



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ»

Кулешов Б.А.

Основы конструирования компиляторов

**Методические указания по рубежному контролю №4
по дисциплине "Машинно-зависимые языки и основы компиляции"
(демо-версия)**

Рецензент: Кулешов Борис Александрович, студент кафедры ИУ6

Кулешов Б.А.

Основы конструирования компиляторов. Методические указания по рубежному контролю №4 по дисциплине "Машинно-зависимые языки и основы компиляции". Электронное учебное издание. - М.: МГТУ имени Н.Э. Баумана, 2019. 17 с.

Определены цели и объем рубежного контроля по дисциплине Машинно-зависимые языки и основы компиляции. Представлен теоретический материал, необходимый для написания рубежного контроля^[1]. Приведены примеры задач, аналогичных решаемым непосредственно во время написания рубежного контроля №4. Описан порядок выполнения работ и определены требования к отчетам.

Для студентов 2 курса кафедры ИУ6 МГТУ имени Н.Э. Баумана, обучающихся по программе бакалавриата направления «Информатика и вычислительная техника», профиль Вычислительные машины, комплексы, системы и сети.

[1] – в демо-версии отсутствует материал по темам нисходящего разбора и фаз компиляции

Оглавление

Задание 1. Лексический и синтаксический анализ...	4
Лексический анализ конструкции...	4
Подготовка к построению диаграммы...	4
Построение синтаксической диаграммы...	5
Построение таблицы конечного автомата...	6
Синтаксический анализ конструкции...	7
Подготовка к построению диаграммы...	7
Построение синтаксической диаграммы...	8
Построение таблицы конечного автомата...	8
Примеры задач...	9
Задание 2. Метод Рутисхаузера, форма Бэкуса-Наура, польская запись...	10
Метод Рутисхаузера...	10
Описание синтаксиса при помощи формы Бэкуса-Наура...	11
Разбор «слева-направо»...	11
Польская запись. Алгоритм Бауэра-Замельзона...	14
Построение польской записи...	15
Выполнение польской записи...	16
Примеры задач...	17

Задание 1. Лексический и синтаксический анализ

Лексический анализ конструкции

Подготовка к построению диаграммы

Рассмотрим процесс решения задачи о построении синтаксической диаграммы и таблицы автомата на примере следующего задания:

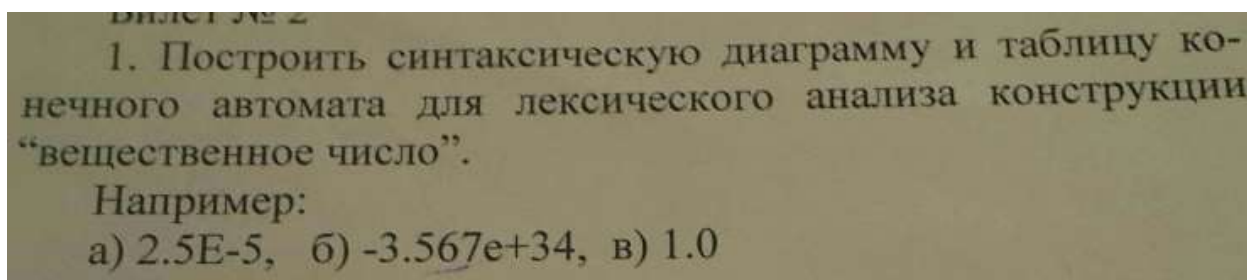


Рисунок 1 – Условие задачи на лексический анализ

Первое, на что стоит обратить внимание, - примеры, указанные в условии задачи. Необходимо проанализировать варианты определения вещественного числа, после чего задать условные обозначения. В нашем случае, есть три группы элементов, которые следует обозначить:

$\mathbf{Ц} - 0|1|2|3|4|5|6|7|8|9;$

$\mathbf{П} - e|E;$

$\mathbf{З} - +|-.$

Здесь **Ц** – любая цифра, **П** – буква e в нижнем или верхнем регистре, **З** – знак + или -. Комбинируя эти элементы, мы можем получить любую из указанных в примере форм определения вещественного числа.

