

Конечные автоматы

1

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

Включение действий в синтаксис

2

Для чего в синтаксис ДКА или ДМПА включают (внедряют) действия?

- Для осуществления дополнительных проверок правил языка L , которые не описываются синтаксисом автомата (т.е. относятся к семантике, или смысловой нагрузке языка L).

Примеры:

$a := (a + a) * a;$

$z = f(a, a, a);$

$\text{var } a, a, a: \text{integer};$

$\text{int } f(\text{int } a, \text{int } a, \text{int } a);$

Включение действий в синтаксис

3

Для чего в синтаксис ДКА или ДМПА включают (внедряют) действия?

- Для выполнения следующего этапа синтаксического анализа и перевода – генерации или интерпретации кода.

Пример:

- построение псевдокода для математического выражения;
- вычисление математического выражения;
- интерпретация кода JavaScript и т.д.

Включение действий в синтаксис

4

Проверка семантики для ЯВУ:

- диапазоны чисел, длина строк;

var a: integer;

int a;

a := 12345678901;

a = 12345678901;

var x: real;

double x;

x := 1.5e+50;

x = 1.5e+500;

var s: string;

s := 'abcde...'; { более 255 символов }

Включение действий в синтаксис

5

Проверка семантики для ЯВУ:

- необъявленные или не инициализированные переменные;

var a, b: integer;

a := 5;

c := a + b;

int a, b;

a = 5;

c = a + b;

Включение действий в синтаксис

6

Проверка семантики для ЯВУ:

- конфликты имён;

```
var a, b, a: integer;
```

```
var real: real;
```

```
var r: record
```

```
  x: integer;
```

```
  x: real;
```

```
end;
```

```
int a, b, a;
```

```
double double;
```

```
struct r {
```

```
  int x;
```

```
  double x;
```

```
};
```

Включение действий в синтаксис

7

Проверка семантики для ЯВУ:

- конфликты имён;

function f(a, a: integer) : real;	double f(int a, int a)
begin	{
end;	}
procedure f;	void f(int a, int a)
begin	{
end.	}

Включение действий в синтаксис

8

Проверка семантики для ЯВУ:

- ошибки с размерностью массивов (неверные границы массива, недопустимые индексы, неправильное количество измерений и т.п.);
- ошибки использования типов данных;

```
var a: array [real] of integer;
```

```
var x: integer;
```

```
x := 1.5;
```

- и т.д.

Основные определения

9

Включённые (внедрённые) в синтаксис действия обозначаются следующим образом:

$$\langle A_1 \rangle, \langle A_2 \rangle, \dots$$

Они добавляются к результату функции переходов:

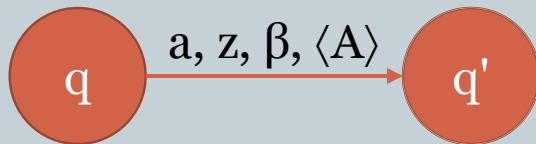
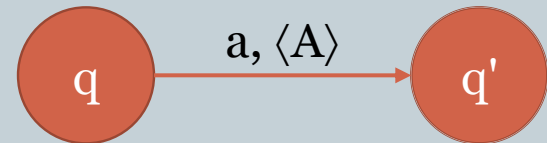
- для ДКА $\delta(q, a) = (q', \langle A \rangle)$;
- для ДМПА $\delta(q, a, z) = (q', \beta, \langle A \rangle)$.

Действие выполняется либо до перехода в состояние q' , либо сразу после (определяется разработчиком).

Основные определения

10

	...	a	...
...
q	...	$q', \langle A \rangle$...
...



	...	a, z	...
...
q	...	$q', \beta, \langle A \rangle$...
...

Основные определения

11

Как описать действие?

- на естественном языке;
- графическим представлением алгоритма;
- словесным описанием алгоритма;
- программным кодом на условном языке;
- программным кодом на ЯВУ.

Т.е. действие, с точки зрения ЯВУ, это некоторая функция.

Основные определения

12

Входные параметры:

- текущее состояние q ;
- текущий символ под считывающей головкой a ;
- верхний символ магазина (стека) z или весь стек в случае ДМПА;
- результат функции переходов $\delta(q, a)$ для ДКА или $\delta(q, a, z)$ для ДМПА (весь или частично, но, как минимум, идентификатор действия $\langle A \rangle$).

