

# Конечные автоматы

1

**Романенко Владимир Васильевич,  
к.т.н., доцент каф. АСУ ТУСУР**

# Формальное описание ДМПА

2

Конечный автомат с магазинной памятью (ДМПА):

$$P = (Q, \Sigma, \Gamma, \delta, q_0, \gamma_0, F),$$

где

- $Q$  – конечное множество состояний;
- $\Sigma$  – конечное множество входных символов (алфавит);
- $\Gamma$  – конечный алфавит магазинных символов;
- $\delta$  – функция переходов, отображение множества  $Q \times (\Sigma \cup \{e\} \cup \{\perp\}) \times \Gamma$  во множество  $Q \times \Gamma^*$ ;
- $q_0 \in Q$  – начальное состояние;
- $\gamma_0 \in \Gamma^*$  – символ (или цепочка символов), находящийся в магазине в начальный момент (начальный символ);
- $F \subseteq Q$  – множество заключительных состояний.

# Формальное описание ДМПА

3

Способы записи функции переходов:

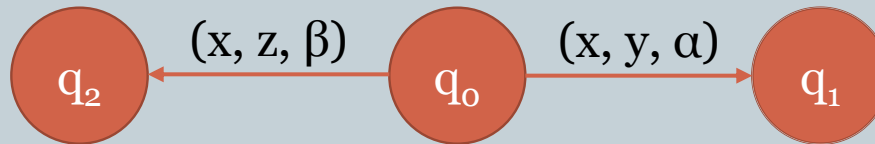
- В виде отображения:

$$\delta(q_0, x, y) = (q_1, \alpha), \delta(q_0, x, z) = (q_2, \beta), \dots$$

- В виде множества:

$$\delta = \{((q_0, x, y), (q_1, \alpha)), ((q_0, x, z), (q_2, \beta)) \dots\}$$

- В виде графа:



# Формальное описание ДМПА

4

Способы записи функции переходов:

- В виде таблицы:

	<b>х, у</b>	<b>х, z</b>
$q_0$	$q_1, \alpha$	$q_2, \beta$
$q_1$		
$q_2$		

# Формальное описание ДМПА

5

## Таблица переходов ДМПА:

	$a_1, z_1$	$a_2, z_2$	...	$\perp, z_n$
$q_0$	$\delta(q_0, a_1, z_1)$	$\delta(q_0, a_2, z_2)$	...	$\delta(q_0, \perp, z_n)$
$q_1$	$\delta(q_1, a_1, z_1)$	$\delta(q_1, a_2, z_2)$	...	$\delta(q_1, \perp, z_n)$
$q_2$	$\delta(q_2, a_1, z_1)$	$\delta(q_2, a_2, z_2)$	...	$\delta(q_2, \perp, z_n)$
...	...	...	...	...

$\delta(q, a, z) :$

- $(q', \beta);$
- $HALT (a = \perp, q \in F);$
- $ERROR.$

# Формальное описание ДМПА

6

## Возможные ситуации:

- $a = e$  (в этом случае символ из входной цепочки не считывается, т.е. не важно, на каком символе стоит считывающая головка, и головка при переходе к следующему состоянию не перемещается);
- $z = e$  (в этом случае символ из стека не извлекается, т.е. не важно, какой символ расположен на вершине стека);
- $z = \emptyset$  (в этом случае переход возможен лишь в том случае, если стек пуст);
- $\beta = e$  (в этом случае новые символы на стек не помещаются).

Если  $z = \beta = e$ , то переход осуществляется без участия стека.

Если  $z = \beta = e$  для всех переходов ДМПА, то имеем обычный ДКА.

# Определение функции переходов

7

1. Проще всего сначала построить граф переходов, а потом по описанным выше правилам преобразовать его в таблицу переходов.
2. Построение графа начинается с начального состояния  $q_0$ . Если начальное состояние может являться также и конечным, помечаем это двойной границей окружности.
3. Для каждого состояния графа  $q_i$  определяем, есть ли из данного состояния такие переходы  $(a, z, \beta)$ , которые соответствуют допустимому символу  $a$  из входной цепочки, допустимому символу  $z$  на вершине стека и при этом требуется помещение на стек цепочки  $\beta$ , которые пока еще отсутствуют в графе. Если есть, то проверяем, ведет ли данный переход в уже имеющееся состояние  $q_k$ . Если да, то добавляем в граф только новый переход  $(a, z, \beta)$  из состояния  $q_i$  в состояние  $q_k$ . Если нет, то добавляем в граф новое состояние  $q_j$  и переход  $(a, z, \beta)$  в него из состояния  $q_i$ . Если новое состояние может являться конечным, помечаем это двойной границей окружности.
4. Если в процессе выполнения шага 3 в графе появились новые состояния или переходы, возвращаемся на шаг 3, иначе граф переходов построен.

# Определение функции переходов

8

Примечание 1. В ДМПА конечное состояние может быть достижимо только при определенном состоянии стека. В таблице это показать легко:

	...	$\perp, z$
...	...	...
$q_i$	...	HALT
...	...	...

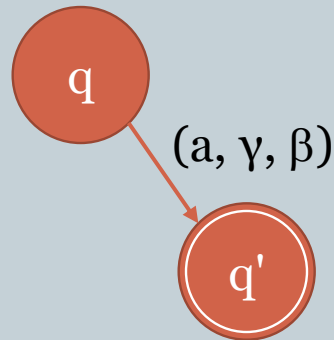
При этом, если  $z = \emptyset$ , то это равносильно требованию пустоты стека (т.е. разбор может быть успешно завершён лишь в том случае, если стек пуст). Если  $z = e$ , то разбор завершается при любом содержимом стека (т.е. успешное завершение разбора возможно при любом содержимом стека).



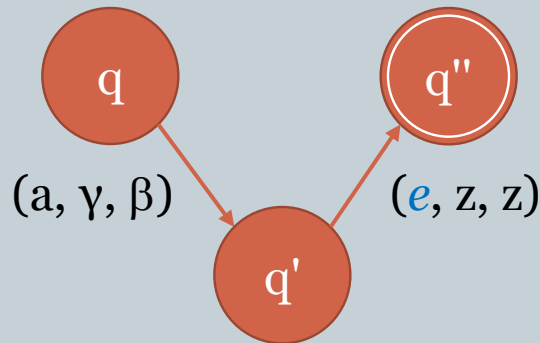
# Определение функции переходов

9

В графе это показать сложнее. По умолчанию, в конечном состоянии разбор завершается при любом содержимом стека:



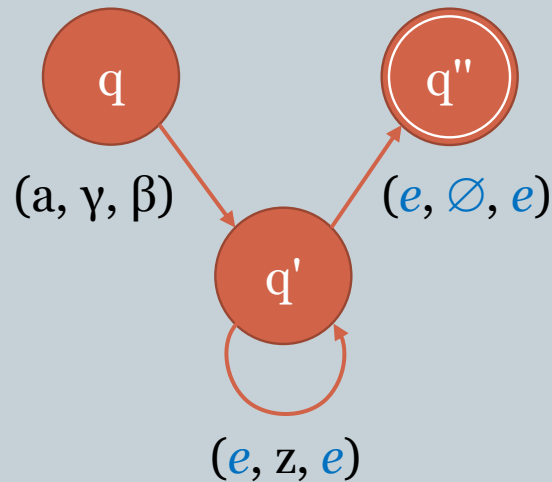
Если это не так, то вводится дополнительное искусственное состояние:



# Определение функции переходов

10

Если таких символов на стеке несколько, можно извлечь их перед завершением разбора:



Примечание 2. Если это более удобно, можно в реализации ДМПА положить, что разбор по умолчанию успешно завершается только при пустом стеке.

