

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Инженерно-технический институт

Кафедра программного обеспечения вычислительной техники
и автоматизированных систем

УТВЕРЖДАЮ

Заведующий кафедрой ПОВТ и АС



С.Г. Федорченко

«29» августа 2022 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

ТЕОРИЯ ФОРМАЛЬНЫХ ЯЗЫКОВ И МЕТОДЫ ТРАНСЛЯЦИИ

Направление подготовки
2.09.03.04 Программная инженерия

Профиль подготовки
Разработка программно-информационных систем

Квалификация (степень)
выпускника:

бакалавр

Форма обучения:

очная, заочная

Год набора:

2020 г.

Разработал:

к.п.н., доцент кафедры ПОВТ и АС,



/С.В. Помян

«29» августа 2022 г.

Тирасполь, 2022

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Теория формальных языков и методы трансляции» у обучающегося должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции
Обязательные профессиональные компетенции выпускников и индикаторы их достижения		
-	ПК-9. Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных	ИД-1 _{ПК-9} Знает методы формальных спецификаций и системы управления базами данных ИД-2 _{ПК-9} Умеет применять современные средства и языки программирования ИД-3 _{ПК-9} Имеет навыки использования операционных систем

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины их название	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
РУБЕЖНЫЙ КОНТРОЛЬ	Раздел 1. Алфавиты, цепочки, языки. Раздел 2. Грамматики. Раздел 3. Распознающие автоматы. Раздел 4. Теория регулярных языков и конечных автоматов.	ПК-9	Контрольная работа №1 Лабораторная работа №1 Лабораторная работа №2
РУБЕЖНАЯ АТТЕСТАЦИЯ	Раздел 5. Теория контекстно-свободных языков и МП-автоматов. Раздел 6. Структура работы компилятора, современные системы программирования. Раздел 7. Распознаватели, задача разбора.		Контрольная работа №2 Тест №1 Лабораторная работа №3
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
№1		ПК-9	Экзамен

3. Показатели и критерии оценивания компетенции по этапам формирования, описание шкал оценивания

Этапы оценивания компетен-	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
Первый этап	ИД-1 _{ПК-9} Знает методы формальных спецификаций и системы управления базами данных	Не знает	Знает основные понятия, но не знает особенностей современных методов формальных спецификаций и систем управления базами данных	Знает основные понятия и основы, но не может применять знания в полной мере в реальных ситуациях	Знает методы формальных спецификаций и системы управления базами данных
Второй этап	ИД-2 _{ПК-9} Умеет применять современные средства и языки программирования	Не умеет	Умеет определить, какие современные средства и языки программирования использовать, но не умеет в полной мере использовать их возможности	Умеет применять современные средства и языки программирования, но не умеет выбирать оптимальные решения	Умеет применять современные средства и языки программирования
Третий этап	ИД-3 _{ПК-9} Имеет навыки использования операционных систем	Не владеет	Владеет навыками использования операционных систем	Имеет навыки использования операционных систем, но не умеет выбирать оптимальные решения	Имеет навыки использования операционных систем для реализации поставленной задачи в полной мере

4. Шкала оценивания

Согласно Положению «О порядке организации аттестации в ИТИ ПГУ им. Т.Г. Шевченко, итоговая оценка представляет собой сумму баллов, полученных студентом по итогу освоения дисциплины (модуля):

Оценка в традиционной шкале	Оценка в 100-балльной шкале	Буквенные эквиваленты оценок в шкале 3Е (% успешно аттестованных)
5 (отлично)	88–100	А (отлично) – 88-100 баллов
4 (хорошо)	70–87	В (очень хорошо) – 80-87баллов
		С (хорошо) – 70-79 баллов
3 (удовлетворительно)	50–69	Д (удовлетворительно) – 60-69 баллов
		Е (посредственно) – 50-59 баллов
2 (неудовлетворительно)	0–49	Гх – неудовлетворительно, с возмож- ной пересдачей – 21-49 баллов
		Г – неудовлетворительно, с повторным изучением дисциплины – 0-20 баллов

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице, указанной ниже

А	“Отлично” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.
В	“Очень хорошо” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
С	“Хорошо” - теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
Д	“Удовлетворительно” - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
Е	“Посредственно” - теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному.
ГХ	“Условно неудовлетворительно” - теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий.

F	“Безусловно неудовлетворительно” - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий.
---	---

5. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций при изучении учебной дисциплины в процессе освоения образовательной программы

5.1 Типовой вариант K1 Контрольная работа 1.

Задание 1. (2 балла)

Дано $V=\{a,b,c, \dots, 0,1,2, \dots, [,], \dots, \text{begin, end, for, } \dots\}$; L = язык Паскаль. Привести примеры цепочек, принадлежащих и не принадлежащих данному языку

Задание 2 (6 баллов)

Дана грамматика $G=(\{S,A,B,C,D,F,E,K\}, \{a,b\}, P, S)$,

$P=\{$

$S \rightarrow KF$	$Bb \rightarrow bB$
$K \rightarrow KB \mid CB$	$Ab \rightarrow bA$
$C \rightarrow CA \mid DA$	$DF \rightarrow E$
$DA \rightarrow aAD$	$BE \rightarrow Eb$
$Aa \rightarrow aA$	$AE \rightarrow Ea$
$DB \rightarrow bBD$	$bE \rightarrow b$

$\}$

- привести примеры цепочек, принадлежащих языку, порождаемому данной грамматикой;
- изобразить дерево вывода для любой приведенной цепочки (для КС-грамматики);
- какой язык порождает грамматика;
- какого типа приведенная грамматика по Хомскому

Задание 3. (5 баллов)

Дан НКА $M=(\{q_0,q_1,q_2,q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$,

$\delta: \delta(q_0,0)=\{q_2\}, \delta(q_2,1)=\{q_1\}, \delta(q_1,0)=\{q_2, q_3\}$

- построить эквивалентный ему ДКА;
- изобразить диаграмму состояний исходного НКА и полученного ДКА.