Основные определения

13

Результат функции $A(q, a, \delta)$ или $A(q, a, z, \delta)$:

- **true**, если автомат может продолжать выполнение синтаксического анализа.
- **false**, если в результате выполнения действия выяснилось, что входная цепочка содержит семантическую ошибку. В этом случае автомат переходит в состояние **ERROR**.

При необходимости, возможны более сложные действия, например, изменение функцией A текущего состояния автомата и т.д.

Основные определения

14

Как меняется алгоритм построения функции переходов?

- При добавлении новой дуги в граф функции переходов необходимо определить – требуется ли в этом случае выполнение какого-то дополнительного действия. Если да, то проверяем – было ли уже ранее описано такое действие. Если да, то указываем его идентификатор на дуге графа. Если нет, то описываем новое действие, и указываем его идентификатор на дуге графа.

Основные определения

15

Как меняется алгоритм построения функции переходов?

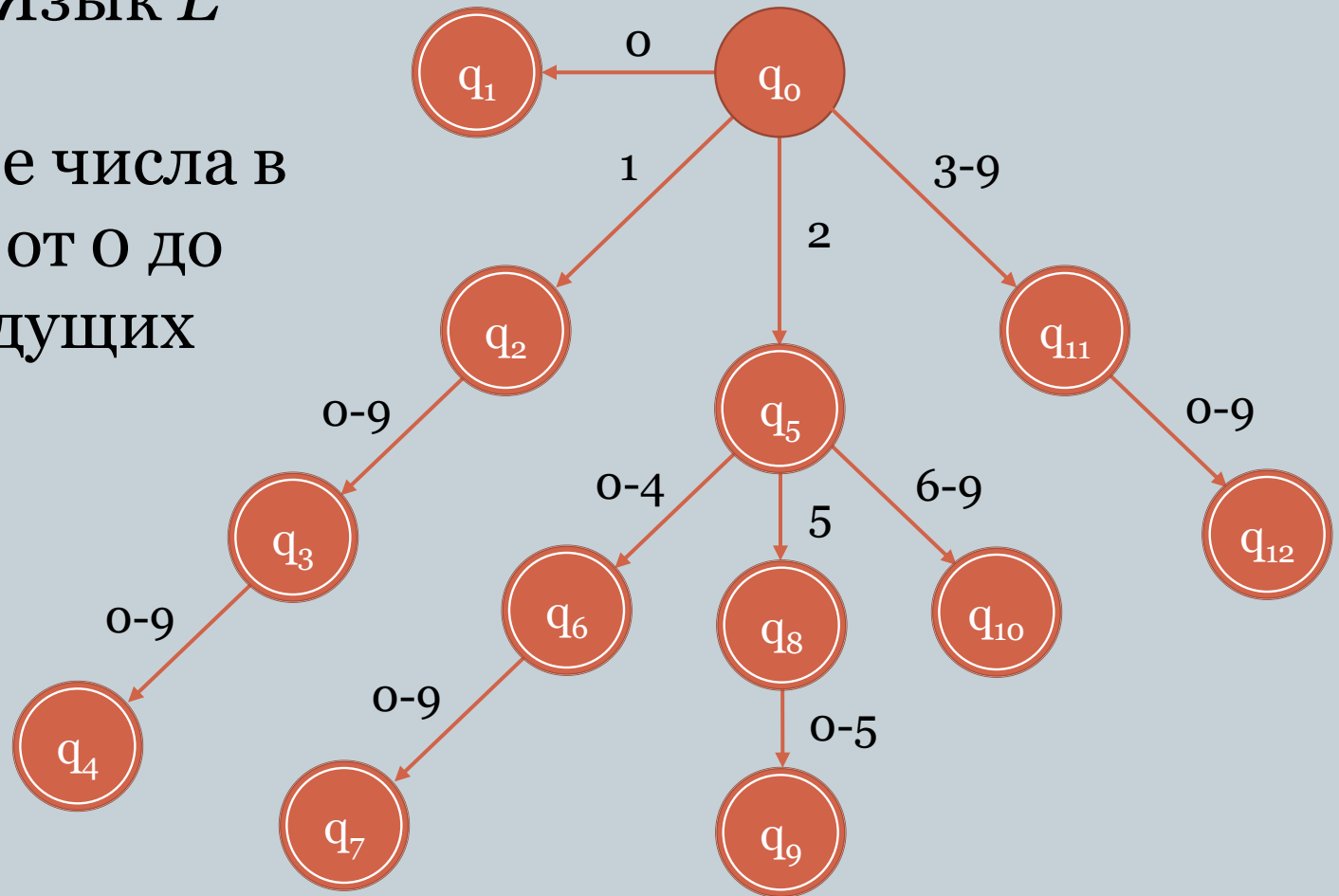
- Иногда требуется совершить дополнительное действие при успешном (**HALT**) или даже неуспешном (**ERROR**) завершении разбора. В графе это отобразить невозможно, только в комментариях к нему, а также в таблице функции переходов:

