q) = {(<И>, q), (<(В)>, q)};  
4) ψ(ε, <И>, q) = {(<Б>, q), (<И><Б>, q), (<И><Ц>, q)};  
5) ψ(ε, <Б>, q) = {(a, q), . . . , (z, q)};  
6) ψ(ε, <Ц>, q) = {(0, q), . . . , (9, q)};  
7) ψ(a, a, q) = (ε, q), . . . , ψ(z, z, q) = (ε, q);  
8) ψ(0, 0, q) = (ε, q), . . . , ψ(9, 9, q) = (ε, q);  
9) ψ((, (, q) = (ε, q);  
10) ψ(), ), q) = (ε, q);  
11) ψ(+, +, q) = (ε, q);  
12) ψ(×, ×, q) = (ε, q).  
Здесь первые шесть переходов появились по первому правилу алгоритма, последние шесть – по второму. Других переходов в автомате нет.

Прочитаем слово (a1 + b × c) × a + c. Работу автомата представим в виде таблицы, в которой представлена конфигурация: остаток строки и состояние магазина.  
Состояние автомата всегда одно и тоже.



Автомат моделирует левый вывод

<В> ⇒ <В> + <T> ⇒ <Т> + <T> ⇒ <Т> × <M> + <T> ⇒  
<М> × <M> + <T> ⇒ (<В>) × <M> + <T> ⇒

(<В> + <T>) × <M> + <T> ⇒ (<В> + <T>) × <M> + <T> ⇒  
(<Т> + <T>) × <M> + <T> ⇒ (<М> + <T>) × <M> + <T> ⇒  
(<И> + <T>) × <M> + <T> ⇒ (<И><Ц> + <T>) × <M> + <T> ⇒  
(<Б><Ц> + <T>) × <M> + <T> ⇒ (a<Ц> + <T>) × <M> + <T> ⇒  
(a1 + <T>) × <M> + <T> ⇒ (a1 + <М>) × <M> + <T> ⇒  
(a1 + <И>) × <M> + <T> ⇒ (a1 + <Б>) × <M> + <T> ⇒  
(a1 + b) × <M> + <T> ⇒ (a1 + b) × <И> + <T> ⇒  
(a1 + b) × <Б> + <T> ⇒ (a1 + b) × c + <T> ⇒ (a1 + b) × c + <М> ⇒  
(a1 + b) × c + <И> ⇒ (a1 + b) × c + <Б> ⇒ (a1 + b) × c + a.