Возможно, вы обратили внимание на то, что ни в одной из групп не указана точка, хотя она и содержится в каждом из примеров. Дело в том, что точка является уникальным символом, не объединяемым в группы ни с каким другим. Если букву «е» мы можем написать как большую, так и маленькую, то точка всегда одинакова. У нее нет вариаций, поэтому для нее не нужна отдельная группа.

При построении диаграммы необходимо соблюдать правила обозначения элементов диаграммы. Заранее обозначенные группы элементов

помещаются в прямоугольники, в то время как отдельные, конкретные символы, будут находиться в кружках.

Пример:

3 - блок, обозначающий знак (+ или -);

. - блок, обозначающий точку.

Также, в большинстве заданий встречается упоминание о символе завершения, или символе конца ввода. Такой символ на синтаксической диаграмме тоже изображается в кружке. В нашей задаче не указан символ конца ввода, но его можно задать. Пусть это будет «#».

Таким образом, мы ввели условные обозначения и определили символ конца ввода. Следующим шагом будет построение самой диаграммы.

Построение синтаксической диаграммы

Для построения синтаксической диаграммы необходимо соединить блоки с ранее обозначенными группами элементов таким образом, чтобы при прохождении по любому пути диаграммы от начала до конца мы получали верную форму вещественного числа.

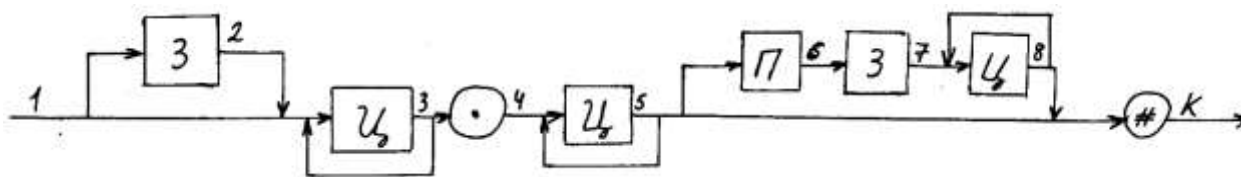


Рисунок 2 – Результирующая синтаксическая диаграмма

Начало диаграммы принято обозначать единицей, конец – буквой К. После каждого блока ставится уникальный номер состояния, в котором находится автомат после получения на вход данного элемента.

Хорошим способом проверить полученную диаграмму будет подстановка примеров из условия задачи. Если какой-то из них нельзя получить прохождением по диаграмме, значит, при ее построении допущена ошибка.

Построение таблицы конечного автомата

При построении таблицы конечного автомата нужно следовать следующим принципам:

- 1) строки таблицы соответствуют состояниям автомата Q ;
- 2) столбцы таблицы соответствуют входным символам W ;
- 3) в ячейках таблицы содержится информация о состоянии, в которое перейдет автомат из состояния Q при поступлении входного символа W .

В нашем автомате всего 8 состояний, не считая конечное состояние **К**. На вход могут поступать символы из описанных ранее групп, точка, символ конца ввода, и любые другие символы. Если на вход приходит символ, который не должен приходить по нашей диаграмме, в таблицу пишется буква **Е**, что обозначает ошибку, то есть введенное выражение не будет являться вещественным числом. Смотрим на построенную синтаксическую диаграмму и заполняем таблицу:

	Ц	П	З	.	#	Другие
1	3	Е	2	Е	Е	Е
2	3	Е	Е	Е	Е	Е
3	3	Е	Е	4	Е	Е
4	5	Е	Е	Е	Е	Е
5	5	6	Е	Е	К	Е
6	Е	Е	7	Е	Е	Е
7	8	Е	Е	Е	Е	Е
8	8	Е	Е	Е	К	Е

Таблица 1 – Таблица конечного автомата для лексического анализа

Таким образом, были построены синтаксическая диаграмма и таблица конечного автомата для лексического анализа конструкции «вещественное число».