Задание 4 (4 балла)

Дана регулярная грамматика $G=(\{S,A\}, \{0,1\}, P, S)$,

$P=\{$

$S \rightarrow 0S \mid 0A$
$A \rightarrow 1A \mid 1\}$

$\}$

- построить НКА, порождаемый грамматикой G ;

Задание 5 (4 балла)

Дан ДКА $M=(\{q_0,q_1,q_2,q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

$\delta: \delta(q_0,0)=q_1, \delta(q_1,0)=q_2, \delta(q_1,1)=q_0, \delta(q_2,0)=q_3, \delta(q_2,1)=q_0$ найти регулярную грамматику $G \mid T(M)=L(G)$.

Задание 6 (4 балла)

Преобразовать выражение $B:=B*(E-D+K*5)*E+1$

- в виде синтаксического дерева операций,
- в виде триад,
- в виде тетрад
- в форме обратной польской записи:

Максимальное количество возможных баллов указано в скобках.

Задание 1 (2 балла): 2 балла - задание выполнено полностью верно, 1 балл – верно указаны принадлежащие либо не принадлежащие языку строки, 0 баллов – неверно указаны примеры принадлежащих и не принадлежащих языку цепочек.

Задание 2 (6 баллов): тип приведенной грамматики по Хомскому – 1 балл; примеры цепочек, принадлежащих языку, порождаемому данной грамматикой - 1 балл; дерево вывода для любой приведенной цепочки (для КС-грамматики – 2 балла); язык, который порождает грамматика – 2 балла;

Задание 3 (5 баллов): построенный ДКА, эквивалентный ему данному – 2 балла, минимизация его – 1 балл, диаграммы состояний исходного НКА и полученного ДКА – 2 балла.

Задание 4 (4 балла): НКА, порождаемый грамматикой данной G полностью верно – 4 балла, неверны множества НКА – 3 балла, неверны правила НКА, но верны множества – 2 балла, неверен НКА полностью – 0 баллов.

Задание 5 (4 балла): полностью верная полученная грамматика – 4 балла; неверны множества грамматики – 3 балла, неверны правила полученной грамматики, но верны множества – 2 балла, неверна грамматика построенная полностью – 0 баллов.

Задание 6 (4 балла): синтаксическое дерево операций – 2 балла, триады – 1 балл, тетрады – 1 балл; обратная польская запись – 2 балла

5.2 Типовой вариант К2 Контрольная работа 2.

Задание 1. Дана грамматика G, принадлежит ли строка $\gamma = aababab$ $L(G)$?

$G = (\{a, b\}, \{S, A, B, C, D\}, P, S)$

$P = \{$
 $S \rightarrow AB \mid C$
 $A \rightarrow aA \mid a \mid D$
 $B \rightarrow bB \mid b$
 $C \rightarrow aD$
 $D \rightarrow bC \mid \varepsilon\}$

Задание 2. Дана грамматика G, преобразовать ее к виду без ε -правил

$G = (\{S, A, B, D, E\}, \{a, b, c, e\}, P, S)$, где

$P = \{$
 $S \rightarrow AB \mid \varepsilon;$
 $A \rightarrow Aa \mid S \mid a;$
 $B \rightarrow bD \mid bS \mid b;$
 $D \rightarrow ccD;$
 $E \rightarrow eE \mid e\}$

Задание 3. Дана грамматика G, привести ее к нормальной форме Хомского

$G = (\{a, +, -\}, \{S, K, L, M\}, P, S)$

$P = \{$
 $S \rightarrow KaL \mid Ka \mid +-$
 $K \rightarrow KL \mid a \mid aM$
 $L \rightarrow La \mid +$
 $M \rightarrow KL \mid -\}$

Задание 4. Дан МП – автомат

$M = (\{q_1, q_2\}, \{a, b, c\}, \{R, B, G\}, \delta, q_1, R, \emptyset)$, где

- | | |
|---|---|
| 1) $\delta(q_1, a, R) = \{(q_1, BR)\},$ | 7) $\delta(q_1, c, R) = \{(q_2, R)\},$ |
| 2) $\delta(q_1, a, B) = \{(q_1, BB)\},$ | 8) $\delta(q_1, c, B) = \{(q_2, B)\},$ |
| 3) $\delta(q_1, a, G) = \{(q_1, BG)\},$ | 9) $\delta(q_1, c, G) = \{(q_2, G)\},$ |
| 4) $\delta(q_1, b, R) = \{(q_1, GR)\},$ | 10) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\},$ |
| 5) $\delta(q_1, b, B) = \{(q_1, GB)\},$ | 11) $\delta(q_2, a, B) = \{(q_2, \varepsilon)\},$ |
| 6) $\delta(q_1, b, G) = \{(q_1, GG)\},$ | 12) $\delta(q_2, b, G) = \{(q_2, \varepsilon)\}.$ |

какие цепочки aababbcbababaa, bbabcbabb, aabcbab принадлежат языку, распознаваемому данным автоматом, а какие нет

Задание 5. Дана КС-грамматика

$G = (\{0,1\}, \{S,P,N\}, P, S)$

$P = \{ S \rightarrow S0 \mid S1 \mid P0 \mid P1$

$N \rightarrow 0 \mid 1 \mid N0 \mid N1$

$P \rightarrow 1N \}$

построить эквивалентный МП - автомат.