	...	a	...	\perp
...
q	...	ERROR , $\langle A_i \rangle$...	HALT , $\langle A_j \rangle$
...

Примеры

16

Пример 1. Язык L описывает десятичные числа в диапазоне от 0 до 255, без ведущих нулей.



Примеры

17

	0	1	2	3-4	5	6-9	⊥
q_0	q_1	q_2	q_5	q_{11}	q_{11}	q_{11}	
q_1							HALT
q_2	q_3	q_3	q_3	q_3	q_3	q_3	HALT
q_3	q_4	q_4	q_4	q_4	q_4	q_4	HALT
q_4							HALT
q_5	q_6	q_6	q_6	q_6	q_8	q_{10}	HALT
q_6	q_7	q_7	q_7	q_7	q_7	q_7	HALT
q_7							HALT
q_8	q_9	q_9	q_9	q_9	q_9		HALT
q_9							HALT
q_{10}							HALT
q_{11}	q_{12}	q_{12}	q_{12}	q_{12}	q_{12}	q_{12}	HALT
q_{12}							HALT

18

[illegible]

Примеры

20

	0	1	2	3-4	5	6-9	⊥
q ₀	q ₁	q ₂	q ₄	q ₃	q ₃	q ₃	
q ₁							HALT
q ₂	q ₃	q ₃	q ₃	q ₃	q ₃	q ₃	HALT
q ₃	q ₁	q ₁	q ₁	q ₁	q ₁	q ₁	HALT
q ₄	q ₃	q ₃	q ₃	q ₃	q ₅	q ₁	HALT
q ₅	q ₁	q ₁	q ₁	q ₁	q ₁		HALT

Примеры

21

Получили следующий ДКА $M = (Q, \Sigma, \delta, q_0, F)$:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$;
- $\Sigma = \{0-9\}$;
- $\delta = \{((q_0, 0), q_1), ((q_0, 1), q_2), ((q_0, 2), q_4), ((q_0, 3-9), q_3), ((q_2, 0-9), q_3), ((q_3, 0-9), q_1), ((q_4, 0-4), q_3), ((q_4, 5), q_5), ((q_4, 6-9), q_1), ((q_5, 0-5), q_1)\}$;
- $q_0 = q_0$;
- $F = \{q_1, q_2, q_3, q_4, q_5\}$.

Примеры

22

Или ДКА $M = (Q, \Sigma, \delta, q_0)$:

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$;
- $\Sigma = \{0-9\}$;
- $\delta = \{((q_0, 0), q_1), ((q_0, 1), q_2), ((q_0, 2), q_4), ((q_0, 3-9), q_3), ((q_1, \perp), \text{HALT}), ((q_2, 0-9), q_3), ((q_2, \perp), \text{HALT}), ((q_3, 0-9), q_1), ((q_3, \perp), \text{HALT}), ((q_4, 0-4), q_3), ((q_4, 5), q_5), ((q_4, 6-9), q_1), ((q_4, \perp), \text{HALT}), ((q_5, 0-5), q_1), ((q_5, \perp), \text{HALT})\}$;
- $q_0 = q_0$.

Примеры

23

Плюсы такого подхода:

- Не требуется внедрение действий.

Минусы:

- Сложная функция переходов.
- При изменении допустимого диапазона значений почти все компоненты автомата (Q , δ , F) придётся определять заново.