Синтаксический анализ конструкции

По своей сути, построение синтаксической диаграммы и таблицы конечного автомата для синтаксического анализа конструкций практически не отличается от построения тех же элементов для лексического анализа. Перед чтением этого раздела, ознакомьтесь с методом лексического анализа конструкций (с. 4).

Подготовка к построению диаграммы

Метод решения этого типа задач будет продемонстрирован на следующем варианте задания:

Построить синтаксическую диаграмму и таблицу конечного автомата для синтаксического анализа конструкции «Объявление массивов и указателей C++». Лексический анализ считать выполненным.

Например:

```
int avp[5], beta [6][7], *cdf[8];
char str[78], *dcvb;
```

Введем условные обозначения:

V – идентификатор (avp, beta, cdf и др.);

N – целочисленная константа;

T – тип данных.

После этого, представим примеры из условия в следующем виде:

T V[N], V[N][N], *V[N];

T V[N], *V;

Таким образом, выделены основные группы элементов синтаксической диаграммы. Символом конца ввода является точка с запятой. Для более подробного ознакомления с принципами построения синтаксических диаграмм прочтите раздел «Подготовка к построению диаграммы» для лексического анализа.

Построение синтаксической диаграммы

Построим синтаксическую диаграмму:

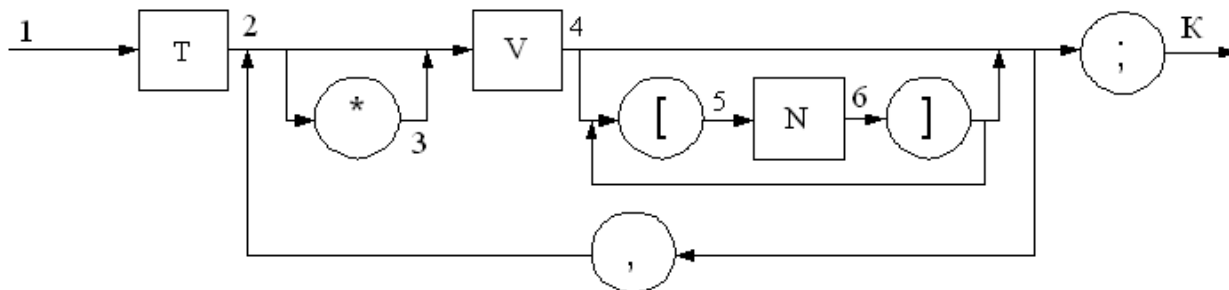


Рисунок 3 – Результирующая синтаксическая диаграмма

Отметим, что в данной диаграмме не ставится номер состояния после блока с запятой. Это связано с тем, что, если бы мы и обозначили это состояние как новое, оно бы ничем не отличалось от состояния 2. При построении таблицы автомата это станет очевидно. То же самое касается блока с закрывающейся квадратной скобкой.

Построение таблицы конечного автомата

В нашем автомате 6 состояний, не считая конечное состояние К. Смотрим на построенную синтаксическую диаграмму и заполняем таблицу:

	Т	N	V	*	,	[]	;	Другие
1	2	Е	Е	Е	Е	Е	Е	Е	Е
2	Е	Е	4	3	Е	Е	Е	Е	Е
3	Е	Е	4	Е	Е	Е	Е	Е	Е
4	Е	Е	Е	Е	2	5	Е	К	Е
5	Е	6	Е	Е	Е	Е	Е	Е	Е
6	Е	Е	Е	Е	Е	Е	4	Е	Е

Таблица 2 – Таблица конечного автомата для синтаксического анализа

Таким образом, были построены синтаксическая диаграмма и таблица конечного автомата для синтаксического анализа конструкции «Объявление массивов и указателей C++».