# Запуск ДМПА

11

Основные понятия:

- конфигурация ДМПА  $P$

$$(q, \alpha, \gamma) \in Q \times \Sigma^* \times \Gamma^*;$$

- начальная конфигурация –  $(q_0, \alpha, \gamma_0)$ , где  $\alpha \in \Sigma^*$ ;
- заключительная конфигурация –  $(q, \perp, \gamma)$ , где  $q \in F, \gamma \in \Gamma^*$  (обычно  $\gamma = e$ );
- такт работы ДМПА  $P$  при  $\delta(q, a, z) = (q', \beta)$ , где  $q, q' \in Q, a \in \Sigma \cup \{e\} \cup \{\perp\}, z \in \Gamma, \beta \in \Gamma^*$ :

$$(q, a\alpha, z\gamma) \Rightarrow (q', \alpha, \beta\gamma)$$

# Запуск ДМПА

12

Частные случаи:

- $a = e$  ( $e$ -такт):

$$(q, a_1 a_2 \dots a_n \perp, z_1 z_2 \dots z_m) \Rightarrow (q', a_1 a_2 \dots a_n \perp, \beta z_2 \dots z_m)$$

- $z = e$ :

$$(q, a_1 a_2 \dots a_n \perp, z_1 z_2 \dots z_m) \Rightarrow (q', a_2 \dots a_n \perp, \beta z_1 z_2 \dots z_m)$$

- $z = \emptyset$ :

$$(q, a_1 a_2 \dots a_n \perp, e) \Rightarrow (q', a_2 \dots a_n \perp, \beta)$$

- $\beta = e$ :

$$(q, a_1 a_2 \dots a_n \perp, z_1 z_2 \dots z_m) \Rightarrow (q', a_2 \dots a_n \perp, z_2 \dots z_m)$$

# Алгоритм работы ДМПА

13

Пусть  $\alpha = a_1 a_2 \dots a_n \perp$  – входная цепочка,  $M$  – магазин (стек). Тогда:

1. Полагаем  $M \leftarrow \gamma_0$ ,  $q \leftarrow q_0$ ,  $k \leftarrow 1$ .
2. Ищем  $\delta(q, a, z)$ , где:  $a = a_k$  или  $a = e$ ;  $z$  – элемент на вершине магазина  $M$ , либо  $z = \emptyset$  и магазин  $M$  пуст, либо  $z = e$ .
3. Если  $\delta(q, a, z)$  не определена, то ошибка в позиции  $k$ . Если значений  $\delta(q, a, z)$  несколько – таблица переходов построена неверно. Если  $\delta(q, a, z) = (q', \beta)$ , то:
  - 3.1. Переходим в новое состояние  $q \leftarrow q'$ .
  - 3.2. Если  $a \neq e$  и  $a \neq \perp$ , то  $k \leftarrow k + 1$  (переходим к следующему символу).
  - 3.3. Если  $z \neq e$  и  $z \neq \emptyset$ , то  $z \leftarrow M$  (т.е. извлекаем элемент с вершины стека).
  - 3.4. Если  $\beta \neq e$ , то  $M \leftarrow \beta$  (помещаем элемент в стек).
4. Если  $\delta(q, a, z) = HALT$ , то разбор успешно завершен.
5. Если  $\delta(q, a, z) = ERROR$ , то имеем во входной цепочке синтаксическую ошибку в позиции  $k$ .
6. Иначе возврат на шаг 2.

# Пример построения ДМПА

14

Пример. Рассмотрим язык  $L$ , описывающий обращение к элементу массива. При этом размерность массива может быть любой, а в качестве индексов можно использовать целые константы  $i \geq 0$ , а также идентификаторы, в т.ч. элементы других массивов. Идентификатор начинается с латинской буквы, после которой могут следовать другие буквы и цифры. Индексы заключаются в квадратные скобки и отделяются друг от друга запятыми. Например:

- `a[1];`
- `a2[5,b[2],z];`
- `mas[x[4],y[4]]` и т.д.

# Пример построения ДМПА

15

Выполним анализ языка:

1. В алфавит войдут буквы латинского алфавита, цифры, квадратные скобки, запятая. Теоретически, буквы могут быть как большими, так и маленькими, но для упрощения в примере будем использовать только маленькие буквы. Также в реальных языках программирования для отделения индексов и других синтаксических конструкций друг от друга можно использовать пробелы и другие символы-разделители, но для упрощения языка мы их рассматривать не будем.

# Пример построения ДМПА

16

2. В магазин будем помещать какой-либо символ, встречая открывающую квадратную скобку. Встречая закрывающую квадратную скобку, будем этот символ из магазина извлекать. Таким образом, мы сможем отследить парность открывающих и закрывающих скобок. Соответственно, в начальный момент времени стек должен быть пустым, т.к. никакие скобки ещё не были считаны. А в конечном состоянии стек должен быть пуст, т.к. любая открывающая скобка должна встретить соответствующую ей закрывающую скобку. Символ, помещаемый на стек, может быть любым, пусть это будет «+».



# Пример построения ДМПА

17

Получили следующие компоненты ДМПА:

- $\Sigma = \{a-z, 0-9, [, ], ,\};$
- $\Gamma = \{+\};$
- $q_0 = q_0;$
- $\gamma_0 = e.$

Также мы определили, что успешное завершение разбора возможно только при пустом стеке.

Далее определим множества  $Q, \delta, F$ .