Каждое задание оценивается в 5 баллов.

5 баллов – задание выполнено полностью, без ошибок, 4 балла – допущена одна ошибка, неверно оформлено, 3 балла – не полностью выполнено задание, 2 балла – неверный ответ, решение содержит критические ошибки; 1 балл – задание было начато, но результата и хода решения нет; 0 баллов – к заданию не приступали. Максимальное количество баллов – 25 баллов

5.3 Лабораторная работа №1 ЛР1.

Тема: Работа с таблицей символов

Цель работы: изучить основные методы организации таблиц идентификаторов, получить представление о преимуществах и недостатках, присущих различным методам организации таблиц символов (идентификаторов).

Для выполнения лабораторной работы требуется написать программу, которая получает на входе набор идентификаторов, организует таблицу по заданному методу и позволяет осуществить многократный поиск идентификатора в этой таблице. Список идентификаторов считать заданным в виде текстового файла. Длина идентификаторов ограничена 32 символами.

Задание: во всех вариантах требуется разработать программу, реализующую комбинированный способ организации таблицы идентификаторов. Для организации таблицы используется простейшая хэш-функция, указанная в варианте задания, а при возникновении коллизий используется дополнительный метод размещения идентификаторов в памяти. Если в качестве этого метода используется дерево или список, то они должны быть связаны с элементом главной хэш-таблицы. В каждом варианте требуется, чтобы программа сообщала среднее число коллизий и среднее количество сравнений, выполненных для поиска идентификатора.

Варианты заданий:

№	Тип хеш-функции (таблицы)	Способ разрешения коллизий
1.	Сумма кодов первой и второй букв	Бинарное дерево
2.	Сумма кодов первой и второй букв	Список с простым перебором
3.	Сумма кодов первой и второй букв	Упорядоченный список с логарифмическим поиском
4.	Сумма кодов первой и второй букв	Простое рехеширование
5.	Сумма кодов первой и второй букв	Рехеширование с использованием случайных чисел
6.	Сумма кодов первой и второй букв	Метод цепочек
7.	Сумма кодов первой и последней букв	Бинарное дерево
8.	Сумма кодов первой и последней букв	Список с простым перебором
9.	Сумма кодов первой и последней букв	Упорядоченный список с логарифмическим поиском
10.	Сумма кодов первой и последней букв	Простое рехеширование

№	Тип хеш-функции (таблицы)	Способ разрешения коллизий
11.	Сумма кодов первой и последней букв	Рехеширование с использованием случайных чисел
12.	Сумма кодов первой и последней букв	Метод цепочек
13.	Бинарное дерево по первой букве	Список с простым перебором
14.	Бинарное дерево по первой букве	Упорядоченный список с логарифмическим поиском
15.	Бинарное дерево по последней букве	Список с простым перебором
16.	Бинарное дерево по последней букве	Упорядоченный список с логарифмическим поиском

Основные контрольные вопросы

1. Что такое таблица символов и для чего она предназначена
 2. Какая информация может храниться в таблице символов
 3. Какие цели преследуются при организации таблицы символов
 4. Какими характеристиками могут обладать константы, переменные
 5. Какие существуют способы организации таблиц символов
 6. В чем заключается алгоритм логарифмического поиска. Какие преимущества он дает по сравнению с простым перебором, и какие он имеет недостатки
 7. Древовидная организация таблиц идентификаторов. В чем ее преимущества и недостатки
 8. В чем суть хеш-адресации
 9. Что такое хеш-функции и для чего они используются
 10. Расскажите о преимуществах, и какие недостатки организации таблицы идентификаторов с помощью хеш-функции.
 11. Что такое коллизия. Почему она происходит. Можно ли полностью избежать коллизий.
 12. Что такое рехеширование. Какие методы рехеширования существуют.
 13. В чем заключается метод цепочек.
 14. Как могут быть скомбинированы различные методы организации хеш-таблиц.
- Полностью выполненное задание – 5 баллов, ответы на вопросы – 5 баллов.

5.4 Лабораторная работа №2 ЛР2.

Тема: Проектирование лексического анализатора

Цель работы: изучение основных понятий теории регулярных грамматик, ознакомление с назначением и принципами работы лексических анализаторов (сканеров), получение практических навыков построения сканера на примере заданного простейшего входного языка.

Задание: Для выполнения лабораторной работы требуется написать программу, которая выполняет лексический анализ входного текста в соответствии с заданием и порождает таблицу лексем с указанием их типов и значений. Текст на входном языке задается в виде символьного (текстового) файла. Программа должна выдавать сообщения о наличии во входном тексте ошибок, которые могут быть обнаружены на этапе лексического анализа. Длину идентификаторов и строковых констант считать ограниченной 32 символами. Программа должна допускать наличие комментариев неограниченной длины во входном файле. Форму организации комментариев предлагается выбрать самостоятельно.