Примеры задач

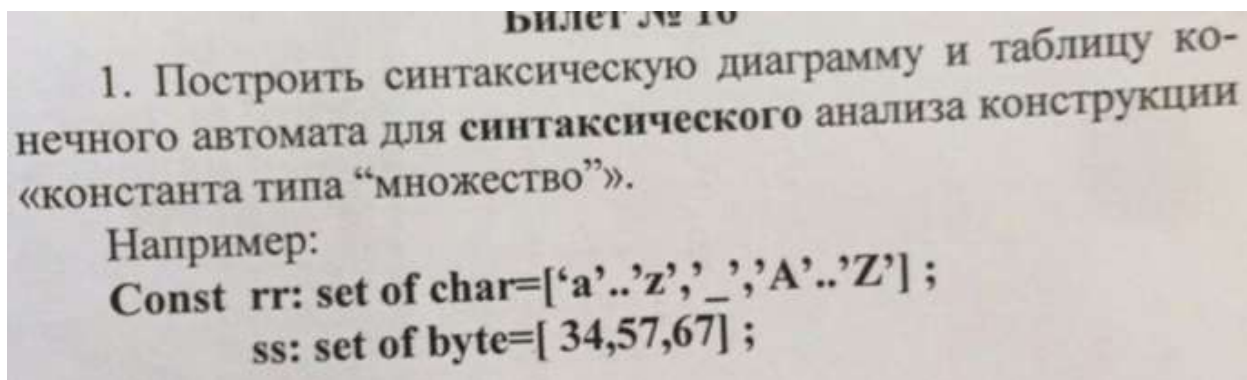


Рисунок 4 – Пример задачи №1 (1)

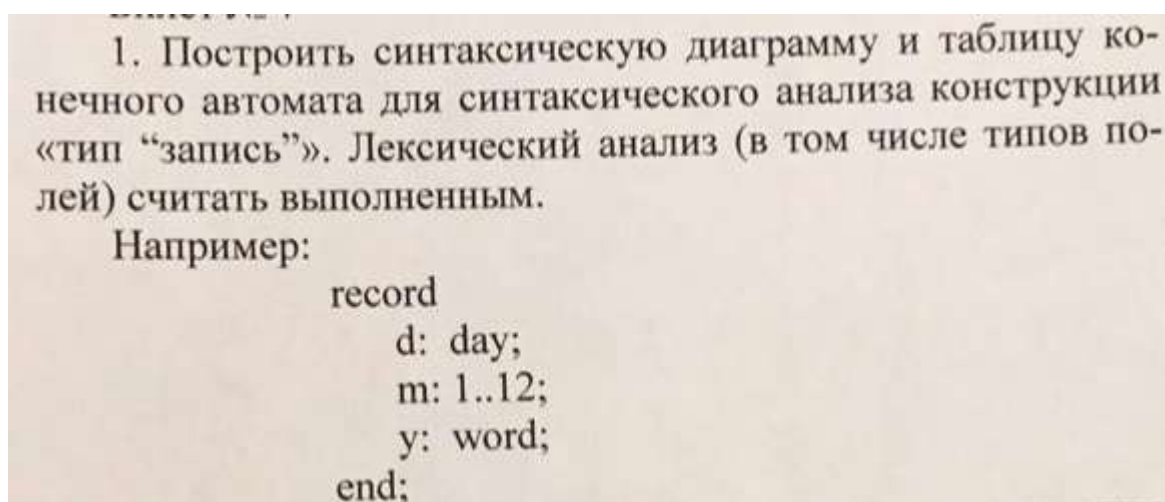


Рисунок 5 – Пример задачи №1 (2)

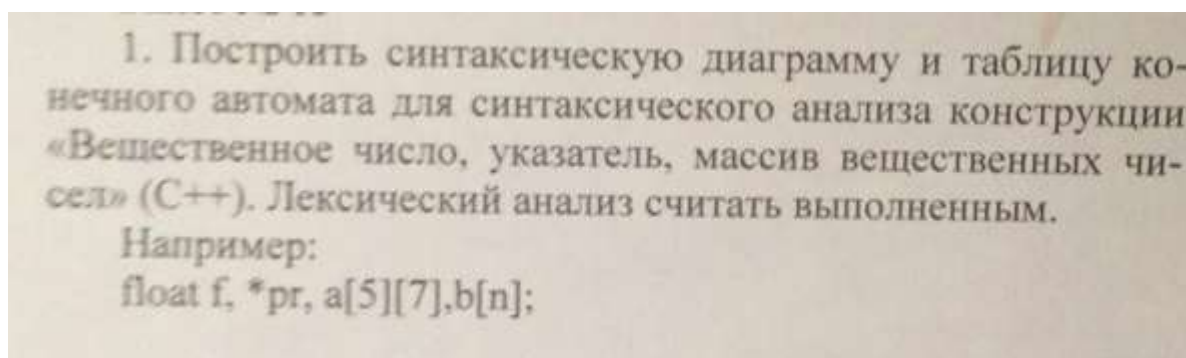


Рисунок 6 – Пример задачи №1 (3)