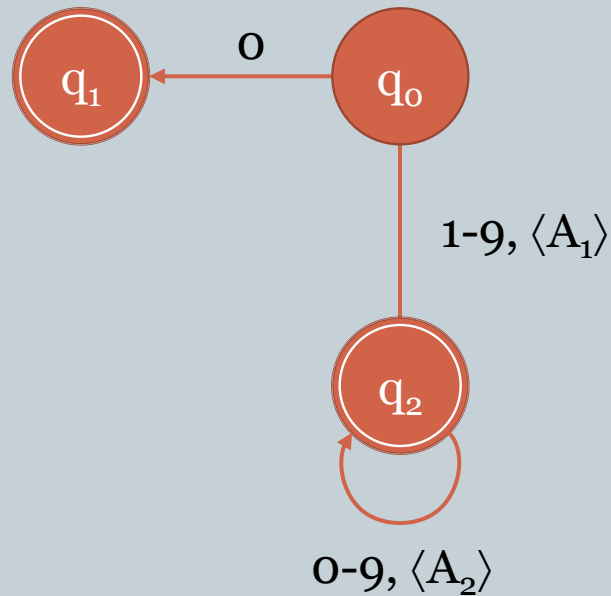
Пример: 4-байтное целое число в диапазоне от $-2\,147\,483\,648$ до $+2\,147\,483\,647$.

- Допустим не для всех типов чисел.

Примеры

24

Вариант с действиями:



Примеры

25

Вариант с действиями:

	0	1-9	\perp
q_0	q_1	$q_2, \langle A_1 \rangle$	
q_1			HALT
q_2	$q_2, \langle A_2 \rangle$	$q_2, \langle A_2 \rangle$	HALT

Функция $A(q, a, \langle A \rangle)$:

- если $\langle A \rangle = \langle A_1 \rangle$, то $n := \text{ЧИСЛО}(a)$, вернуть **true**;
- если $\langle A \rangle = \langle A_2 \rangle$, то $n := n \times 10 + \text{ЧИСЛО}(a)$, если $n \leq 255$, вернуть **true**, иначе вернуть **false**.

Здесь n – целое число.

Примеры

26

Получили следующий ДКА $M = (Q, \Sigma, \delta, q_o, F)$:

- $Q = \{q_o, q_1, q_2\}$;
- $\Sigma = \{0-9\}$;
- $\delta = \{((q_o, 0), (q_1, \langle \rangle)), ((q_o, 1-9), (q_2, \langle A_1 \rangle)), ((q_2, 0-9), (q_2, \langle A_2 \rangle))\}$;
- $q_o = q_o$;
- $F = \{q_1, q_2\}$.

Примеры

27

Запуск автомата для правильной цепочки:

$(q_0, \text{«125}\perp\text{»}) \Rightarrow^1 (q_2, \text{«25}\perp\text{»}), n = 1;$

$\Rightarrow^2 (q_2, \text{«5}\perp\text{»}), n = 1 \times 10 + 2 = 12 < 255;$

$\Rightarrow^3 (q_2, \text{«}\perp\text{»}), n = 12 \times 10 + 5 = 125 < 255;$

$\Rightarrow^4 \text{HALT}$

Запуск автомата для неправильной цепочки:

$(q_0, \text{«5210}\perp\text{»}) \Rightarrow^1 (q_2, \text{«210}\perp\text{»}), n = 5;$

$\Rightarrow^2 (q_2, \text{«10}\perp\text{»}), n = 5 \times 10 + 2 = 52 < 255;$

$\Rightarrow^3 \text{ERROR, т.к. } n = 52 \times 10 + 1 = 521 > 255.$

Примеры

28

Плюсы такого подхода:

- Простая функция переходов.
- При изменении допустимого диапазона значений необходимо лишь исправить число 255 в действии $\langle A_2 \rangle$ на какое-либо другое.
- Допустим для всех типов чисел.

Минусы:

- Требуется написание дополнительной функции для выполнения внедрённых действий.

Примеры

29

Пример 2. Пусть язык L описывает список идентификаторов, разделенных запятыми. Идентификатор должен начинаться с буквы латинского алфавита или подчёркивания, далее могут следовать другие необязательные буквы, подчёркивания или цифры. Также в этой записи допустимы пробелы и другие символы-разделители в начале и конце входной цепочки, а также между идентификаторами и запятыми. Идентификаторы должны быть уникальными.

Примеры

30

Выполним анализ языка.

1. В алфавит войдут большие и маленькие буквы латинского алфавита, символ подчёркивания, цифры, запятая и символы-разделители (обычно в ЯВУ это символы пробела, табуляции, возврата каретки и перехода на следующую строку). Для их обозначения будем использовать символ « □ ».