Варианты заданий

1. Входной язык содержит арифметические выражения, разделенные символом ;(точка с запятой). Арифметические выражения состоят из идентификаторов, десятичных чисел с плавающей точкой (в обычной и логарифмической форме), знака присваивания (:=), знаков операций +, -, *, / и круглых скобок.
2. Входной язык содержит логические выражения, разделенные символом ;(точка с запятой). Логические выражения состоят из идентификаторов, констант **true** и **false**, знака присваивания (:=), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
3. Входной язык содержит операторы условия типа **if ... then ... else** и **if ... then**, разделенные символом ;(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, десятичные числа с плавающей точкой (в обычной и логарифмической форме), знак присваивания (:=).
4. Входной язык содержит операторы цикла типа **for (...; ...; ...) do**, разделенные символом ;(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, десятичные числа с плавающей точкой (в обычной и логарифмической форме), знак присваивания (:=).
5. Входной язык содержит арифметические выражения, разделенные символом ;(точка с запятой). Арифметические выражения состоят из идентификаторов, римских чисел, знака присваивания (:=), знаков операций +, -, *, / и круглых скобок.
6. Входной язык содержит логические выражения, разделенные символом ;(точка с запятой). Логические выражения состоят из идентификаторов, констант **0** и **1**, знака присваивания (:=), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
7. Входной язык содержит операторы условия типа **if ... then ... else** и **if ... then**, разделенные символом ;(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, римские числа, знак присваивания (:=).
8. Входной язык содержит операторы цикла типа **for (...; ...; ...) do**, разделенные символом ;(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, римские числа, знак присваивания (:=).
9. Входной язык содержит арифметические выражения, разделенные символом ;(точка с запятой). Арифметические выражения состоят из идентификаторов, шестнадцатеричных чисел, знака присваивания (:=), знаков операций +, -, *, / и круглых скобок.
10. Входной язык содержит логические выражения, разделенные символом ;(точка с запятой). Логические выражения состоят из идентификаторов, шестнадцатеричных чисел, знака присваивания (:=), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.
11. Входной язык содержит операторы условия типа **if ... then ... else** и **if ... then**, разделенные символом ;(точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, шестнадцатеричные числа, знак присваивания (:=).
12. Входной язык содержит операторы цикла типа **for (...; ...; ...) do**, разделенные символом ;(точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, шестнадцатеричные числа, знак присваивания (:=).
13. Входной язык содержит арифметические выражения, разделенные символом ;(точка с запятой). Арифметические выражения состоят из идентификаторов, символьных констант (один символ в одинарных кавычках), знака присваивания (:=), знаков операций +, -, *, / и круглых скобок.

14. Входной язык содержит логические выражения, разделенные символом ; (точка с запятой). Логические выражения состоят из идентификаторов, символьных констант 'T' и 'F', знака присваивания (:=), знаков операций **or**, **xor**, **and**, **not** и круглых скобок.

15. Входной язык содержит операторы условия типа **if ... then ... else** и **if ... then**, разделенные символом ; (точка с запятой). Операторы условия содержат идентификаторы, знаки сравнения <, >, =, строковые константы (последовательность символов в двойных кавычках), знак присваивания (:=).

16. Входной язык содержит операторы цикла типа **for (...; ...; ...) do**, разделенные символом ; (точка с запятой). Операторы цикла содержат идентификаторы, знаки сравнения <, >, =, строковые константы (последовательность символов в двойных кавычках), знак присваивания (:=).

Основные контрольные вопросы

1. Что такое трансляция, компиляция, транслятор, компилятор
2. Из каких процессов состоит компиляция. Общая структура работы компилятора.
3. Какую роль выполняет лексический анализ в процессе компиляции.
4. Как связаны лексический и синтаксический анализ
5. Дайте определение цепочки, языка. Что такое синтаксис и семантика языка
6. Какие существуют методы задания языков. Какие дополнительные вопросы необходимо решить при задании языка программирования
7. Определение грамматики.
8. Как выглядит описание грамматики в форме Бэкуса-Наура.
9. Какие классы грамматик существуют. Что такое регулярные грамматики.
10. Дайте определения контекстно-свободной грамматики, выводимости цепочки, непосредственной выводимости, длины вывода.
11. Что такое лексема. Расскажите, какие типы лексем существуют в языках программирования.
12. Что такое конечный автомат. Дайте определение детерминированного и недетерминированного конечных автоматов.
13. Расскажите о возможности преобразования недетерминированного конечного автомата в детерминированный.
14. Какие проблемы необходимо решить при построении сканера на основе конечного автомата.

Полностью выполненное задание – 5 баллов, ответы на вопросы – 5 баллов.

5.5 Лабораторная работа №3 ЛР3.

Тема: Построение простейшего дерева вывода

Цель работы: изучение основных понятий теории грамматик простого и операторного предшествования, ознакомление с алгоритмами синтаксического анализа (разбора) для некоторых классов КС-грамматик, получение практических навыков создания простейшего синтаксического анализатора для заданной грамматики операторного предшествования.