Задание 2. Метод Рутисхаузера, форма Бэкуса-Наура, польская запись

Метод Рутисхаузера

Задача на метод Рутисхаузера – самая простая из тех, которые встречаются во второй части рубежного контроля №4. В ней вам дается выражение, которое необходимо разбить на тройки. Например:

$((a+b)*c+d)/k$

Первым шагом необходимо привести выражение к полной скобочной форме. Анализатор не знает о том, что умножение имеет приоритет над сложением, поэтому для задания очередности выполнения операций необходимо проставить скобки так, чтобы действия выполнялись в верной последовательности.

В нашем выражении нужно сделать такое изменение:

$((a+b)*c+d)/k \Rightarrow (((a+b)*c)+d)/k$

После получения полной скобочной формы выражения можно приступать к разбору. Сопоставим символам строки **S** цифровые значения **V**. Для этого необходимо знать следующие правила:

- 1) нумерация строки начинается с нуля;
- 2) если встречается открывающаяся скобка или идентификатор, значение увеличивается на единицу;
- 3) если встречается закрывающаяся скобка, знак операции или конец строки, значение уменьшается на единицу.

S		(((a	+	b)	*	c)	+	d)	/	k	
V	0	1	2	<u>3</u>	<u>4</u>	<u>3</u>	<u>4</u>	<u>3</u>	2	3	2	1	2	1	0	1	0

После этого, ищем тройку с наибольшими значениями **V**. В нашем случае это **a+b**. Если получится так, что таких троек несколько, берем первую из них. Выписываем:

$T_1 = a+b$

Далее, заменяем тройку (и скобки, если они есть) на T_1 . Когда мы составили тройку, ее значение V становится равным значению V для операции в ней. Например, для $T_1 = a+b$, $V(+)=3$, значит, $V(T_1)=3$.

$$\begin{array}{cccccccccccc} & (& (& T_1 & * & c &) & + & d &) & / & k \\ 0 & 1 & 2 & 3 & 2 & 3 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \end{array}$$

Повторяем эти действия до тех пор, пока не останется лишь один идентификатор тройки.

$$T_2 = T_1 * c$$

$$\begin{array}{cccccccc} & (& T_2 & + & d &) & / & k \\ 0 & 1 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \end{array}$$

$$T_3 = T_2 + d$$

$$\begin{array}{cccc} & T_3 & / & k \\ 0 & 1 & 0 & 1 & 0 \end{array}$$

$$T_4 = T_3 / k$$

$$\begin{array}{ccc} & T_4 & \\ 0 & 1 & 0 \end{array}$$

Выражение было разбито на тройки по методу Рутисхаузера. Задание выполнено.

Описание синтаксиса при помощи формы Бэкуса-Наура

В рубежном контроле №4 встречаются задания на разбор «слева-направо» и «сверху-вниз». В данной версии методических указаний метод «сверху-вниз» не рассматривается. Вы можете изучить его самостоятельно, ознакомившись с материалами презентации №4 Ивановой Г.С. по дисциплине "Машинно-зависимые языки и основы компиляции".

Разбор «слева-направо»

Сразу стоит оговорить, что весь алгоритм предназначен не для нашего удобства в определении того, целое ли это число, или нет. Человек и так

поймет, что за выражение перед ним; метод необходим для описания того, как машина будет определять, является ли поступившая последовательность символов целым числом.