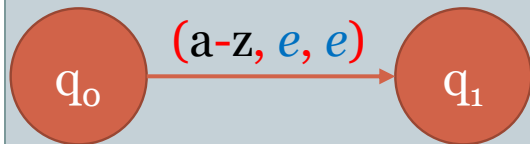
# Определение функции переходов

18

$q_0$

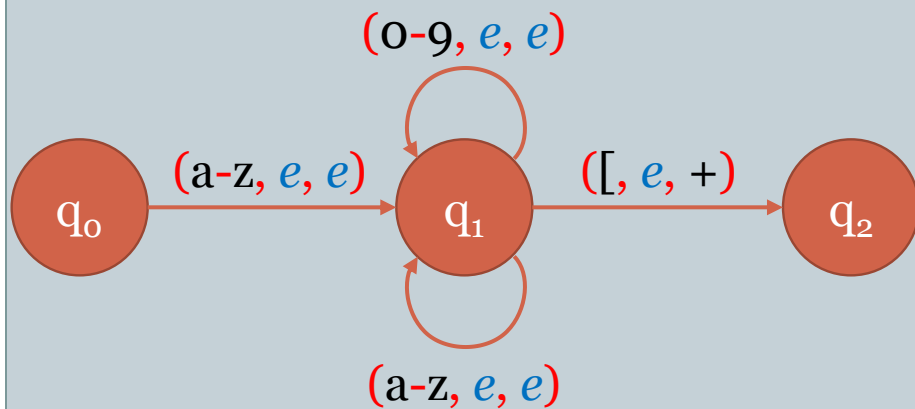
# Определение функции переходов

19



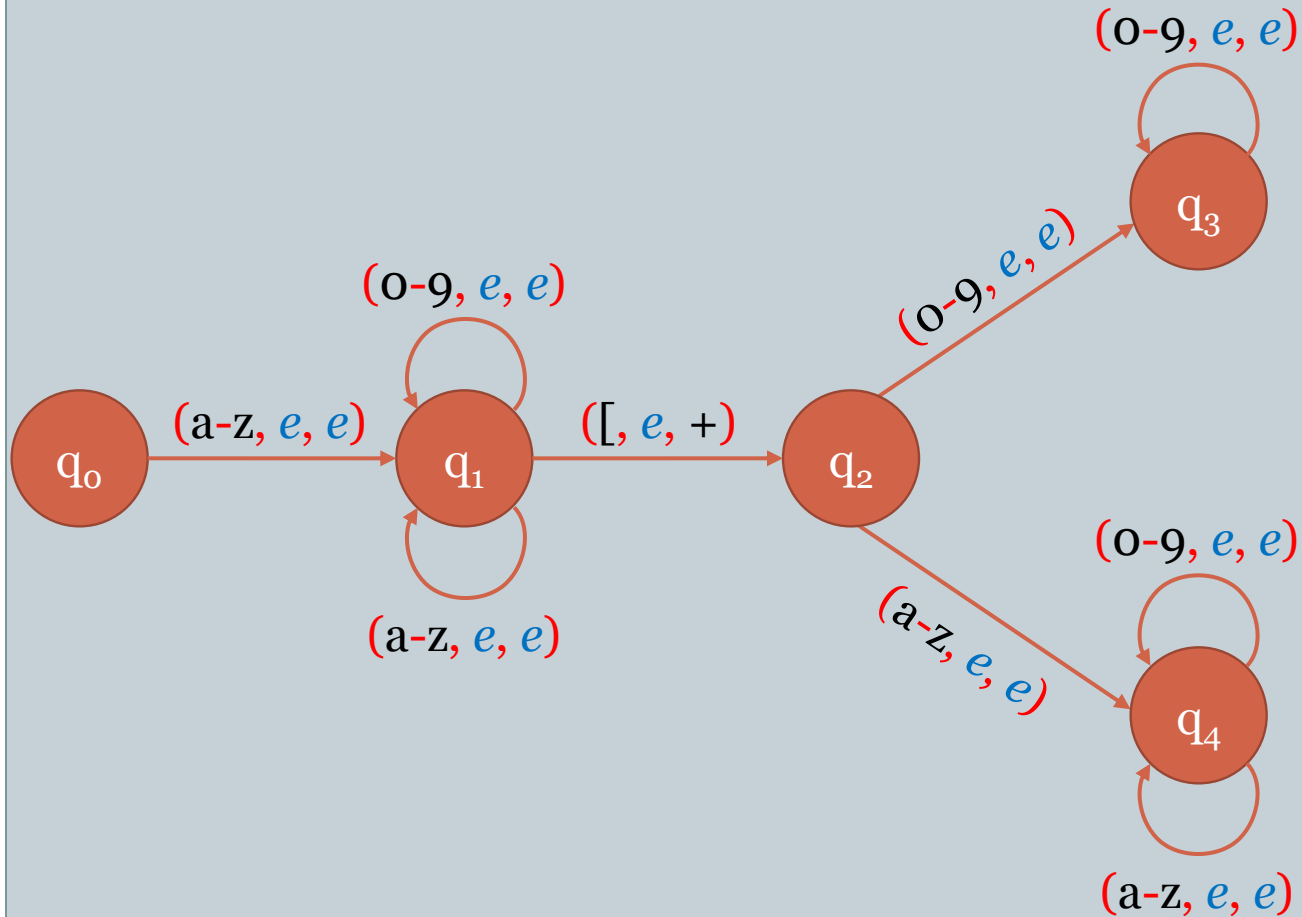
# Определение функции переходов

20



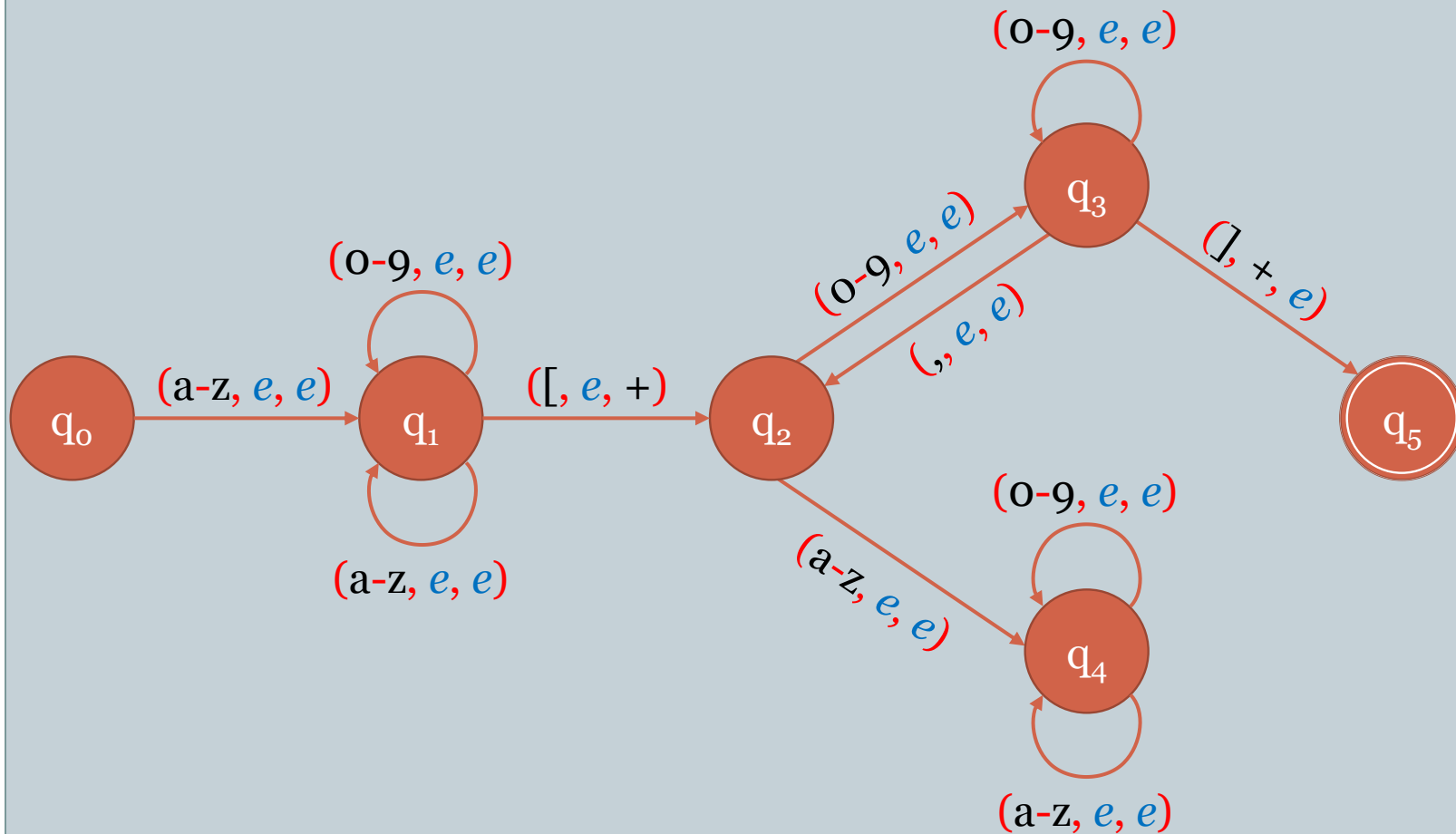
# Определение функции переходов

21



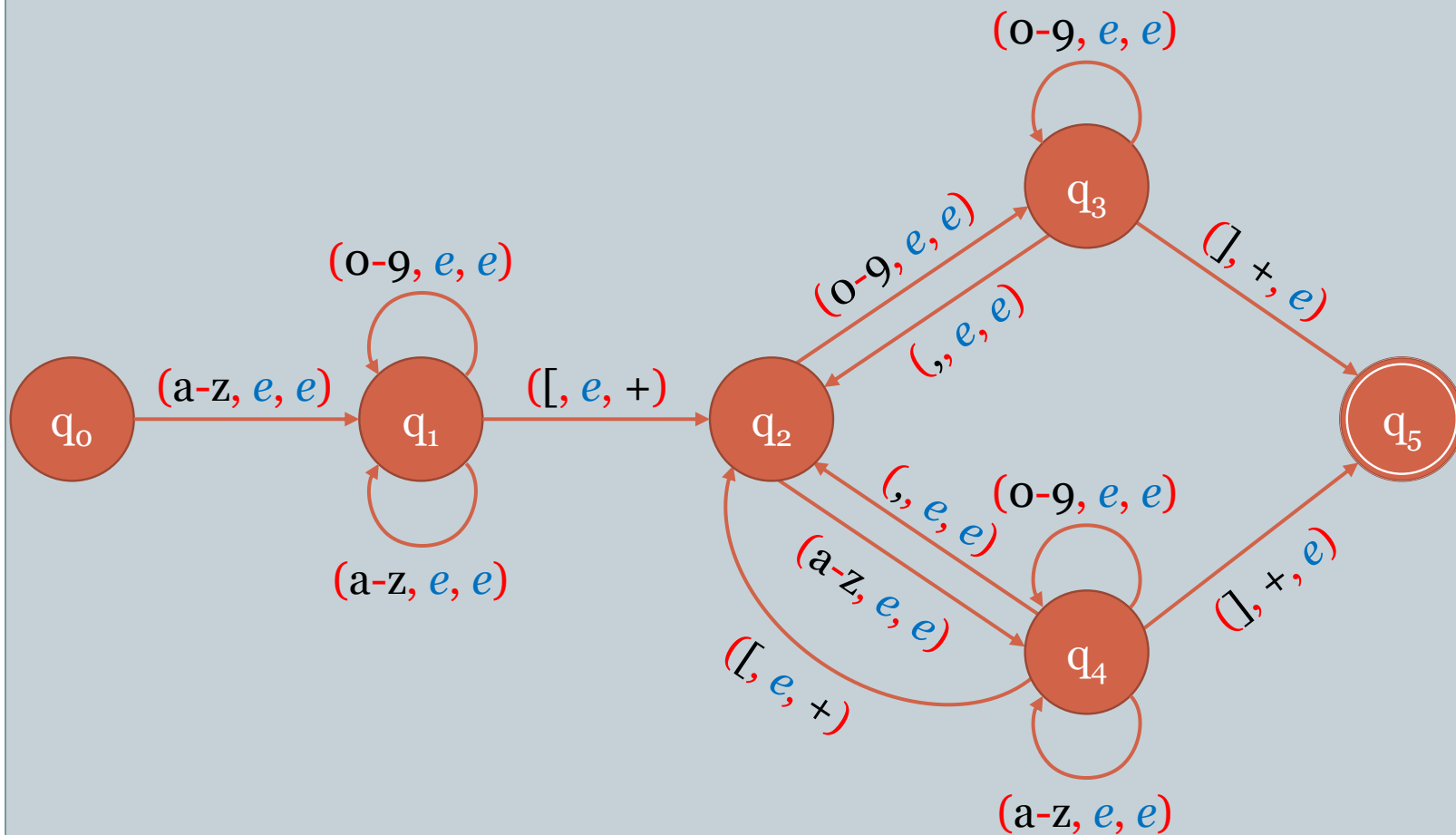
# Определение функции переходов

22



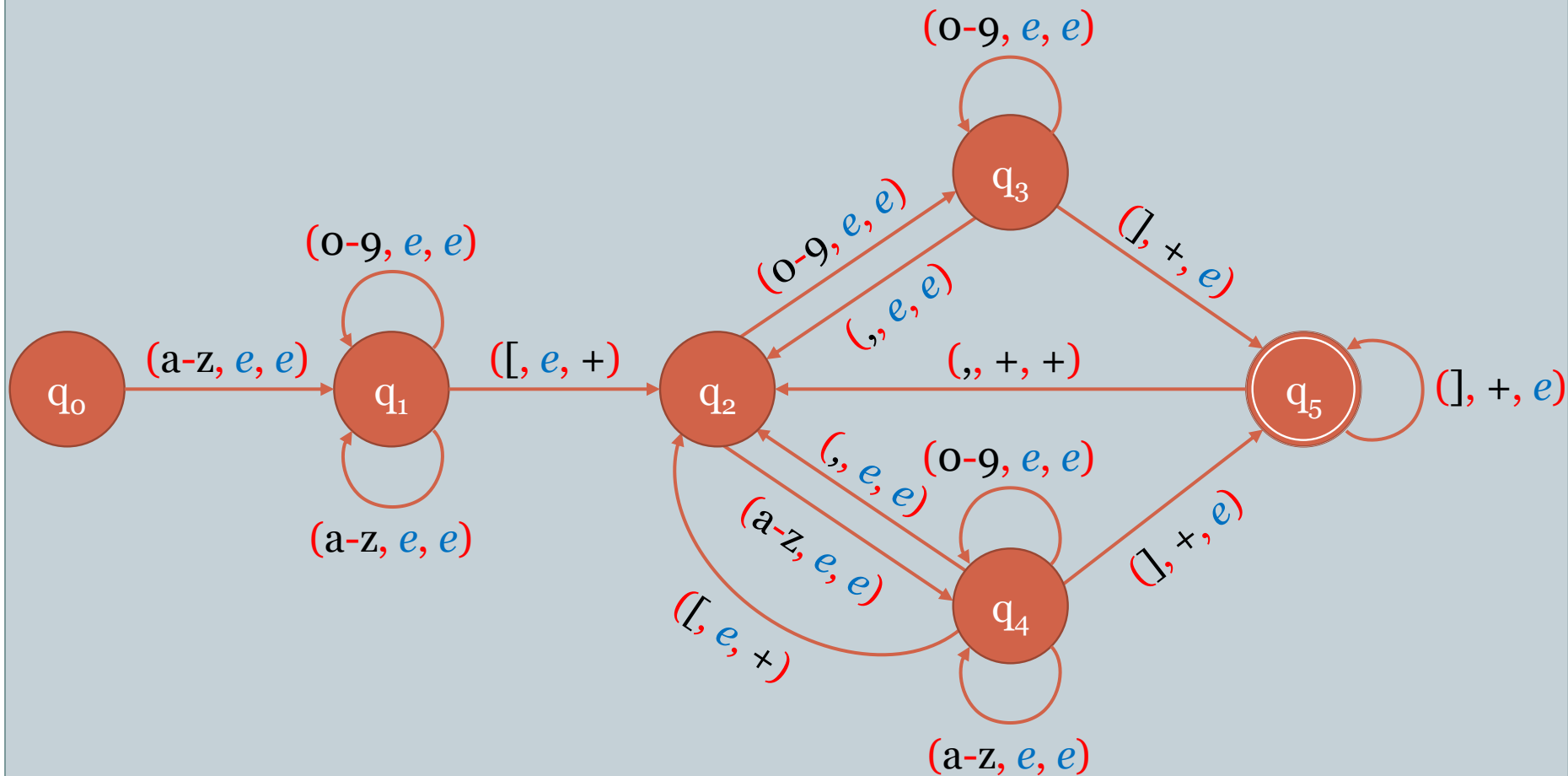
# Определение функции переходов

23



# Определение функции переходов

24





# Определение функции переходов

25

Табличное представление функции переходов:

	a-z, $e$	0-9, $e$	[, $e$	], +	,, $e$	,, +	$\perp$ , $\emptyset$
$q_0$	$q_1, e$						
$q_1$	$q_1, e$	$q_1, e$	$q_2, +$				
$q_2$	$q_4, e$	$q_3, e$					
$q_3$		$q_3, e$		$q_5, e$	$q_2, e$		
$q_4$	$q_4, e$	$q_4, e$	$q_2, +$	$q_5, e$	$q_2, e$		
$q_5$				$q_5, e$		$q_2, +$	HALT

В 5-м столбце стек не может быть пуст, поэтому можно объединить столбцы №5 и №6.