Задание: Для выполнения лабораторной работы требуется написать программу, которая выполняет лексический анализ входного текста в соответствии с заданием, порождает таблицу лексем и выполняет синтаксический разбор текста по заданной грамматике с построением дерева разбора. Текст на входном языке задается в виде символьного (тексто-

вого) файла. Допускается исходить из условия, что текст содержат не более одного предложения входного языка. Программа должна выдавать сообщения о наличии во входном тексте ошибок.

Рекомендуется программу разбить на три составные части: лексический анализ, построение цепочки вывода и построение дерева вывода. Лексический анализатор должен выделять в тексте лексемы языка и заменять их на терминальный символ грамматики (который в задании обозначен как “**a**”). Полученная после лексического анализа цепочка должна во второй части программы рассматриваться в соответствии с алгоритмом разбора. При неудачном завершении алгоритма выдается сообщение об ошибке, при удачном – строится цепочка вывода. После построения цепочки вывода на ее основе строится дерево разбора, в котором символы **a** последовательно заменяются на лексемы из таблицы лексем.

Длину идентификаторов и строковых констант считать ограниченной 32 символами. Программа должна допускать наличие комментариев неограниченной длины во входном файле. Форму организации комментариев предлагается выбрать самостоятельно.

Ниже приведены варианты грамматик.

Во всех вариантах символ **S** является начальным символом грамматики; **S**, **F**, **T** и **E** обозначают нетерминальные символы. Терминальные символы выделены жирным шрифтом. Вместо символа **a** должны подставляться лексемы.

1. $S \rightarrow a := F;$

$F \rightarrow F + T \mid T$

$T \rightarrow T * E \mid T / E \mid E$

$E \rightarrow (F) \mid \neg(F) \mid a$

2. $S \rightarrow a := F;$

$F \rightarrow F \text{ or } T \mid F \text{ xor } T \mid T$

$T \rightarrow T \text{ and } E \mid E$

$E \rightarrow (F) \mid \text{not } (F) \mid a$

3. $S \rightarrow F;$

$F \rightarrow \text{if } E \text{ then } T \text{ else } F \mid \text{if } E \text{ then } F \mid a := a$

$T \rightarrow \text{if } E \text{ then } T \text{ else } T \mid a := a$

$E \rightarrow a < a \mid a > a \mid a = a$

4. $S \rightarrow F;$

$F \rightarrow \text{for } T \text{ do } F \mid a := a$

$T \rightarrow (F; E; F) \mid (; E; F) \mid (F; E;) \mid (; E;)$

$E \rightarrow a < a \mid a > a \mid a = a$

Варианты заданий

№ варианта	№ варианта грамматики	Допустимые лексемы входного языка
1	1	Идентификаторы, десятичные числа с плавающей точкой
2	2	Идентификаторы, константы true и false
3	3	Идентификаторы, десятичные числа с плавающей точкой
4	4	Идентификаторы, десятичные числа с плавающей точкой
5	1	Идентификаторы, римские числа
6	2	Идентификаторы, константы 0 и 1
7	3	Идентификаторы, римские числа
8	4	Идентификаторы, римские числа
9	1	Идентификаторы, шестнадцатеричные числа
10	2	Идентификаторы, шестнадцатеричные числа
11	3	Идентификаторы, шестнадцатеричные числа
12	4	Идентификаторы, шестнадцатеричные числа
13	1	Идентификаторы, символьные константы (в одинарных кавычках)
14	2	Идентификаторы, символьные константы ‘T’ и ‘F’
15	3	Идентификаторы, строковые константы (в двойных кавычках)
16	4	Идентификаторы, строковые константы (в двойных кавычках)

Примечание:

- римскими числами считать последовательности больших латинских букв X, V и I;
- шестнадцатеричными числами считать последовательность цифр и символов 'a', 'b', 'c', 'd', 'e' и 'f', начинающуюся с цифры (например: 89, 45ac9, 0abc4);
- для выполнения работы рекомендуется использовать лексический анализатор, построенный в ходе выполнения лабораторной работы №2.

Основные контрольные вопросы

1. Какую роль выполняет синтаксический анализ в процессе компиляции.
 2. Какие типы грамматик существуют. Как связаны типы грамматик и языков.
 3. Что такое КС-грамматики. Расскажите об их использовании в компиляторе.
 4. Дайте определение приведенной грамматики. Какие преобразования грамматик существуют.
 5. Поясните правила построения дерева вывода грамматики.
 6. Что такое грамматики простого предшествования.
 7. Как вычисляются отношения предшествования для грамматик простого предшествования.
 8. Что такое грамматика операторного предшествования.
 9. Как вычисляются отношения для грамматик операторного предшествования.
 10. Расскажите о задаче разбора. Что такое распознаватель языка.
 11. Расскажите об общих принципах работы распознавателя языка.
 12. Что такое перенос, свертка. Для чего необходим алгоритм «перенос-свертка».
 13. Расскажите, как работает алгоритм «перенос-свертка» в общем случае (с возвратами).
 14. Как работает алгоритм «перенос-свертка» без возвратов (объясните на своем примере).
- Полностью выполненное задание – 5 баллов, ответы на вопросы – 5 баллов.