При выполнении разбора слева направо форма Бэкуса-Наура записывается в варианте с левосторонней рекурсией. Это значит, что сначала вам необходимо написать следующее:

**<целое> ::= <знак><целое без знака>|<целое без знака>,
 <целое без знака> ::= <целое без знака><цифра>|<цифра>,
 <цифра> ::= 0|1|2|3|4|5|6|7|8|9,
 <знак> ::= +| - .**

Рисунок 7 – Форма Бэкуса-Наура с левосторонней рекурсией

В БНФ (форме Бэкуса-Наура) знак | обозначает «или», а знак ::= обозначает «можно заменить на». Здесь рекурсия левосторонняя, так как во второй строке ЦБЗ (целое без знака) можно заменить на ЦБЗ и цифру, при этом ЦБЗ стоит слева от цифры. Если будет написано <ЦБЗ>::=<Цифра><ЦБЗ>|<Цифра> - это будет правосторонняя рекурсия. Она применяется при разборе сверху-вниз, и в этой задаче нас это не интересует.

После записи БНФ приступаем к разбору числа. Для примера будет взято следующее задание:

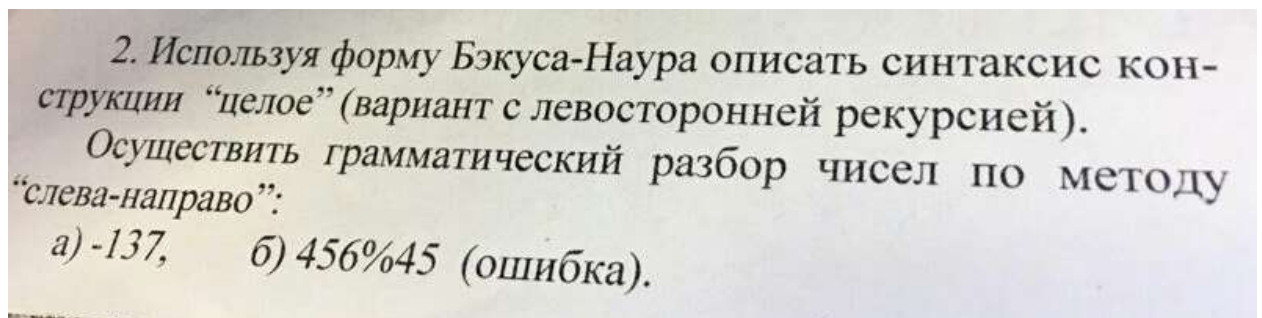


Рисунок 8 - Условие задачи на разбор «слева-направо»

Выполним разбор числа -137, задание пункта «б» можете выполнить самостоятельно. У записанных ранее нами правил БНФ есть номера. Например, если мы производим замену знака и ЦБЗ на целое, то это происходит по правилу 1а (первая строчка, первый вариант выбора). Если бы

мы заменили ЦБЗ на целое, это было бы правило 1б. Если мы заменяем цифру 3 на нетерминал <Цифра>, то этому соответствует правило 3г (третья строка, четвертый вариант).

При выполнении данного вида разбора мы стараемся как можно раньше применять правила более высокого уровня, то есть как можно раньше прийти к форме <Целое>. Начнем разбор.

Правило	Форма числа	Комментарий
	-137	
4б	<Знак>137	Заменяли знак - по правилу 4б. Знак не сворачивается ни по одному правилу
3б	<Знак><Цифра>37	Заменяли цифру 1 по 3б. Цифру мы можем заменить на ЦБЗ по правилу 2б
2б	<Знак><ЦБЗ>37	Заменяли цифру на ЦБЗ. Теперь мы можем заменить знак и ЦБЗ на целое
1а	<Целое>37	Мы пришли в тупик, потому что в итоге нам нужно получить просто <Целое>. Совершаем откат
	<Знак><ЦБЗ>37	Откатились в ближайшую точку с альтернативой
1б	<Знак><Целое>37	Заменяли ЦБЗ на Целое. Все равно тупик. Ищем другую альтернативу
3г	<Знак><ЦБЗ><Цифра>7	Заменяем 3 на <Цифра> по 3г
1а	<Целое><Цифра>7	Заменяем <Знак><ЦБЗ> на <Целое>. Это тупик
	<Знак><ЦБЗ><Цифра>7	
1б	<Знак><Целое><Цифра>7	Заменяем <ЦБЗ> на <Целое>. Это тупик
	<Знак><ЦБЗ><Цифра>7	
2а	<Знак><ЦБЗ> 7	Заменяем <ЦБЗ><Цифра> на <ЦБЗ> по 2а
1а	<Целое>7	Заменяем <Знак><ЦБЗ> на <Целое>. Это тупик
	<Знак><ЦБЗ> 7	
1б	<Знак><Целое> 7	Заменяем <ЦБЗ> на <Целое>. Тупик
	<Знак><ЦБЗ> 7	
3и	<Знак><ЦБЗ> <Цифра>	Заменяем 7 на <Цифра> по 3и