Примеры

31

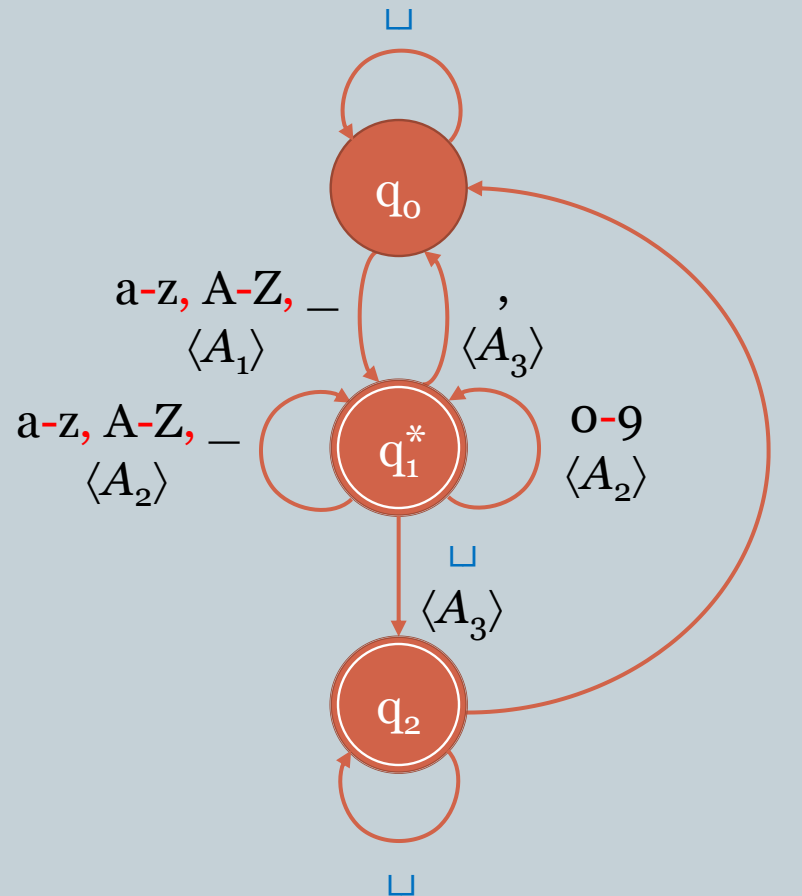
Выполним анализ языка.

2. Внедрённые действия будем использовать для проверки уникальности идентификаторов. Для этого нам нужен какой-то контейнер **CONTAINER** для хранения прочитанных из входной цепочки идентификаторов (массив, список и т.п.). Для выделения из входной цепочки идентификаторов (а мы их считываем посимвольно) также потребуется некоторый строковый буфер **BUFFER**. Соответственно, нужны внедрённые действия для заполнения этого буфера.

Примеры

32

Построим граф функции переходов:



Примеры

33

Табличное представление функции переходов:

	\sqcup	a-z, A-Z, _	0-9	,	\perp
q_0	q_0	$q_1, \langle A_1 \rangle$			
q_1	$q_2, \langle A_3 \rangle$	$q_1, \langle A_2 \rangle$	$q_1, \langle A_2 \rangle$	$q_0, \langle A_3 \rangle$	HALT, $\langle A_3 \rangle$
q_2	q_2			q_0	HALT

Функция $A(q, a, \langle A \rangle)$:

- если $\langle A \rangle = \langle A_1 \rangle$, то BUFFER := a , вернуть true;
- если $\langle A \rangle = \langle A_2 \rangle$, то BUFFER := BUFFER + a , вернуть true;
- если $\langle A \rangle = \langle A_3 \rangle$, то если CONTAINER содержит BUFFER, вернуть false, иначе CONTAINER := CONTAINER + BUFFER, очистить BUFFER, вернуть true.