5.6 Тест Т1. Итоговый тест по дисциплине. Типовой вариант теста

I/1. Цепочкой или словом в данном алфавите называется

- a) [] любая бесконечная последовательность символов алфавита
- b) [] любая конечная последовательность символов алфавита
- c) [] любая конечная последовательность слов в алфавите

2. Язык L в алфавите V определяется как

- a) [] $L=V$
- b) [] $L \subseteq V$
- c) [] $L=V^*$
- d) [] $L \subseteq V^*$

3. n-ой степенью слова α называется слово

- a) [] $\alpha^n = \underbrace{\varepsilon \dots \varepsilon}_{n \text{ раз}}$
- b) [] $\alpha^n = \underbrace{\alpha \alpha \dots \alpha}_{n \text{ раз}}$
- c) [] $\alpha^n = \underbrace{\alpha \cup \alpha \cup \dots \cup \alpha}_{n \text{ раз}}$

4. Алфавит – это

- a) [] конечное множество цепочек
- b) [] бесконечное множество символов

- с) [] конечное множество символов
5. Существует процедура распознавания предложений языка, тогда этот язык
- [] рекурсивно-перечислимый язык
 - [] рекурсивный язык
6. Существует алгоритм распознавания предложений языка, тогда этот язык
- [] рекурсивно-перечислимый язык
 - [] рекурсивный язык
7. Обращением или реверсом слова $\alpha = abba$ является слово α^{-1}
- [] baba
 - [] abba
 - [] baab
8. Какое свойство не выполняется для операции конкатенации
- [] $(\alpha\beta)\gamma = \alpha(\beta\gamma)$
 - [] $\alpha\beta = \beta\alpha$
 - [] $\alpha(\beta \cup \gamma) = \alpha\beta \cup \alpha\gamma$
9. Чему равно L^0
- [] $\emptyset$
 - [] L
 - [] $\{\varepsilon\}$
10. Дано $V = \{\text{словоформы русского языка}\}$; $L = \text{русский язык}$: Дано $V = \{a, b, c\}$; $L = \{a^n b^n c^n | n > 0\}$, какие из приведенных цепочек не принадлежат языку L
- [] цветы росли на клумбе
 - [] цветы рос клумба
 - [] цветы клумбе росли на
 - [] на клумбе росли цветы
- II/11. Грамматики G_1 и G_2 называются эквивалентными, если
- [] $L(G_1) \subseteq L(G_2)$
 - [] $G_1 = G_2$
 - [] $L(G_1) = L(G_2)$
12. Дана грамматика $G = (\{S, A, B, D\}, \{0, 1\}, P, S)$,
 $P = \{S \rightarrow 0S | S0 | D, D \rightarrow DD | 1A | \varepsilon, A \rightarrow 0B | \varepsilon, B \rightarrow 0A | 0\}$. Определить тип грамматики по Н. Хомскому
- [] 0
 - [] 1
 - [] 2
 - [] 3
13. Цепочка α , для которой $S \xRightarrow{G} \alpha$, называется **сентенциальной формой** в грамматике $G = \langle V_T, V_N, P, S \rangle$ – это
- [] $\alpha \in (V_T \cup V_N)^*$
 - [] $\alpha \in V_T^*$
 - [] $\alpha \in V_N^*$
14. Все правила грамматики типа 0 имеют вид
- [] $A \rightarrow \beta \in P$, где $A \in V_N, \beta \in V^+$.
 - [] $A \rightarrow aB$ или $A \rightarrow a$, где $a \in V_T; A, B \in V_N$.
 - [] $\alpha \rightarrow \beta \in P, |\beta| \geq |\alpha|$
 - [] $\alpha \rightarrow \beta \in P$, где $a, \beta \in V^+$
15. Все правила грамматики типа 1 имеют вид
- [] $A \rightarrow \beta \in P$, где $A \in V_N, \beta \in V^+$.
 - [] $A \rightarrow aB$ или $A \rightarrow a$, где $a \in V_T; A, B \in V_N$.