1a	<Целое><Цифра>	Заменяем <Знак><ЦБЗ> на <Целое>. Это тупик
	<Знак><ЦБЗ><Цифра>	
1б	<Знак><Целое><Цифра>	Заменяем <ЦБЗ> на <Целое>. Тупик
	<Знак><ЦБЗ><Цифра>	
2a	<Знак><ЦБЗ>	Заменяем <ЦБЗ><Цифра> на <ЦБЗ> по 2a
1a	<Целое>	Заменяем <Знак><ЦБЗ> на <Целое>. Получаем ответ, что введенная последовательность символов действительно является целым числом

Таблица 3 – Разбор «слева-направо» с комментариями

Внимательно изучите последовательность разбора. Основной принцип заключается в применении правил более высокого уровня как можно раньше, и в откатах с поиском альтернатив.

При решении такой задачи на рубежном контроле не нужно чертить таблицу и писать комментарии, достаточно указывать информацию из первых двух колонок Таблицы 3, указывая тупики. Например:

1a <Целое>37 → Тупик

Польская запись. Алгоритм Бауэра-Замельзона

Для решения задачи этого типа необходимо ознакомиться со следующими обозначениями (рисунок 9):

Польская запись представляет собой последовательность команд двух типов:

K_I , где I – идентификатор операнда – выбрать число по имени I и заслать его в стек операндов;

K_{ξ} , где ξ – операция – выбрать два верхних числа из стека операндов, произвести над ними операцию ξ и занести результат в стек операндов.

Пример:
 $A+B \cdot C \Rightarrow K_A K_B K_C K_{\cdot} K_+$

Рисунок 9 – Команды польской записи

Построение польской записи

При построении польской записи используются следующие правила:

- 1) если символ – операнд, то вырабатывается команда K_i
- 2) если символ – операция, то выполняются действия согласно таблице (рисунок 10):

$\eta \backslash \xi$	+	*	()	←
→	I	I	I	?	Вых
+	II	I	I	IV	IV
*	IV	II	I	IV	IV
(I	I	I	III	?

Рисунок 10 – Таблица операций при построении польской записи

Операции:

- I – заслать ξ в стек операций и читать следующий символ;
- II – генерировать K_η , заслать ξ в стек операций и читать следующий символ;
- III – удалить верхний символ из стека операций и читать следующий символ;
- IV – генерировать K_η и повторить с тем же входным символом.

Рисунок 11 – Описание операций, выполняемых при построении польской записи

Разберем данный тип задач на примере:

Построить тройки для $(a+b*c)/d$.

Рисуем таблицу:

Стек операций	Символ	Действие	Команда

Таблица 4 – Пример таблицы построения польской записи

В стеке операций находятся занесенные в стек операции и скобки. В колонке «Символ» пишется входной символ. В Действии пишется одно из четырех действий из Рисунка 11. В колонку «Команда» заносится команда,

которая должна быть выполнена. Начнем построение. Символом \rightarrow обозначается начало строки, а \leftarrow - конец строки. Построение проводим посимвольно, слева направо.