# Определение функции переходов

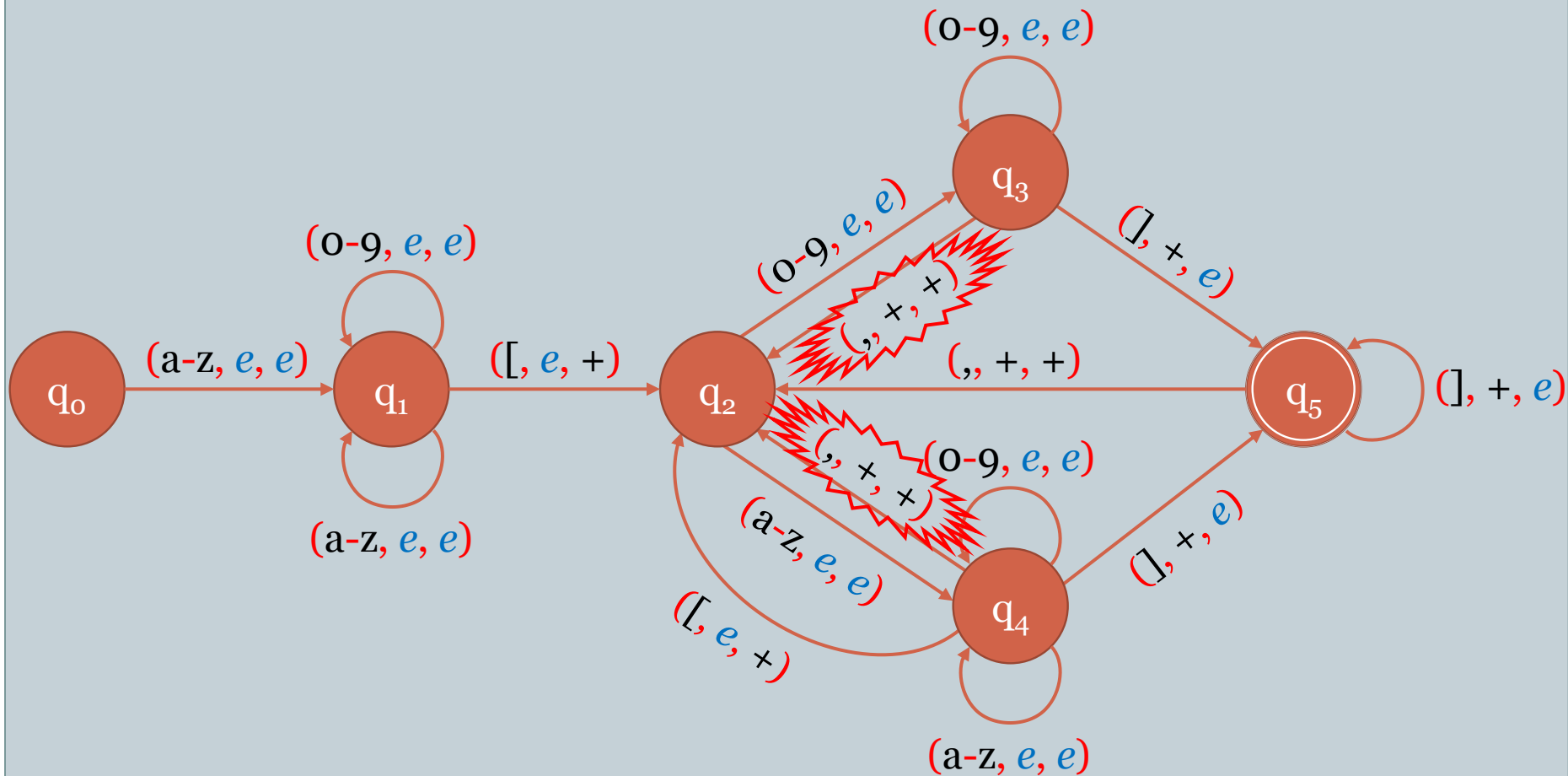
26

Табличное представление функции переходов:

	a-z, <i>e</i>	0-9, <i>e</i>	[, <i>e</i>	], +	,, +	⊥, ∅
q <sub>0</sub>	q <sub>1</sub> , <i>e</i>					
q <sub>1</sub>	q <sub>1</sub> , <i>e</i>	q <sub>1</sub> , <i>e</i>	q <sub>2</sub> , +			
q <sub>2</sub>	q <sub>4</sub> , <i>e</i>	q <sub>3</sub> , <i>e</i>				
q <sub>3</sub>		q <sub>3</sub> , <i>e</i>		q <sub>5</sub> , <i>e</i>	q <sub>2</sub> , +	
q <sub>4</sub>	q <sub>4</sub> , <i>e</i>	q <sub>4</sub> , <i>e</i>	q <sub>2</sub> , +	q <sub>5</sub> , <i>e</i>	q <sub>2</sub> , +	
q <sub>5</sub>				q <sub>5</sub> , <i>e</i>	q <sub>2</sub> , +	HALT

# Определение функции переходов

27



# Пример построения ДМПА

28

При этом на графе переходов ДМПА полагается, что для успешного завершения разбора в состоянии  $q_5$  стек должен быть пуст.

Таким образом,

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$ ;
- $\delta = \{((q_0, a-z, e), (q_1, e)), ((q_1, a-z, e), (q_1, e)), ((q_1, 0-9, e), (q_1, e)), ((q_1, [, e), (q_2, +)), ((q_2, 0-9, e), (q_3, e)), ((q_2, a-z, e), (q_4, e)), ((q_3, 0-9, e), (q_3, e)), ((q_3, ,, +), (q_2, +)), ((q_3, ], +), (q_5, e)), ((q_4, a-z, e), (q_4, e)), ((q_4, 0-9, e), (q_4, e)), ((q_4, ,, +), (q_2, +)), ((q_4, [, e), (q_2, +)), ((q_4, ], +), (q_5, e)), ((q_5, ,, +), (q_2, +)), ((q_5, ], +), (q_5, e)), ((q_5,  $\perp$ ,  $\emptyset$ ), **HALT**)\}$ ;
- $F = \{q_5\}$ .

# Проверка ДМПА

29

Как проверить, что ДМПА описан правильно?

1. Построить дерево вывода. Если дерево содержит все правильные цепочки языка, и не содержит неправильные цепочки, то автомат описан правильно.
2. Осуществить запуск ДМПА. Если для всех правильных цепочек разбор окончится символом **HALT**, а всех неправильных — символом **ERROR**, то автомат описан правильно.