- c) ☐ $\alpha \rightarrow \beta \in P, |\beta| \geq |\alpha|$ d) ☐ $\alpha \rightarrow \beta \in P$, где $a, \beta \in V^+$.
- 16. Все правила грамматики типа 2 имеют вид**
- a) ☐ $A \rightarrow \beta \in P$, где $A \in V_N, \beta \in V^+$
b) ☐ $A \rightarrow aB$ или $A \rightarrow a$, где $a \in V_T; A, B \in V_N$.
c) ☐ $\alpha \rightarrow \beta \in P, |\beta| \geq |\alpha|$
d) ☐ $\alpha \rightarrow \beta \in P$, где $a, \beta \in V^+$.
- 17. Все правила грамматики типа 3 имеют вид**
- a) ☐ $A \rightarrow \beta \in P$, где $A \in V_N, \beta \in V^+$.
b) ☐ $A \rightarrow aB$ или $A \rightarrow a$, где $a \in V_T; A, B \in V_N$.
c) ☐ $\alpha \rightarrow \beta \in P, |\beta| \geq |\alpha|$
d) ☐ $\alpha \rightarrow \beta \in P$, где $a, \beta \in V^+$.
- 18. Язык, порождаемый грамматикой G – это**
- a) ☐ множество сентенциальных форм
b) ☐ множество нетерминальных сентенциальных форм
c) ☐ множество терминальных сентенциальных форм
- 19. Языком, порождаемым грамматикой $G = \langle V_T, V_N, P, S \rangle$, называется множество**
- a) ☐ $L(G) = \{ \alpha | \alpha \in V_N^*, S \xRightarrow{G} \alpha \}$
b) ☐ $L(G) = \{ \alpha | \alpha \in V_T^*, S \xRightarrow{G} \alpha \}$
c) ☐ $L(G) = \{ \alpha | \alpha \in (V_T \cup V_N)^*, S \xRightarrow{G} \alpha \}$
- III/20. Распознающее устройство грамматики типа 0**
- a) ☐ конечные автоматы
b) ☐ автоматы с магазинной памятью
c) ☐ линейно-ограниченные автоматы
d) ☐ машины Тьюринга
- 21. Распознающее устройство грамматики типа 1**
- a) ☐ конечные автоматы
b) ☐ автоматы с магазинной памятью
c) ☐ линейно-ограниченные автоматы
d) ☐ машины Тьюринга
- 22. Распознающее устройство грамматики типа 2**
- a) ☐ конечные автоматы
b) ☐ автоматы с магазинной памятью
c) ☐ линейно-ограниченные автоматы
d) ☐ машины Тьюринга
- 23. Распознающее устройство грамматики типа 3**
- a) ☐ конечные автоматы
b) ☐ автоматы с магазинной памятью
c) ☐ линейно-ограниченные автоматы
d) ☐ машины Тьюринга
- 24. Если язык L принимается линейно ограниченным автоматом, то L —**
- a) ☐ контекстно-зависимый язык
b) ☐ контекстно-свободный язык
c) ☐ регулярный язык
- 25. Если M — недетерминированный магазинный автомат, и $L = N(M)$, то L —**
- a) ☐ контекстно-свободный язык
b) ☐ контекстно-зависимый язык
c) ☐ регулярный язык

- 26. Распознают входные слова и отвечают на вопрос, принадлежит ли поданное слово данному языку**
- ☐ автоматные сети
 - ☐ автоматы-распознаватели
 - ☐ автоматы преобразователи
- IV/27. Какая команда МП-автомата соответствует правилу КС-грамматики $A \rightarrow \beta$**
- ☐ $\delta(q, \varepsilon, A) = \delta(q, \beta)$
 - ☐ $\delta(q, a, a) = \delta(q, \varepsilon)$
 - ☐ $\delta(q, A, A) = \delta(q, \beta)$
- 28. Если символ $X \in V_T \cup V_N$ не появляется ни в одной сентенциальной форме грамматики, то он называется**
- ☐ бесполезным
 - ☐ недостижимым
 - ☐ бесплодным
- 29. Распознавание принадлежности произвольной строки $\alpha \in V_T^*$ языку $L(G)$ называется**
- ☐ пустотой языка
 - ☐ разрешимостью языка
 - ☐ бесконечностью языка
- 30. В нормальной форме Хомского все правила грамматики имеют вид:**
- ☐ $A \rightarrow BC, A \rightarrow a$, где $a \in V_T; A, B, C \in V_N$
 - ☐ $A \rightarrow a\alpha$, где $a \in V_T; \alpha \in V_N^*$
- 31. В нормальной форме Грейбах все правила грамматики имеют вид:**
- ☐ $A \rightarrow BC, A \rightarrow a$, где $a \in V_T; A, B, C \in V_N$
 - ☐ $A \rightarrow a\alpha$, где $a \in V_T; \alpha \in V_N^*$
- 32. Дан МП-автомат**
 $M = (\{q_1, q_2\}, \{0, 1, c\}, \{R, B, G\}, \delta, q_1, R, \emptyset)$, где
- $\delta(q_1, 0, R) = \{(q_1, BR)\}$,
 - $\delta(q_1, 0, B) = \{(q_1, BB)\}$,
 - $\delta(q_1, 0, G) = \{(q_1, BG)\}$,
 - $\delta(q_1, 1, R) = \{(q_1, GR)\}$,
 - $\delta(q_1, 1, B) = \{(q_1, GB)\}$,
 - $\delta(q_1, 1, G) = \{(q_1, GG)\}$,
 - $\delta(q_1, c, R) = \{(q_2, R)\}$,
 - $\delta(q_1, c, B) = \{(q_2, B)\}$,
 - $\delta(q_1, c, G) = \{(q_2, G)\}$,
 - $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$,
 - $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$,
 - $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$.
- какие цепочки принадлежат языку, распознаваемому данным автоматом
- ☐ 0100c00010
 - ☐ 001c100
 - ☐ 010c01
 - ☐ 101100c001101
- VI/33. На какой фазе работы компилятора получается объектный файл**
- ☐ лексический анализатор
 - ☐ синтаксический анализ
 - ☐ семантический анализ
 - ☐ подготовка к генерации кода
 - ☐ генерация кода
- 34. Исполнимый файл является результатом работы**
- ☐ компилятора
 - ☐ интерпретатора
 - ☐ компоновщика (линковщика)
 - ☐ загрузчика
- 35. Проходом работы компилятора называется**
- ☐ процесс последовательного чтения компилятором данных из внутренней памяти, их обработки и помещения результата работы во внешнюю память

- b) ☐ процесс последовательного чтения компилятором данных из внешней памяти, их обработки и помещения результата работы во внешнюю память
- c) ☐ процесс последовательного чтения компилятором данных из внешней памяти, их обработки и помещения результата работы во внутреннюю память