Примеры

34

Получили следующий ДКА $M = (Q, \Sigma, \delta, q_0, F)$:

- $Q = \{q_0, q_1, q_2\}$;
- $\Sigma = \{a-z, A-Z, _, 0-9, ,, \sqcup\}$;
- $\delta = \{((q_0, \sqcup), (q_0, \langle \rangle)), ((q_0, a-zA-Z_), (q_1, \langle A_1 \rangle)), ((q_1, \sqcup), (q_2, \langle A_3 \rangle)), ((q_1, a-zA-Z_), (q_1, \langle A_2 \rangle)), ((q_1, 0-9), (q_1, \langle A_2 \rangle)), ((q_1, ,,), (q_0, \langle A_3 \rangle)), ((q_2, \sqcup), (q_2, \langle \rangle)), ((q_2, ,,), (q_0, \langle \rangle))\}$;
- $q_0 = q_0$;
- $F = \{q_1, q_2\}$.

Примеры

35

Запуск автомата для правильной цепочки:

$(q_0, \langle x1, x2 \perp \rangle) \Rightarrow^1 (q_1, \langle 1, x2 \perp \rangle), B = \langle x \rangle, C = ()$;
 $\Rightarrow^2 (q_1, \langle , x2 \perp \rangle), B = \langle x1 \rangle, C = ()$;
 $\Rightarrow^3 (q_0, \langle x2 \perp \rangle), B = \langle \rangle, C = (\langle x1 \rangle)$;
 $\Rightarrow^4 (q_0, \langle x2 \perp \rangle), B = \langle \rangle, C = (\langle x1 \rangle)$;
 $\Rightarrow^5 (q_1, \langle 2 \perp \rangle), B = \langle x \rangle, C = (\langle x1 \rangle)$;
 $\Rightarrow^6 (q_1, \langle \perp \rangle), B = \langle x2 \rangle, C = (\langle x1 \rangle)$;
 $\Rightarrow^7 \text{HALT}, B = \langle \rangle, C = (\langle x1 \rangle, \langle x2 \rangle)$.

Примеры

36

Пример 3. Пусть язык L описывает математические выражения, состоящие из целых чисел и знаков сложения и умножения. Необходимо построить автомат, описывающий язык L , а также строящий для выражения обратную польскую запись (ОПЗ), т.е. переводящий выражение в постфиксную форму:

$$1 + 2 \times 3 \Rightarrow 1 \ 2 \ 3 \ \times \ +$$

$$12 \times 5 - 10 / 5 + 1 \Rightarrow 12 \ 5 \ \times \ 10 \ 5 \ / \ - \ 1 \ +$$

$$5 + 7 + 2 \Rightarrow 5 \ 7 \ + \ 2 \ + \text{ или } 5 \ 7 \ 2 \ + \ +$$

Примеры

37

Для чего нужна польская запись?

- Это более простая форма представления математических выражений, в которой не требуются скобки, т.к. порядок выполнения операций определяется только порядком следования операций и операндов ОПЗ:

$$1 + 2 \times 3 \Rightarrow 1 \ 2 \ 3 \ \times \ +$$

$$(1 + 2) \times 3 \Rightarrow 1 \ 2 \ + \ 3 \ \times$$

Примеры

38

Для чего нужна польская запись?

- Выражения, записанные в ОПЗ, компиляторам и интерпретаторам гораздо проще вычислять.

Для ОПЗ, содержащей только бинарные операции, алгоритм вычисления следующий:

1. Пока входная цепочка не пуста:
 - 1.1. Извлечь из неё очередной элемент.
 - 1.2. Если это операнд (идентификатор, число и т.п.), поместить его в стек.
 - 1.3. Если это знак операции, извлечь из стека два операнда, выполнить над ними данную операцию, результат поместить обратно в стек.
2. Результатом вычисления выражения является элемент на вершине стека.

Примеры

39

Например:

$$1 + 2 \times 3 \Rightarrow 1 \ 2 \ 3 \times +$$

Строка	1 2 3 × +	2 3 × +	3 × +	× +	+	
Стек		1	2 1	3 2 1	6 1	7

$$12 \times 5 - 10 / 5 + 1 \Rightarrow 12 \ 5 \times 10 \ 5 / - 1 +$$

Строка	12 5 × 10 5 / - 1 +	5 × 10 5 / - 1 +	× 10 5 / - 1 +	10 5 / - 1 +
Стек		12	5 12	60