Стек операций	Символ	Действие	Команда
\rightarrow	(I	
\rightarrow (a		K_a
\rightarrow (+	I	
\rightarrow (+	b		K_b
\rightarrow (+	*	I	
\rightarrow (+*	c		K_c
\rightarrow (+*)	IV	K_*
\rightarrow (+)	IV	K_+
\rightarrow ()	III	
\rightarrow	/	I	
\rightarrow /	d		K_d
\rightarrow /	\leftarrow	IV	$K_/_$
\rightarrow	\leftarrow	Выход	

Таблица 5 – Таблица построения польской записи

Важно понять, что при поступлении на вход любого идентификатора выполняется команда (например, K_a). При поступлении символов операций или скобок, необходимо выполнять действия в соответствии с таблицей операций на Рисунке 10.

После построения таблицы выписываем последовательность команд:

$K_a K_b K_c K_* K_+ K_d K_/_$

Польская запись построена.

Выполнение польской записи

Выполнение польской записи – элементарная задача. Строим таблицу и заносим в нее все команды по очереди. При поступлении команды с идентификатором (например, K_a) заносим a в стек. При поступлении команды с операцией, выполняем эту операцию с двумя последними операндами в стеке, при этом записывая новую тройку.

Стек операндов	Команда	Тройка
\emptyset	K_a	
a	K_b	
ab	K_c	
abc	K_*	$T_1 = b * c$
aT ₁	K_+	$T_2 = a + T_1$
T ₂	K_d	
T ₂ d	$K_/_$	$T_3 = T_2 / d$
T ₃		

Таблица 6 – Выполнение польской записи

Конечное выражение, обозначенное за T₃, и будет нашим исходным выражением. Используйте этот факт для самопроверки.

Примеры задач

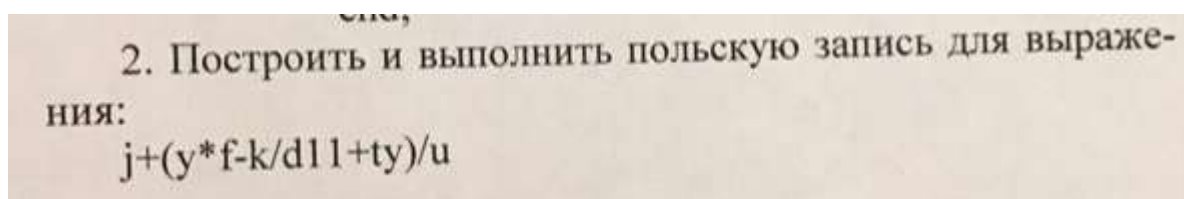


Рисунок 12 – Пример задачи №2 (1)

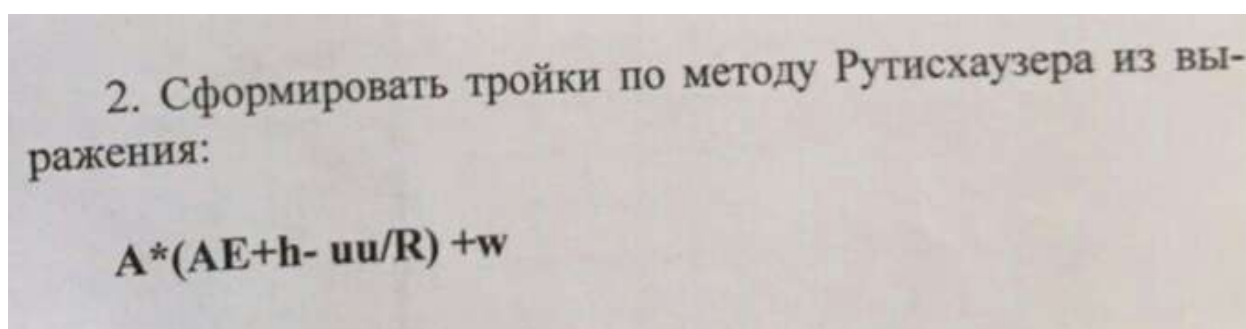


Рисунок 13 – Пример задачи №2 (2)

Используя БНФ описать синтаксис конструкции «целое» (вариант с левосторонней рекурсией).

Осуществить грамматический разбор чисел по методу «слева-направо»:

а) 241\$;

б) -315.