36. Результатом работы компилятора является

- a) ☐ выполненная программа
- b) ☐ объектный файл
- c) ☐ исполнимый файл

37. Результатом работы интерпретатора

- a) ☐ выполненная программа
- b) ☐ объектный файл
- c) ☐ исполнимый файл

38. Выберите наиболее распространенные способы организации таблиц идентификаторов

- a) ☐ размещение элементов в порядке поступления
- b) ☐ используя списки
- c) ☐ по методу бинарного дерева
- d) ☐ на основе хэш-функций
- e) ☐ двумерный неупорядоченный массив

VII/39. Составные компоненты распознавателя

- a) ☐ лента
- b) ☐ устройство управления
- c) ☐ внешняя память
- d) ☐ внутренняя память
- e) ☐ язык программирования

40. Распознаватель – это

- a) ☐ вычислительная машина
- b) ☐ специальный алгоритм, позволяющий определить принадлежность цепочки языку
- c) ☐ грамматика, позволяющий определить принадлежность цепочки языку

Каждый полностью верный ответ теста оценивается в 0,5 баллов

5.7 Список вопросов к экзамену по дисциплине

1. Трансляция, компиляция, транслятор, компилятор.
2. Из каких процессов состоит компиляция. Общая структура работы компилятора.
3. Какую роль выполняет лексический анализ в процессе компиляции.
4. Как связаны лексический и синтаксический анализ.
5. Что такое лексема, какие типы лексем существуют в языках программирования. Какие проблемы необходимо решить при построении сканера на основе конечного автомата.
6. Какую роль выполняет синтаксический анализ в процессе компиляции.
7. Что такое КС-грамматики, их использование в компиляторе.
8. Задача разбора, распознаватель языка.
9. Распознаватели КС-языков с возвратом.
10. Принципы работы распознавателей с возвратом
11. Нисходящий распознаватель с возвратом. Принцип работы нисходящего распознавателя с подбором альтернатив
12. Распознаватель на основе алгоритма «сдвиг-свертка». Принцип работы восходящего распознавателя по алгоритму «сдвиг-свертка»
13. Принципы построения распознавателей КС-языков без возвратов

14. Нисходящий распознаватель без возвратов. Левосторонний разбор по методу рекурсивного спуска
15. Определение LL(k)-грамматики, свойства LL(k)-грамматик
16. Восходящие распознаватели КС-языков без возвратов. Определение LR(k)-грамматики, свойства LR(k)-грамматик
17. За счет чего обеспечивается возможность генерации кода на разных объектных языках по одному и тому же дереву.
18. Табличные распознаватели
19. Какую роль выполняет генерация объектного кода в процессе компиляции.
20. Структура системы программирования
21. Какие данные необходимы компилятору для генерации объектного кода. Какие действия выполняет компилятор перед генерацией.
22. Какие еще существуют методы для внутреннего представления объектных команд.
23. Компилятор можно построить, не используя лексический анализатор. Объясните, почему подавляющее большинство современных компиляторов используют фазу лексического анализа.
24. От чего зависит количество проходов, необходимых компилятору для построения результирующей объектной программы на основе исходной программы. Как влияют на необходимое количество проходов синтаксис исходного языка программирования, семантика этого языка, архитектура целевой вычислительной системы
25. Почему для языка ассемблера, который имеет простую (часто - регулярную) грамматику, чаще всего используется двухпроходный компилятор. Что мешает свести компиляцию исходного кода на языке ассемблера в один проход.
26. Сравните между собой основные способы внутреннего представления программ. На каких этапах компиляции лучше всего использовать каждый из этих способов. Какие из способов внутреннего представления программы обязательно должен уметь обрабатывать компилятор.
27. Почему имеются трудности с оптимизацией выражений, представленных в форме обратной польской записи
28. Задача разбора. Классификация распознавателей

Практические задания к экзамену

- 1) Дана грамматика G
 - привести примеры цепочек, принадлежащих языку, порождаемому данной грамматикой;
 - изобразить дерево вывода для приведенных двух цепочек;
 - какой язык порождает грамматика;
 - какого типа приведенная грамматика по Хомскому
- 2) Дана грамматика G , привести ее к нормальной форме Хомского
- 3) Дан НКА
 - построить эквивалентный ему ДКА;
 - изобразить диаграмму состояний исходного НКА и полученного ДКА.
- 4) Дана грамматика G , принадлежит ли строка $\gamma \in L(G)$?
- 5) Дана регулярная грамматика G
 - построить НКА, порождаемый грамматикой G ;
 - изобразить диаграмму состояний НКА

- 6) Дан МП-автомат M , какие цепочки принадлежат языку, распознаваемому данным автоматом, а какие нет
- 7) Дана грамматика G , преобразовать ее к виду без цепных правил
- 8) Дан ДКА M , найти регулярную грамматику $G \mid T(M)=L(G)$
- 9) Дана грамматика G , удалить бесплодные символы
- 10) Дана грамматика G , удалить ε - правила
- 11) Дана грамматика G , является ли язык $L(G)$ пустым, конечным или бесконечным
- 12) Дана грамматика G , удалить недостижимые символы
- 13) Дана КС-грамматика, построить эквивалентный МП - автомат M .