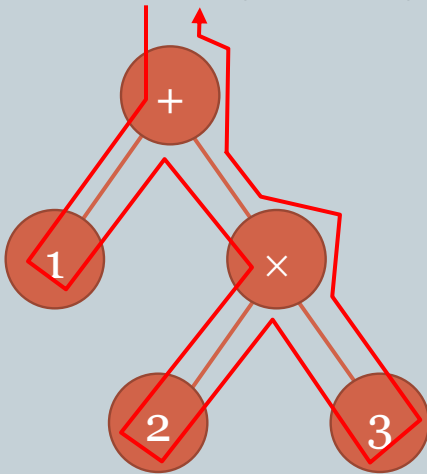
Строка	5 / - 1 +	/ - 1 +	- 1 +	1 +	+	
Стек	10 60	5 10 60	2 60	58	1 58	59

Примеры

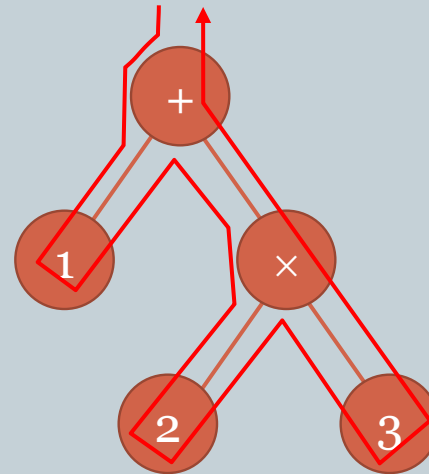
40

Для чего нужна польская запись?

- Это представление дерева, полученное при его обходе в глубину.



+ 1 × 2 3 – префиксная запись



1 2 3 × + – постфиксная запись

1 + 2 × 3 – инфиксная запись

Примеры

41

Алгоритм построения ОПЗ:

1. Пока входная цепочка не пуста:

1.1. Извлечь из неё очередной элемент.

1.2. Если этот операнд, то просто помещаем его в выходную строку.

1.3. Если это открывающая скобка, то помещаем ее в стек.

1.4. Если это закрывающая скобка, то извлекаем символы из стека в выходную строку до тех пор, пока не встретим в стеке открывающую скобку. Саму открывающую скобку со стека просто удаляем.

1.5. Если символ – знак операции, то определяем приоритет данной операции.

Примеры

42

Алгоритм построения ОПЗ:

1.5.1. Если стек пуст, или находящийся на его вершине знак операций имеет меньший приоритет, чем приоритет текущего символа, то помещаем текущий символ в стек.

1.5.2. Пока символы, находящиеся на вершине стека, имеют приоритет, больший или равный приоритету текущего символа, то извлекаем их из стека в выходную строку. После этого текущий символ помещаем в стек.

2. Если вся входная строка разобрана, а в стеке еще остаются знаки операций, извлекаем их из стека в выходную строку.

Примечание. Операции умножения и деления имеют наивысший приоритет (3). Операции сложения и вычитания имеют меньший приоритет (2). Наименьший приоритет (1) имеет открывающая скобка.

Примеры

43

Например: $12 \times 5 - 10 / 5 + 1$

Входная строка	$12 \times 5 - 10 / 5 + 1$	$\times 5 - 10 / 5 + 1$	$5 - 10 / 5 + 1$	$- 10 / 5 + 1$	$10 / 5 + 1$
Выходная строка		12	12	12 5	12 5 ×
Стек			×	×	—

Входная строка	$/ 5 + 1$	$5 + 1$	$+ 1$	1	
Выходная строка	$12\ 5 \times 10$	$12\ 5 \times 10$	$12\ 5 \times 10\ 5$	$12\ 5 \times 10\ 5 / -$	$12\ 5 \times 10\ 5 / - 1$
Стек	—	— /	— /	+	+

Результат: $12\ 5 \times 10\ 5 / - 1 +$

Примеры

44

Выполним анализ языка.

1. В алфавит войдут цифры, а также знаки операций – сложения и умножения.
2. Для формирования ОПЗ нам потребуется стек, т.е. будем строить ДМПА.
3. Согласно алгоритму построения ОПЗ, алфавит магазинных символов включает знаки операций.
4. В начале разбора стек пуст, в конце разбора он также должен быть пуст (это следует из алгоритма построения ОПЗ).