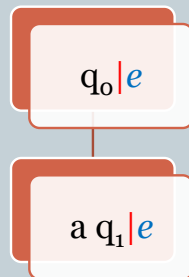
# Дерево вывода ДМПА

30

$q_0|e$

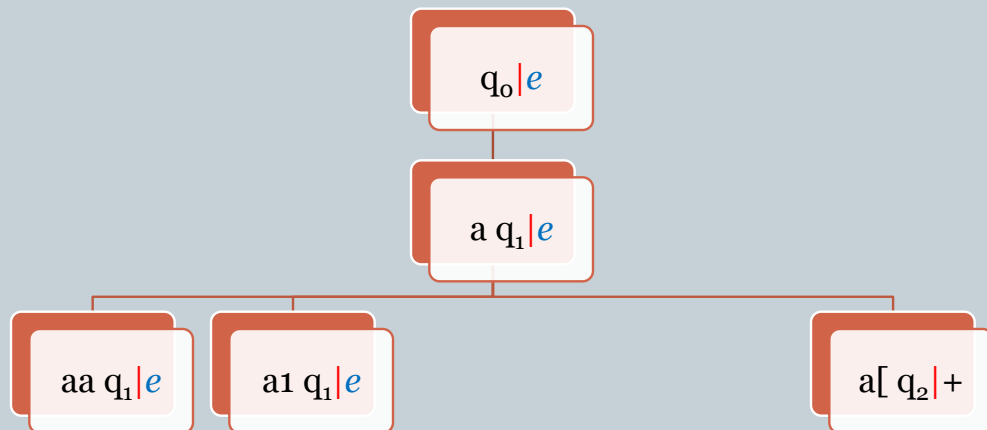
# Дерево вывода ДМПА

31



# Дерево вывода ДМПА

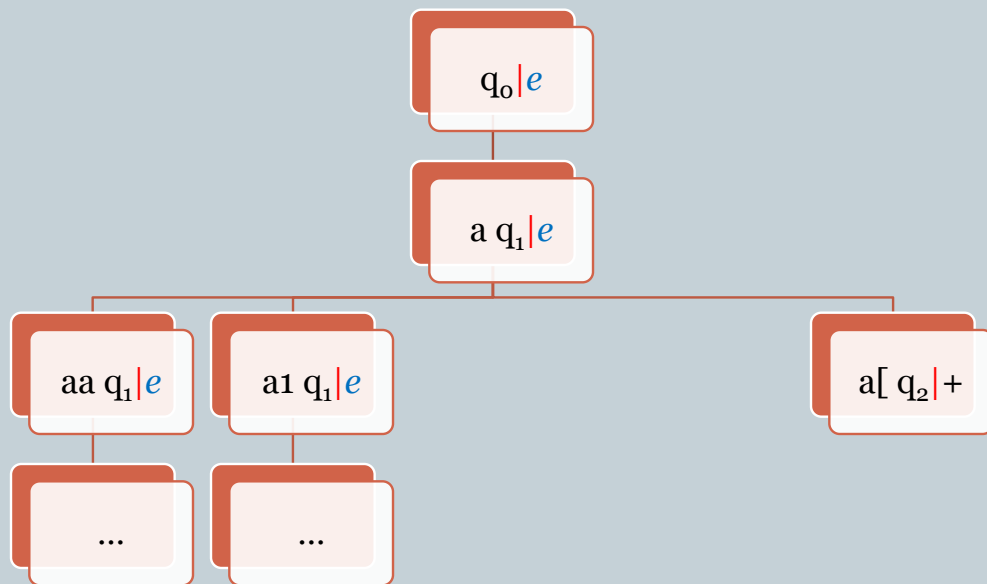
32





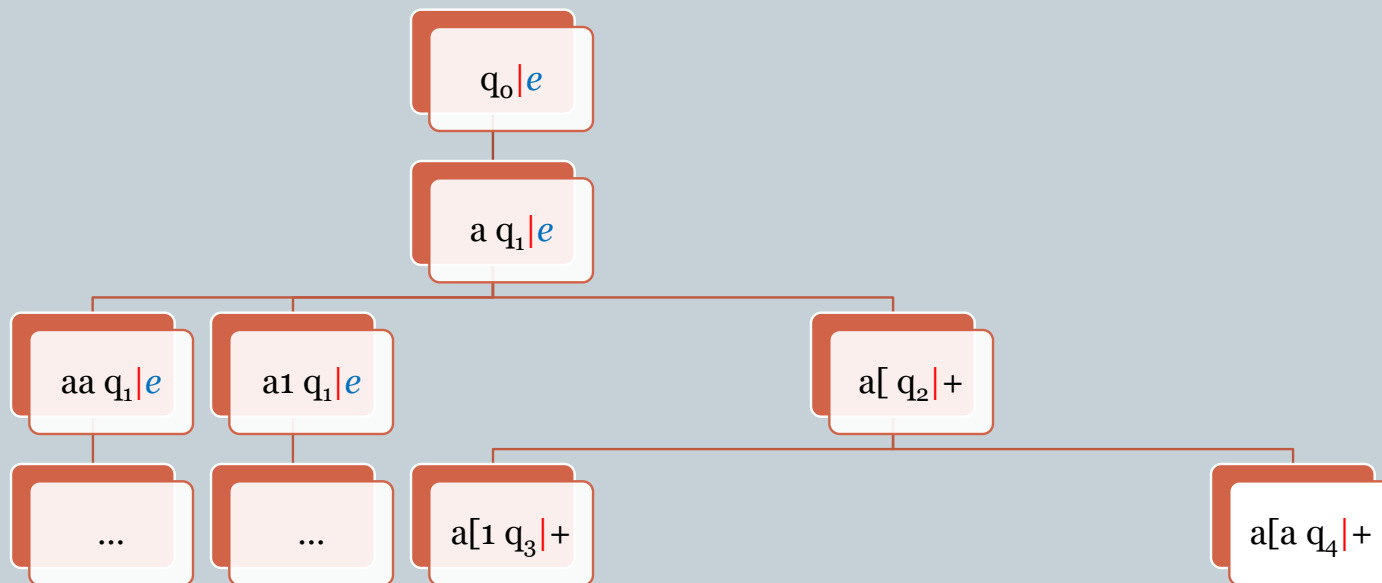
# Дерево вывода ДМПА

33



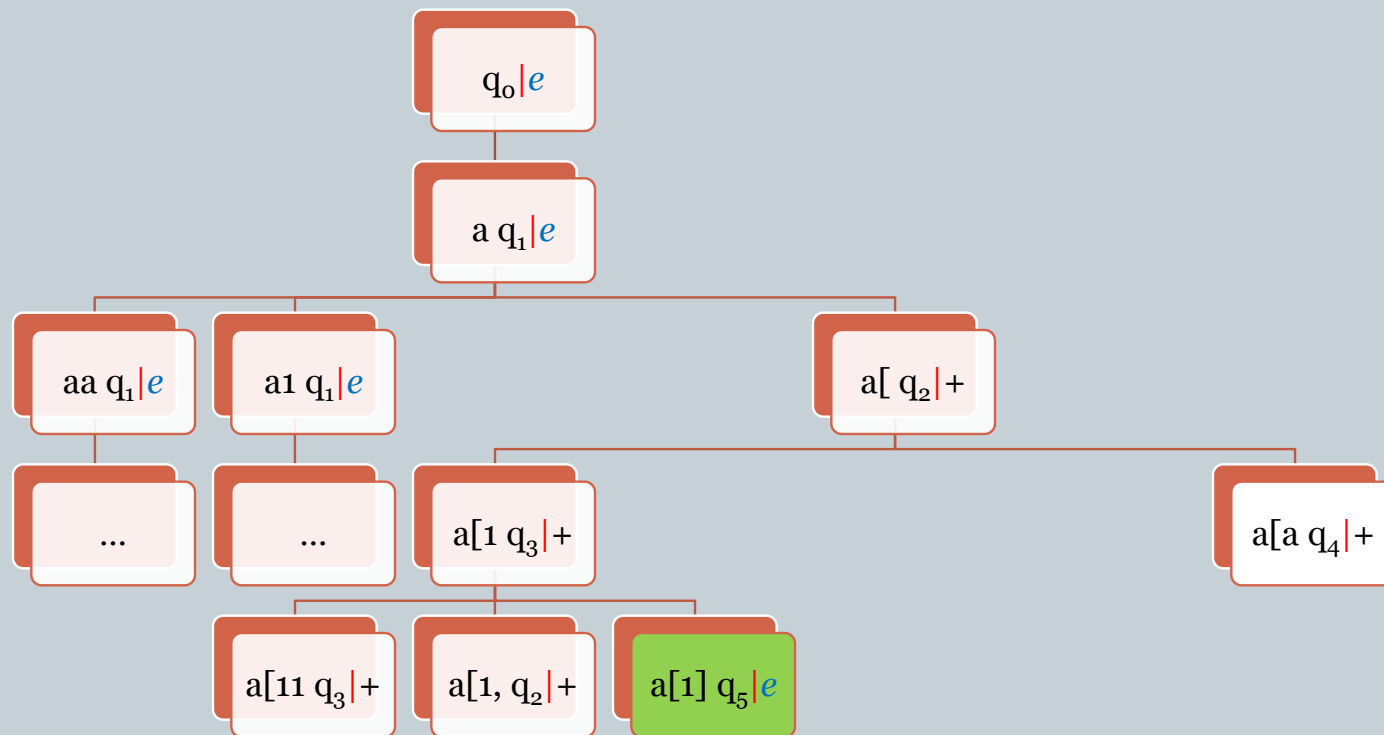
# Дерево вывода ДМПА

34



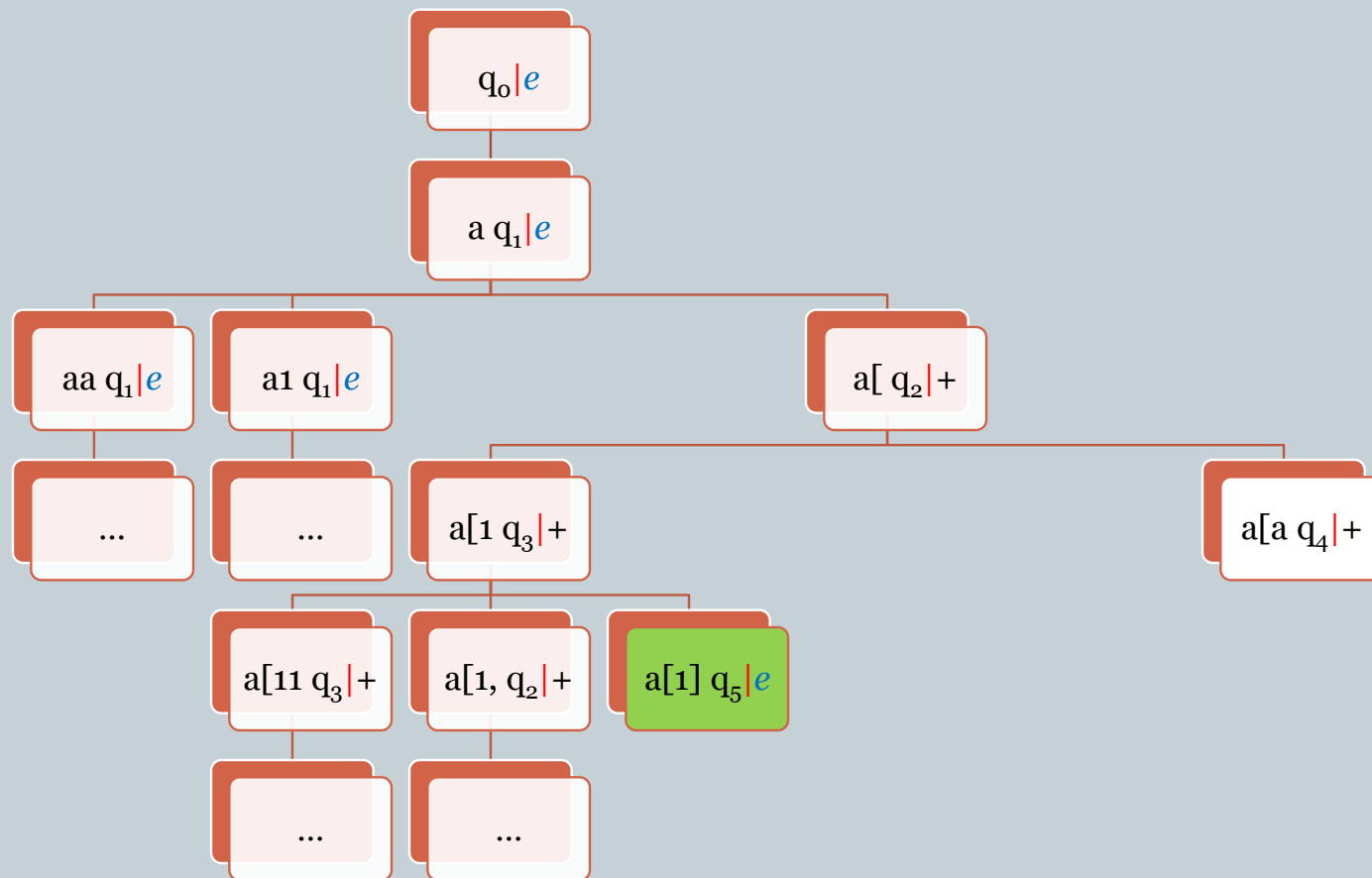
# Дерево вывода ДМПА

35



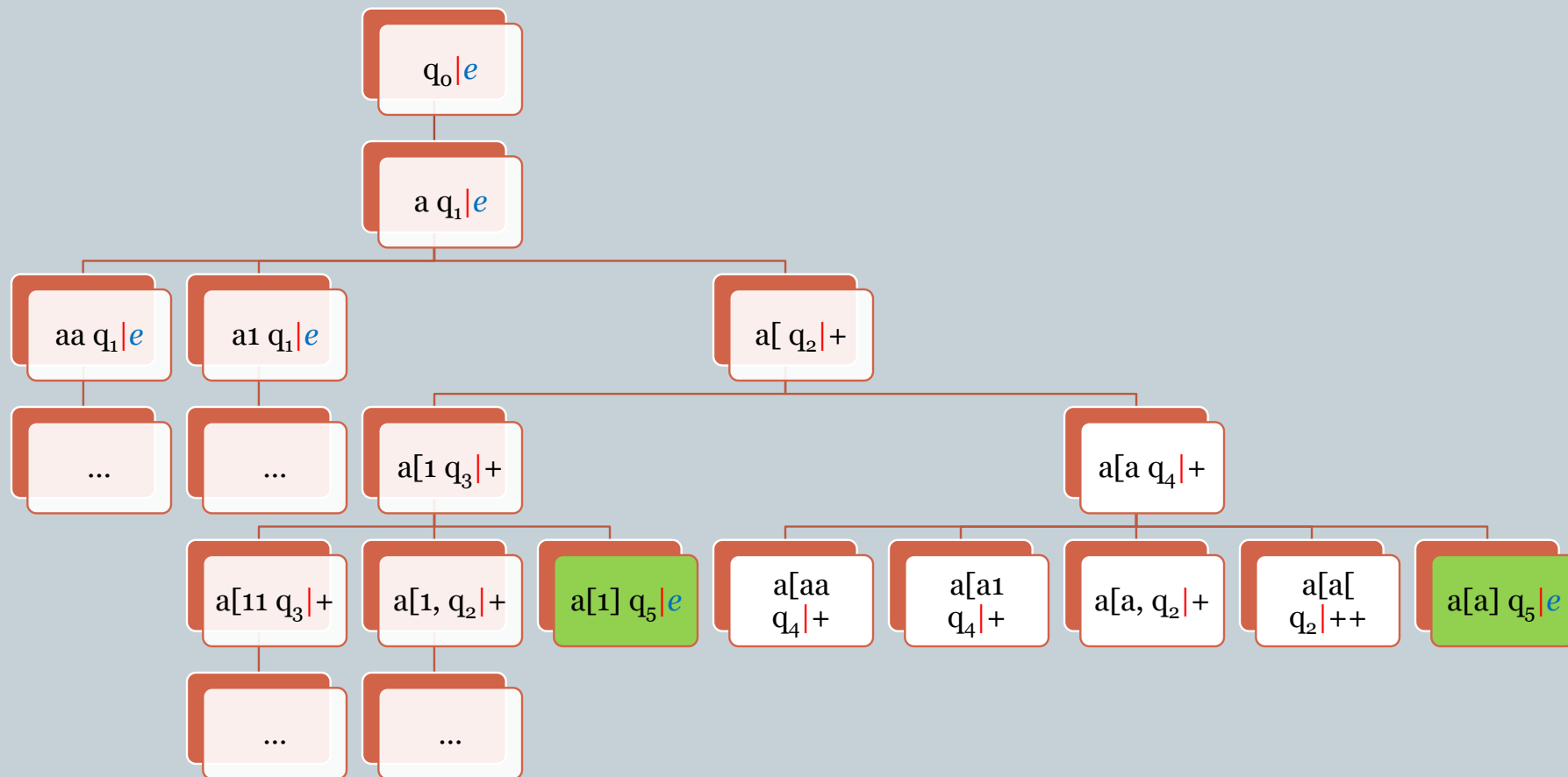
# Дерево вывода ДМПА

36



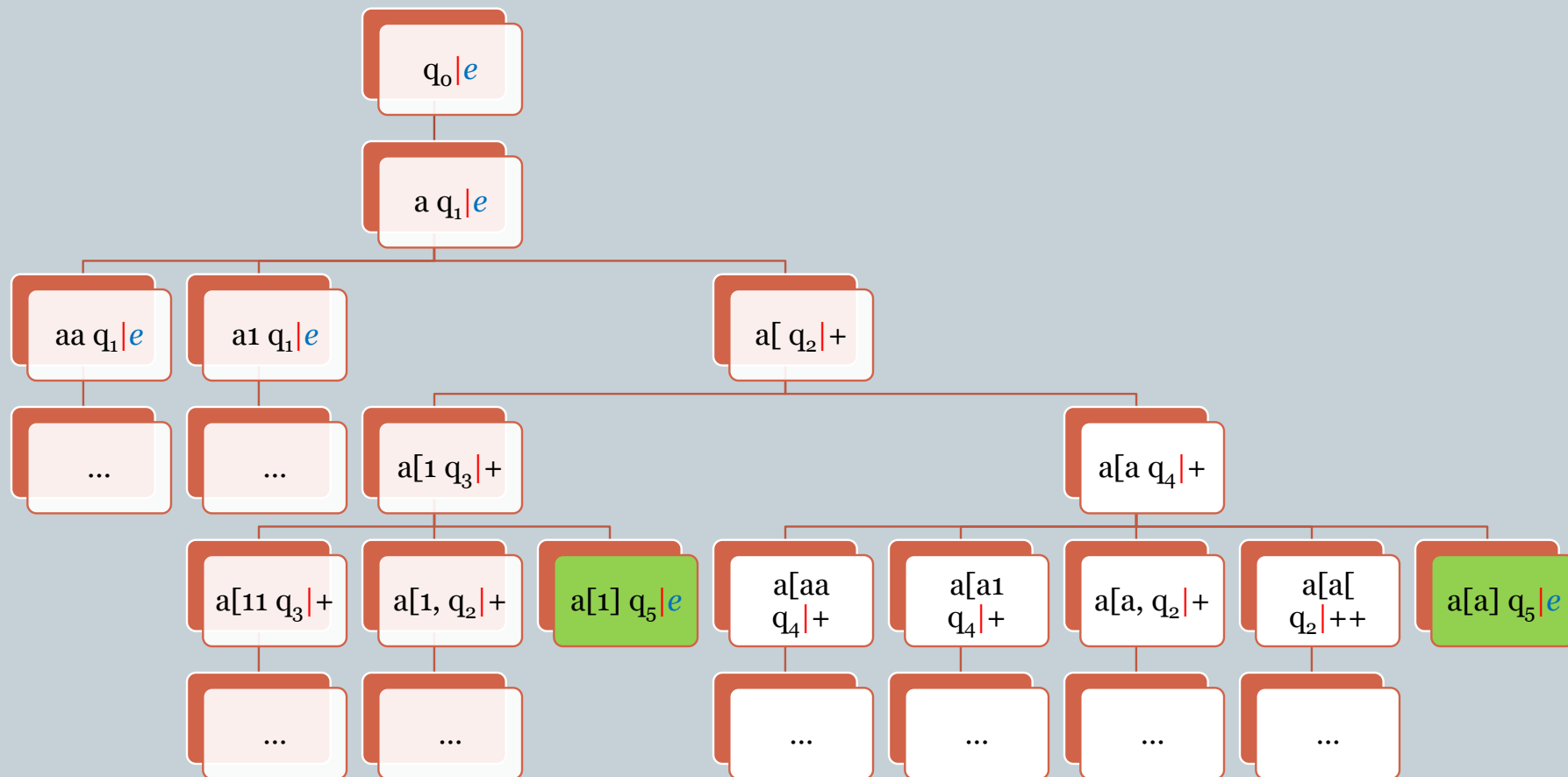
# Дерево вывода ДМПА

37



# Дерево вывода ДМПА

38



# Запуск ДМПА

39

- Для правильной цепочки:

$$\begin{aligned} (q_0, a[b[c], 2] \perp, e) &\Rightarrow^1 (q_1, [b[c], 2] \perp, e) \\ &\Rightarrow^2 (q_2, b[c], 2] \perp, +) \\ &\Rightarrow^3 (q_4, [c], 2] \perp, +) \\ &\Rightarrow^4 (q_2, c], 2] \perp, ++ ) \\ &\Rightarrow^5 (q_4, ], 2] \perp, ++ ) \\ &\Rightarrow^6 (q_5, , 2] \perp, +) \\ &\Rightarrow^7 (q_2, 2] \perp, +) \\ &\Rightarrow^8 (q_3, ] \perp, +) \\ &\Rightarrow^9 (q_5, \perp, e) \\ &\Rightarrow^{10} \text{HALT} \end{aligned}$$

# Запуск ДМПА

40

- Для неправильной цепочки:

$$\begin{aligned}(q_0, a[2], b[2] \perp, e) &\Rightarrow^1 (q_1, [2], b[2] \perp, e) \\ &\Rightarrow^2 (q_2, 2], b[2] \perp, +) \\ &\Rightarrow^3 (q_3, ], b[2] \perp, +) \\ &\Rightarrow^4 (q_5, , b[2] \perp, e) \\ &\Rightarrow^5 \text{ERROR}\end{aligned}$$

Причина ошибки – в состоянии  $q_5$  определён только один переход, когда текущим символом входной цепочки является запятая:

$$\delta(q_5, ,, +) = (q_2, +),$$

но стек пуст.



# Запуск ДМПА

41

- Для неправильной цепочки:

$$\begin{aligned}(q_0, a[b[2]\perp, e) &\Rightarrow^1 (q_1, [b[2]\perp, e) \\ &\Rightarrow^2 (q_2, b[2]\perp, +) \\ &\Rightarrow^3 (q_4, [2]\perp, +) \\ &\Rightarrow^4 (q_2, 2]\perp, ++)) \\ &\Rightarrow^5 (q_3, ]\perp, ++)) \\ &\Rightarrow^6 (q_5, \perp, +) \\ &\Rightarrow^7 \text{ERROR}\end{aligned}$$

Причина ошибки – в состоянии  $q_5$  функция переходов  $\delta(q_5, \perp, +)$  не определена, т.е.  $\delta(q_5, \perp, +) = \text{ERROR}$ . Успешное завершение разбора возможно только в том случае, если стек пуст:  $\delta(q_5, \perp, \emptyset) = \text{HALT}$ .