Примеры

45

Построим табличное представление функции переходов:

	0-9, e	+, \emptyset	+, +	+, \times	\times , \emptyset	\times , +	\times , \times	\perp , \emptyset	\perp , +	\perp , \times
q_0	$q_1, e, \langle A_1 \rangle$									
q_1	$q_1, e, \langle A_2 \rangle$	$q_0, +, \langle A_3 \rangle$	$q_0, +, \langle A_4 \rangle$	$q_0, +, \langle A_4 \rangle$	$q_0, \times, \langle A_3 \rangle$	$q_0, +\times, \langle A_3 \rangle$	$q_0, \times, \langle A_4 \rangle$	HALT, $\langle A_3 \rangle$	$q_2, e, \langle A_5 \rangle$	$q_2, e, \langle A_5 \rangle$
q_2								HALT	$q_2, e, \langle A_6 \rangle$	$q_2, e, \langle A_6 \rangle$

$1 + 2 + 3 \Rightarrow 1\ 2 + 3 +$, если $\langle A_4 \rangle$
или $1\ 2\ 3 + +$, если $\langle A_3 \rangle$

Примеры

46

	0-9, e	+, \emptyset	+, +	+, \times	\times , \emptyset	\times , +	\times , \times	\perp , \emptyset	\perp , +	\perp , \times
q_0	$q_1, e, \langle A_1 \rangle$									
q_1	$q_1, e, \langle A_2 \rangle$	$q_0, +, \langle A_3 \rangle$	$q_0, +, \langle A_4 \rangle$	$q_0, +, \langle A_4 \rangle$	$q_0, \times, \langle A_3 \rangle$	$q_0, +\times, \langle A_3 \rangle$	$q_0, \times, \langle A_4 \rangle$	HALT, $\langle A_3 \rangle$	$q_2, e, \langle A_5 \rangle$	$q_2, e, \langle A_5 \rangle$
q_2								HALT	$q_2, e, \langle A_6 \rangle$	$q_2, e, \langle A_6 \rangle$

$\langle A_1 \rangle$: BUFFER := a ;

$\langle A_2 \rangle$: BUFFER := BUFFER + a ;

$\langle A_3 \rangle$: POLSTR := POLSTR + BUFFER, очистить BUFFER;

$\langle A_4 \rangle$: $\langle A_3 \rangle$; пока ПРИОР(z) \geq ПРИОР(a), $M \rightarrow$ POLSTR;

$\langle A_5 \rangle$: $\langle A_3 \rangle$; $M \rightarrow$ POLSTR;

$\langle A_6 \rangle$: $M \rightarrow$ POLSTR.

Примеры

47

Запуск автомата для правильной цепочки:

$(q_0, \langle 1 \times 2 + 3 \times 4 \perp \rangle, \langle \rangle) \Rightarrow^1 (q_1, \langle \times 2 + 3 \times 4 \perp \rangle, \langle \rangle), B = \langle 1 \rangle;$
 $\Rightarrow^2 (q_0, \langle 2 + 3 \times 4 \perp \rangle, \langle \times \rangle), B = \langle \rangle, P = \langle 1 \rangle;$
 $\Rightarrow^3 (q_1, \langle + 3 \times 4 \perp \rangle, \langle \times \rangle), B = \langle 2 \rangle, P = \langle 1 \rangle;$
 $\Rightarrow^4 (q_0, \langle 3 \times 4 \perp \rangle, \langle + \rangle), B = \langle \rangle, P = \langle 1 \ 2 \ \times \rangle;$
 $\Rightarrow^5 (q_1, \langle \times 4 \perp \rangle, \langle + \rangle), B = \langle 3 \rangle, P = \langle 1 \ 2 \ \times \rangle;$
 $\Rightarrow^6 (q_0, \langle 4 \perp \rangle, \langle + \times \rangle), B = \langle \rangle, P = \langle 1 \ 2 \ \times \ 3 \rangle;$
 $\Rightarrow^7 (q_1, \langle \perp \rangle, \langle + \times \rangle), B = \langle 4 \rangle, P = \langle 1 \ 2 \ \times \ 3 \rangle;$
 $\Rightarrow^8 (q_2, \langle \perp \rangle, \langle + \rangle), B = \langle \rangle, P = \langle 1 \ 2 \ \times \ 3 \ 4 \ \times \rangle;$
 $\Rightarrow^9 (q_2, \langle \perp \rangle, \langle \rangle), B = \langle \rangle, P = \langle 1 \ 2 \ \times \ 3 \ 4 \ \times \ + \rangle;$
 $\Rightarrow^{10} \text{HALT}$